ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ
ΤΜΗΜΑ ΒΙΟΜΗΧΑΝΙΚΗΣ ΔΙΟΙΚΗΣΗΣ ΚΑΙ ΤΕΧΝΟΛΟΓΙΑΣ
ΠΜΣ ΣΤΗ ΒΙΟΜΗΧΑΝΙΚΗ ΔΙΟΙΚΗΣΗ ΚΑΙ ΤΕΧΝΟΛΟΓΙΑ
ΕΙΔΙΚΕΥΣΗ: LOGISTICS

ΠΡΟΒΛΕΨΗ ΖΗΤΗΣΗΣ ΜΕ ΧΡΗΣΗ ΜΗΧΑΝΙΚΗΣ ΜΑΘΗΣΗΣ ΚΑΙ ΒΑΣΕΩΝ ΔΕΔΟΜΕΝΩΝ ΣΤΗΝ ΕΦΟΔΙΑΣΤΙΚΗ ΑΛΥΣΙΔΑ

ΝΙΝΟΣ ΚΩΝΣΤΑΝΤΙΝΟΣ

ΕΠΙΒΛΕΠΩΝ: ΓΡΗΓΟΡΙΟΣ ΧΟΝΔΡΟΚΟΥΚΗΣ

ΠΕΙΡΑΙΑΣ, 2025

ΔΗΛΩΣΗ

Η εργασία αυτή είναι πρωτότυπη και εκπονήθηκε αποκλειστικά και μόνο για την απόκτηση του συγκεκριμένου μεταπτυχιακού τίτλου.

Τα πνευματικά δικαιώματα χρησιμοποίησης του μη πρωτότυπου υλικού της ΜΔΕ ανήκουν στον μεταπτυχιακό φοιτητή και στο επιβλέπον μέλος ΔΕΠ εις ολόκληρο, δηλαδή εκάτερος μπορεί να κάνει χρήση αυτών χωρίς τη συναίνεση άλλου. Τα πνευματικά δικαιώματα χρησιμοποίησης του πρωτότυπου μέρους της ΜΔΕ ανήκουν στον μεταπτυχιακό φοιτητή και στον/στην επιβλέποντα από κοινού, δηλαδή δεν μπορεί ο ένας από τους δύο να κάνει χρήση αυτού χωρίς τη συναίνεση του άλλου. Κατ' εξαίρεση, επιτρέπεται η δημοσίευση του πρωτότυπου μέρους της διπλωματικής εργασίας σε επιστημονικό περιοδικό ή πρακτικά συνεδρίου από τον ένα εκ των δύο, με την προϋπόθεση ότι αναφέρονται τα ονόματα και των δύο (ή των τριών σε περίπτωση συνεπιβλέποντα) ως συν-συγγραφέων. Στην περίπτωση αυτή προηγείται γραπτή ενημέρωση του/της μη συμμετέχοντα στη συγγραφή του επιστημονικού άρθρου. Δεν επιτρέπεται η κατά οποιοδήποτε τρόπο δημοσιοποίηση υλικού το οποίο έχει δηλωθεί εγγράφως ως απόρρητο.

Ο/Η Φοιτητής/Φοιτήτρια	Ο/Η Επιβλέπων/Επιβλέπουσα
ΚΩΝΣΤΑΝΤΙΝΟΣ ΝΙΝΟΣ	ΓΡΗΓΟΡΙΟΣ ΧΟΝΔΡΟΚΟΥΚΗΣ

ΠΕΡΙΛΗΨΗ

Στόχος αυτής της μεταπτυχιακής εργασίας είναι η ανάλυση της ζήτησης στην εφοδιαστική αλυσίδα ,πιο συγκεκριμένα ο αναγνώστης έχοντας διαβάσει την παρούσα εργασία θα είναι σε θέση να κατανοήσει την έννοια της ζήτησης , τους προσδιοριστικούς της παράγοντες αλλά και μέσα από ποσοτικές μεθόδους μηχανικής μάθησης να κάνει τις αντίστοιχες προβλέψεις. Η εργασία θεωρεί ότι ο αναγνώστης δεν έχει έρθει σε προηγουμένη επαφή με στατιστικές μεθόδους και βάσεις δεδομένων για αυτό τον λόγο γίνεται αναλυτική αναφορά σε βασικές στατιστικές έννοιές όπως ο μέσος ορός η διακύμανση η τυπική απόκλιση κλπ., επίσης στην εργασία κατασκευάζεται και μια απλή βάση δεδομένων για την διαχείριση προϊόντων ενός μικρού καταστήματος ηλεκτρονικών ειδών. Στην συνέχεια έχοντας αναλύσει τις βασικές έννοιές γίνεται αναφορά σε στατιστικές μεθόδους πρόβλεψης της ζήτησης όπως για παράδειγμα το αυτοπαλίνδρομο τον κινήτο μέσο ορό τον συνδυασμό αυτών των δυο(Arma) και της μεθόδου Arima , προφανώς τα χαρακτηριστικά των παραπάνω αναλύονται σε βάθος όπως για παράδειγμα η στασιμότητα ,η εποχικότητα ,η κυκλικότητα κλπ. Στην συνέχεια γίνεται η ανάλυση της ζήτησης μέσα από μοντέλα μηχανικής μάθησης όπως τα δέντρα αποφάσεων ,οι κ-κοντινότεροι γείτονες ,η γραμμική και λογιστική παλινδρόμηση για ταξινόμηση αλλά και για πρόβλεψη των επομένων τιμών των χρονοσειρων. Τέλος εφαρμόζονται πιο σύγχρονα μοντέλα για την πρόβλεψη της ζήτησης όπως τα νευρωνικά δίκτυα με την χρήση των long sort term memory κυττάρων και του Facebook prophet. Για τις παραπάνω αναλύσεις χρησιμοποιείται η γλώσσα προγραμματισμού python και τα δεδομένα όσο αναφορά την ταξινόμηση έχουν να κάνουν με την πιθανότητα ένας πελάτης να ξανά αγοράσει από το κατάστημα με βάση διάφορες μεταβλητές όπως η ηλικία ,το φύλο ,τα χρήματα που ξόδεψε στην τελευταία του αγορά και αλλά. Ενώ η πρόβλεψη της ζήτησης εφαρμόζεται στο πρώτο παράδειγμα μέσο της πρόβλεψης της τιμή του κινητού τηλεφώνου Samsung s24 ενώ στο δεύτερο παράδειγμα αναλύονται οι εξαμηνιαίες πωλήσεις από το 2012 έως το 2024 των καταστημάτων πώλησης ηλεκτρικών συσκευών πλαίσιο.

Λέξεις-κλειδιά: μηχανική μάθηση , πρόβλεψη ζήτησης, χρόνο σειρές, ταξινόμηση

ABSTRACT

The aim of this master's thesis is to analyze demand in the supply chain. More specifically, after reading this thesis, the reader will be able to understand the concept of demand, its determining factors, and how to make corresponding predictions through quantitative machine learning methods. The thesis assumes that the reader has no prior exposure to statistical methods and databases, and for this reason, it includes a detailed introduction to fundamental statistical concepts such as mean, variance, standard deviation, etc. Additionally, the thesis involves the development of a simple database for managing the products of a small electronics store.

Subsequently, after analyzing the basic concepts, statistical demand forecasting methods are presented, such as autoregression, the moving average, the combination of these two (ARMA), and the ARIMA method. The characteristics of these methods are thoroughly analyzed, including stationarity, seasonality, cyclicality, and so on.

Following this, demand analysis is carried out using machine learning models such as decision trees, Naive Bayes, k-nearest neighbors, linear regression, and logistic regression, for both classification and forecasting the next values of time series data. Finally, more modern models are applied for demand forecasting, such as neural networks, long short-term memory (LSTM) cells, and Facebook Prophet.

For the above analyses, the Python programming language is used. The classification data pertains to the likelihood of a customer repurchasing from a store (customer churn), based on various variables such as age, gender, the amount spent on their last purchase, and others. Demand forecasting is demonstrated through two examples: predicting the price of the Samsung S24 smartphone, and analyzing biannual sales of electronic device stores (Plaisio) from 2012 to 2024.

Key words: machine learning, demand forecasting, time series, classification

ΕΥΧΑΡΙΣΤΙΑΙΕΣ

Με την ολοκλήρωση της παρούσας μεταπτυχιακής εργασίας, θα ήθελα να εκφράσω τις ειλικρινείς μου ευχαριστίες σε όλους εκείνους που με στήριξαν και συνέβαλαν καθοριστικά στην επιτυχή έκβασή της.

Αρχικά, οφείλω να ευχαριστήσω θερμά τον επιβλέποντα καθηγητή μου, κ. Γεώργιο Χονδροκούκη, για την πολύτιμη επιστημονική του καθοδήγηση, τη διαρκή υποστήριξη και τις εύστοχες παρατηρήσεις του σε όλη τη διάρκεια εκπόνησης της εργασίας καθώς η συμβολή του υπήρξε καταλυτική για την ολοκλήρωση της παρούσας μελέτης.

Επίσης, θα ήθελα να εκφράσω την ευγνωμοσύνη μου στην οικογένειά μου για την αμέριστη ηθική και συναισθηματική υποστήριξη, την κατανόηση και την ενθάρρυνση που μου προσέφεραν σε κάθε στάδιο των σπουδών μου.

Τέλος, ευχαριστώ θερμά τους συμφοιτητές μου για τη συνεργασία, την ανταλλαγή ιδεών και την υποστήριξη καθ’ όλη τη διάρκεια του μεταπτυχιακού προγράμματος. Η παρουσία και η συμβολή τους αποτέλεσαν σημαντικό μέρος της ακαδημαϊκής μου εμπειρίας.

ΠΕΡΙΕΧΟΜΕΝΑ

1. ΑΝΑΛΥΣΗ ΖΗΤΗΣΗΣ ΣΤΗΝ ΕΦΟΔΙΑΣΤΙΚΗ ΑΛΥΣΙΔΑ

Η ανάλυση της ζήτησης αποτελεί θεμελιώδη διαδικασία στο πλαίσιο της διοίκησης της εφοδιαστικής αλυσίδας, καθώς παρέχει τον αναγκαίο μηχανισμό για τον αποτελεσματικό προγραμματισμό και τη λήψη στρατηγικών αποφάσεων. Σε ένα οικονομικό και επιχειρησιακό περιβάλλον που χαρακτηρίζεται από υψηλή αβεβαιότητα, έντονη παγκοσμιοποίηση και ταχύτατα μεταβαλλόμενα καταναλωτικά πρότυπα, η ακρίβεια στην κατανόηση και πρόβλεψη της ζήτησης αποκτά κρίσιμο ρόλο για τη βιωσιμότητα και την ανταγωνιστικότητα των επιχειρήσεων. Ειδικότερα στον τομέα των logistics, η ανάλυση της ζήτησης συμβάλλει καθοριστικά στη βελτιστοποίηση κρίσιμων λειτουργιών, όπως η διαχείριση των αποθεμάτων, η κατανομή των μεταφορικών πόρων και η δυναμική ρύθμιση των επιπέδων παραγωγής και αναπλήρωσης. Για παράδειγμα, μια εταιρεία διανομής που βασίζεται σε ενοποιημένο δίκτυο αποθηκών μπορεί, μέσω ακριβών προβλέψεων ζήτησης, να κατανείμει στρατηγικά τα αποθέματά της σε περιφερειακά κέντρα διανομής, μειώνοντας τον χρόνο παράδοσης και το μεταφορικό κόστος. Παράλληλα, ένας πάροχος ταχυμεταφορών μπορεί να προβλέψει αυξήσεις στη ζήτηση κατά την περίοδο των εορτών ή του ηλεκτρονικού εμπορίου (π.χ. Black Friday), ενισχύοντας προσωρινά τον στόλο του και ρυθμίζοντας τους χρόνους παραλαβής και παράδοσης. Η σημασία της ανάλυσης της ζήτησης δεν περιορίζεται μόνο στην αποδοτικότητα. Επεκτείνεται και στην ενίσχυση της προσαρμοστικότητας έναντι εξωτερικών διαταραχών, όπως η μεταβολή των τιμών καυσίμων, η έλλειψη πρώτων υλών ή η διακοπή εφοδιασμού λόγω γεωπολιτικών εντάσεων. Μέσω τεχνικών όπως η ανάλυση χρονοσειρών, τα νευρωνικά δίκτυα, ή τα μοντέλα παλινδρόμησης οι logistics managers μπορούν να εντοπίσουν μοτίβα, εποχικότητες και ασυνήθιστες διακυμάνσεις στη ζήτηση και να προσαρμόσουν εγκαίρως τη στρατηγική τους. Ακόμη και σε μικρότερες επιχειρήσεις του τομέα, όπως καταστήματα ηλεκτρονικών ή εταιρείες αποστολής εμπορευμάτων,η ορθή πρόβλεψη της ζήτησης οδηγεί σε σημαντικά επιχειρηματικά οφέλη. Για παραδειγμα μπορεί να μειώσει τις ελλείψεις αποθεμάτων, να αποφύγει την υπερσυσσώρευση μη αναγκαίων προϊόντων και να βελτιώσει τον ρυθμό εξυπηρέτησης των πελατών. Επιπλέον, η πρόβλεψη της ζήτησης επηρεάζει έμμεσα και τον σχεδιασμό του εργατικού δυναμικού, τη χρήση των αποθηκευτικών χώρων, ακόμα και τη διαχείριση επιστροφών και επισφαλών παραγγελιών. Κατά συνέπεια, η ανάλυση της ζήτησης αποτελεί ουσιώδες εργαλείο πρόβλεψης, συντονισμού και ευθυγράμμισης όλων των κρίκων της εφοδιαστικής αλυσίδας,

προάγοντας την αποδοτικότητα, μειώνοντας το κόστος και ενισχύοντας τη συνολική ικανοποίηση των πελατών.

Η παρούσα διπλωματική εργασία έχει δομηθεί με τον εξής τρόπο στο πρώτο κεφάλαιο γίνεται εισαγωγή στην εφοδιαστική αλυσίδα , στους παράγοντες που επηρεάζουν την ζήτηση και στους τρόπους με τους οποίους μπορούν να εξαχθούν προβλέψεις. Στο δεύτερο κεφάλαιο γίνεται εισαγωγή σε βασικές στατιστικές μεθοδολογίες όπως ο μέσος ορός η διακύμανση , τα διαστήματα εμπιστοσύνης και ο έλεγχος υποθέσεων , στο τρίτο κεφάλαιο γίνεται μια εισαγωγή στις βάσεις δεδομένων πιο συγκεκριμένα αναλύονται τα πλεονεκτήματα της χρήσης βάσεων δεδομένων όπως Erp και crm ενώ στην συνέχεια κατασκευάζεται μια βάση δεδομένων από το μηδέν. Στο κεφάλαιο 4 γίνεται αναφορά στο τι είναι ai και αναλύονται μέθοδοι μηχανικής μάθησης όπως η γραμμική παλινδρόμηση τα support vector machine κλπ. ενώ περιγράφονται και τρόποι αξιολόγησης των προβλέψεων των μοντέλων. Στο κεφάλαιο 5 γίνεται αναφορά στις βασικές βιβλιοθήκες της python για την ανάλυση δεδομένων όπως η NumPy η pandas η TensorFlow η matplotlib κλπ. Στο κεφάλαιο 6 γίνεται η πρώτη εφαρμογή των παραπάνω μοντέλων μηχανικής μάθησης για την ταξινόμηση πελατών ενός ηλεκτρονικού καταστήματος. Ενώ στο κεφάλαιο 7 γίνεται αναλυτική παρουσίαση φαινομένων που παρατηρούνται στις χρονομέτρες όπως η τάση ,η εποχικότητα ,η κυκλικότητα και αλλά ενώ στην συνέχεια αναπτύσσονται μοντέλα ανάλυσης χρονομέτρων όπως τα αυτοπαλινδρομα τα lstm και το Facebook prophet. Στο κεφάλαιο 8 προσπαθούμε να αναπτύξουμε εκείνο το μοντέλο οπού θα μπορέσει να βρει εκείνα τα χαρακτηριστικά των λαπτοπ που προτιμούν οι καταναλωτές και κατά επέκταση είναι διαθετιμενοι να πληρώσουν παραπάνω, με τελικό σκοπό τον καθορισμό της τιμής του εκάστοτε λαπτοπ με βάση τα χαρακτηριστικά του. Τέλος στο κεφάλαιο 9 γίνεται μια ιστορική αναδρομή και ανάλυση του ισολογισμού του ομίλου πλαίσιο ενώ στο τέλος με βάση τις μεθοδολογίες που αναπτυχθήκαν στο κεφάλαιο 7 γίνεται πρόβλεψη των μελλοντικών εξαμηνιαίων πωλήσεων του ομίλου.

1.1 ΕΦΟΔΙΑΣΤΙΚΗ ΑΛΥΣΙΔΑ

Σε αυτήν την ενότητα θα αναλύσουμε το τι είναι και από ποιους κρίκους αποτελείται η εφοδιαστική αλυσίδα. Πιο συγκεκριμένα η εφοδιαστική αλυσίδα ενός προϊόντος ή μιας υπηρεσίας αποτελείται από διάφορους κρίκους αρχικά ο προμηθευτής ή ιδιά η επιχείρηση συλλεγεί τις πρώτες ύλες για την παραγωγή του προϊόντος , στην συνέχεια αυτές οι πρώτες

ύλες μεταφέρονται σε κάποιο εργοστάσιο για την επεξεργασία τους , εν συνέχεια αυτές οι ημικατεργασμενες ύλες συνδυάζονται μαζί με αλλά ημικατεργασμενα προϊόντα για την δημιουργία του τελικού προϊόντος. Στην συνέχεια το προϊόν αποθηκεύεται και πωλείται σε χονδρέμπορες είτε μεταφέρεται στα καταστήματα τις ιδίας της επιχείρησης για την πώληση του. Τέλος ένας ακόμα κρίκος της εφοδιαστικής αλυσίδας είναι εκείνος της διαχείρισης των επιστροφών και των κατεστραμμένων υλικών και πρώτων υλών. Μέσα στην εφοδιαστική αλυσίδα σχηματίζονται διάφορες ροές όπως για παράδειγμα ροές πληροφοριών για παράδειγμα οι εργαζόμενοι στα καταστήματα μπορούν να μεταφέρουν πληροφορίες για τις καταναλωτικές συνήθειες των πελατών στα στελέχη της εταιρεία εκείνοι με την σειρά τους μεταφέρουν αυτήν την πληροφορία στους προμηθευτές της ώστε να μεταβάλουν την παραγωγή τους. Μια άλλη ροή μέσα στην εφοδιαστική αλυσίδα είναι εκείνη των χρήματων , τα χρήματα εισέρχονται στην εφοδιαστική αλυσίδα από τους τελικούς πελάτες του προϊόντος ή της υπηρεσίας και από εκεί μεταφέρονται προς τα πίσω για την πληρωμή των εργαζομένων στο κατάστημα , της μεταφορικής εταιρίας που μετέφεραν τα προϊόντα απευθείας στον καταναλωτή ή στο κατάστημα, την πληρωμή των εργαζομένων της αποθήκης , της παραγωγής και του προμηθευτή των πρώτων υλών και ενδιάμεσων αγαθών. Πιο συγκεκριμένα οι βασικοί κρίκοι της εφοδιαστικής αλυσίδας είναι οι προμηθευτές οι οποίοι διαδραματίζουν τον πιο κρίσιμο ρολό μέσα στην εφοδιαστική αλυσίδα καθώς από αυτούς εξαρτάται η ποιότητα των πρώτων υλών ή των ενδιάμεσων αγαθών , μια μείωση της ποιότητας μπορεί να οδηγήσει σε ελλαττωματικά τελικά προϊόντα που θα έχουν ως αποτέλεσμα η εταιρεία να χάσει πελάτες , να πρέπει να αφιερώσει χρόνο και πόρους για την διαχείριση των επιστροφών και γενικότερα να μειώσει την εικόνα των πελατών προς την μάρκα της εταιρείας. Επίσης ένας άλλος σημαντικός παράγοντας που διαδραματίζει ο προμηθευτής είναι εκείνος της συνέπειας στις παραδόσεις των παραγγελιών , καθώς αν ο προμηθευτής καθυστερεί να αποστείλει τις πρώτες ύλες αυτό θα έχει ως αποτέλεσμα καθυστερήσεις ή ακόμα και τερματισμό της παραγωγικής διαδικασίας στο εργοστάσιο της εταιρείας , αυτό με την σειρά του θα έχει ως αποτέλεσμα ελλείψεις στα ράφια των καταστημάτων και θα οδηγήσει σε χαμένες πωλήσεις , εάν οι προμηθευτές τις εταιρείας δεν είναι συνεπής δεν μπορούν να χρησιμοποιηθούν στρατηγικές όπως η just in time δηλαδή το προϊόν να παράγεται μόνο εφόσον πρώτα ο πελάτης έχει κάνει παραγγελία του προϊόντος, επίσης λόγο αυτού η επιχείρηση θα πρέπει να διακρατει μεγαλύτερες ποσότητες πρώτων υλών και τελικών

προϊόντων ώστε να αποφευχθεί η πιθανότητα δημιουργίας ελλείψεων με αποτέλεσμα να αυξάνονται τα κόστη αποθήκευσης και διαχείρισης των αποθεμάτων.

Ένας άλλος κρίκος της εφοδιαστικής αλυσίδας είναι εκείνος της αποθήκευσης των προϊόντων , προφανώς στόχος της εταιρείας είναι να διακρατει όσο το δυνατόν μικρότερες ποσότητες αποθεμάτων καθώς η υπεραποθεματοποιηση έχει ως αποτέλεσμα τα λειτουργικά κόστη να αυξάνονται όπως για παράδειγμα η πληρωμή των ενοικίων για μια δεύτερη αποθήκη , το ηλεκτρικό ρεύμα , η πληρωμή ασφαλείας κλοπής ή καταστροφής από φυσικά αιτία του αποθέματος , η αγορά επιπλέον εξοπλισμού όπως παλετοφορων και συστημάτων rfid , αυξημένα κόστη μισθοδοσίας λόγο της πρόσληψης επιπλέων picker και packer και διαχειριστών της αποθήκης κτλπ. Επίσης τα αποθέματα ανάλογα το είδος του προϊόντος με το πέρασμα του χρόνου χάνει την αξία του όπως για παράδειγμα προϊόντα τεχνολογίας ή φθείρεται όπως για παράδειγμα τα τρόφιμα οπού λήγουν με το πέρας μερικών ημέρων , γενικότερα τα αποθέματα είναι στοιχεία του ενεργητικού της εταιρείας οπού πρέπει να ρευστοποιούνται άμεσα.

Ένας ακόμα πολύ σημαντικός κρίκος της εφοδιαστικής αλυσίδας είναι εκείνος της μεταφοράς καθώς συνδέει όλους τους κρίκους της εφοδιαστικής αλυσίδας μεταξύ τους. Δηλαδή οι πρώτες ύλες με κάποιο τρόπο μεταφέρονται στο εργοστάσιο από εκεί το τελικό προϊόν μεταφέρεται σε χονδρέμπορες ή σε καταστήματα της ιδίας της εταιρείας ή απευθείας στο καταναλωτή. Η διαφορά μεταξύ μεταφοράς και διανομής είναι ότι η διανομή είναι η μεταφορά των τελικών αγαθών προς τον τελικό χρήση αγοραστή ενώ από την άλλη πλευρά η μεταφορά αναφέρεται σε όλες εκείνες τις μετακινήσεις των συστατικών του τελικού προϊόντος εντός της εφοδιαστικής αλυσίδας. Τα μέσα με τα οποία επιτυγχάνεται χωρίζονται σε τέσσερεις κατηγορίες η πρώτη είναι τα οδικά μέσα όπως τα φορτηγά τα βαν και τα αυτοκίνητα , η δεύτερη είναι μέσο των σιδηροδρόμων η τρίτη κατηγορία είναι η αεροπορική ή τέταρτή είναι αυτή μέσο της θαλάσσιας οδού δηλαδή με πλοία (συμπεριλαμβάνει και την μεταφορά μέσο της ποτάμιας οδού) , και τέλος στην εφοδιαστική αλυσίδα μπορεί να υπάρξει και ο συνδυασμός όλων ή μέρος των παραπάνω για την πραγματοποίηση της μεταφοράς. Προφανώς για να μπορούν να επιτευχθούν οι μεταφορές εκτός από την ύπαρξη των μέσων μεταφοράς θα πρέπει να υπάρχει ένα καλό

σύστημα υποδομών για να επιτευχθεί η μεταφορά δηλαδή αποθήκες , καλό οδικό δίκτυο ,τερματικοί σταθμοί όπως λιμάνια αεροδρόμια κλπ.

Ένας ακόμα κρίκος της εφοδιαστικής αλυσίδας είναι αυτός της διανομής οπού αναφέρθηκε παραπάνω δηλαδή το επονομαζόμενο last mile το οποίο συνηθώς εντός των πόλεων επιτυγχάνεται με την χρήση μικρών οχημάτων όπως μοτοσυκλετών και μικρών βαν , προφανώς υπάρχουν και νέες πιο φιλικές προς το περιβάλλον τάσεις με την χρήση ηλεκτροκίνητων βαν , μικρών αυτονόμων ρομπότ ,ή και cargo bikes δηλαδή ποδήλατων που μπορούν να μεταφέρουν εμπορεύματα. Στόχος της εφοδιαστικής αλυσίδας σε αυτόν τον κρίκο είναι να επιτευχθεί με τον πιο οικονομικό τρόπο δηλαδή να εξοικονομηθούν όσο το δυνατών περισσότερα καύσιμα και εργατοώρες , ενώ παράλληλα προστατεύεται το κοινωνικό συμφέρον δηλαδή αποφυγής δημιουργίας ηχορύπανσης κατά την διανομή , μείωση της κυκλοφοριακής συμφόρησης μέσο βραδύνων δρομολογίων ή μέσο της συνένωσης των εμπορευμάτων ώστε τα οχήματα να είναι πάντα πλήρως φορτωμένα κλπ.

Ο τελευταίος κρίκος της εφοδιαστικής αλυσίδας είναι ο καταναλωτής δηλαδή τελικός αποδέκτης των προϊόντων της εφοδιαστικής , είναι αυτός που δίνει αξία σε όλη την παραπάνω διαδικασία. Μέσο από εκείνον διαμορφώνεται η ζήτηση για τα προϊόντα και τις υπηρεσίες και κατά επέκταση εκείνος είναι που διαμορφώνει κατά ένα μεγάλο ποσοστό την εφοδιαστική αλυσίδα καθώς αν για κάποιον λόγο μειώσει την ζήτηση για ένα προϊόν τότε θα μειωθούν και οι παραγγελίες πρώτων υλών από τον προμηθευτή , αν αυτές μειωθούν τότε θα μειωθεί και ο αριθμός των μεταφορικών μέσων που τις μεταφέρουν προς το εργοστάσιο, το εργοστάσιο ίσως χρειαστεί να μην λειτουργεί 3 βάρδιες αλλά 2 , τα αποθέματα ίσως αυξηθούν κλπ., επίσης εκείνος διαμορφώνει και την ποιότητα του προϊόντος αλλά και της εφοδιαστικής αλυσίδας μέσο της αγοράς του για παράδειγμα ο καταναλωτής μπορεί να προτιμήσει την αγορά προϊόντων οπού η διανομή τους γίνεται με ηλεκτροκίνητα οχήματα που δεν μολύνουν το περιβάλλον.

Αρά από την παραπάνω σύντομη περιγραφή της εφοδιαστικής αλυσίδας γίνεται κατανοητό ποσό σημαντική είναι η ανάλυση της ζήτησης. Καθώς μέσα από αυτήν προσδιορίζεται και σχεδιάζεται η παραγωγική διαδικασία της εφοδιαστικής αλυσίδας, δηλαδή αποφασίζεται ποσά εργοστάσια θα λειτουργήσουν πόσες ώρες θα παραμείνουν ανοιχτά , πόση ποσότητα προϊόντων θα παράγουν, η επένδυση σε νέα μηχανήματα , το ωράριο και ο αριθμός των

εργαζομένων κλπ., επίσης θα παρθούν αποφάσεις για το πόσες πρώτες ύλες και ποτέ πρέπει να γίνει παραγγελία αυτών στους προμηθευτές. Καθώς ο λάθος σχεδιασμός τον παραπάνω μπορεί να οδηγήσει σε κόστη υπερ. αποθεματοποίησης δηλαδή να παραχθούν πολλά προϊόντα τα οποία δεν μπορούν να πουληθούν ή από την άλλη σε ελλείψεις και χαμένες πωλήσεις. Ο ίδιος ακριβός σχεδιασμός μπορεί να πραγματοποιηθεί και για την διαχείριση των αποθεμάτων , δηλαδή μέσο αυτής μπορούν να παρθούν αποφάσεις για την επέκταση της αποθήκης , την αγορά νέου εξοπλίσου και παλετοφορων , την πρόσληψη νέων εργαζομένων , το επίπεδο του αποθέματος ασφάλειας (σε περίπτωση που η ζήτηση αυξηθεί πολύ πανό από τον μέσο ορό που έχει διαμορφωθεί από τις προηγούμενες χρονικές περιόδους) κλπ. οι βάσεις δεδομένων όπως τα wms και η ανάλυση μοντέλων όπως τα EOQ (economic order quantity) μπορούν επίσης να διασφαλίσουν μια σταθερότητα στην διαχείριση των αποθεμάτων. Επίσης γνωρίζοντας την ζήτηση μπορεί να αναδιαμορφωθεί και ο τρόπος που πραγματοποιούνται οι μεταφορές , για παράδειγμα αν η εταιρεία γνωρίζει ότι η ζήτηση θα αυξηθεί μπορεί να κάνει μεγάλες παραγγελίες στους προμηθευτές πρώτων υλών και να επωφεληθεί από εκπτώσεις στην συνολική τιμή. Επίσης με βάση την ζήτηση μπορούν να επιλέγουν τα καταλληλά οχήματα για την εκάστοτε μεταφορά και να οργανωθούν με τον πιο βέλτιστο τρόπο τα δρομολόγια ελαχιστοποιώντας την απόσταση μεταξύ των σημείων παράδοσης επιτυγχάνοντας με αυτόν τον τόπο την μείωση διάφορων λειτουργικών εξόδων όπως την κατανάλωση καυσίμου.

1.2 ΑΝΑΛΥΣΗ ΖΗΤΗΣΗΣ ΚΑΙ ΟΙ ΠΡΟΣΔΙΟΡΙΣΤΙΚΟΙ ΤΗΣ ΠΑΡΑΓΩΝΤΕΣ

Σε αυτήν την ενότητα θα ορίσουμε την έννοια της ζήτησης και στην συνέχεια θα αναλύσουμε αρκετούς παράγοντες οπού την επηρεάζουν. Στην συγκεκριμένη εργασία επειδή τα περισσότερα παραδείγματα είναι στον τομέα του λιανικού εμπορίου όπως για παράδειγμα η πρόβλεψη της τιμής του κινητού s24 , η μοντελοποιηση της τιμής των λαπτοπ με βάση τα χαρακτηριστικά τους ,και η ανάλυση του ισολογισμού της εταιρείας πλαίσιο , θα εστιάζουμε κυρίως στους παράγοντες που επηρεάζουν την ζήτηση στο λιανικό εμπόριο και πιο συγκεκριμένα στα είδη τεχνολογίας, προφανώς και υπόλοιποι τομείς της οικονομίας όπως τα τουριστικά, οι βιομηχανίες κλπ. επηρεάζονται σε μεγάλο βαθμό από τους ιδίους παράγοντες.

Η ζήτηση είναι η ποσότητα ενός αγαθού ή μιας υπηρεσίας που είναι διαθετιμενοι να αγοράσουν οι καταναλωτές σε μια συγκεκριμένη τιμή , χρονική περίοδο , και γεωγραφική περιοχή. Ο πιο βασικός προσδιοριστικός παράγοντας αυτής είναι η τιμή του προϊόντος , με βάση τον νομό της ζήτησης και για τα περισσότερα προϊόντα ισχύει ότι κρατώντας όλους τους άλλους προσδιοριστικούς παράγοντες σταθερούς μια αύξηση στην τιμή ενός προϊόντος θα έχει ως αποτέλεσμα την μείωση της ζητούμενης ποσότητας του, από την άλλη πλευρά μια μείωση στην τιμή θα έχει ως αποτέλεσμα την αύξηση της ζήτησης για το συγκεκριμένο προϊόν. Το μετρό που δείχνει κατά ποσό η ζητούμενη ποσότητα ενός αγαθού αντιδρά στις μεταβολές της τιμής του ονομάζεται ελαστικότητα ζήτησης, και υπολογίζεται διαιρώντας την ποσοστιαία μεταβολή της ζητούμενης ποσότητας ως προς την ποσοστιαία μεταβολή της τιμής. Δηλαδή η ελαστικότητα ζήτησης δείχνει ποσό διαθετιμενοι είναι οι καταναλωτές να μην αγοράσουν ένα αγαθό όταν αυξάνεται η τιμή του.

$$ed = \frac{\text{ποσοστιαια μεταβολη ζητουμενης ποσοστητα}}{\text{ποσοστιαια μεταβολη της τιμης}} = \frac{\frac{\Delta Q}{Q}}{\frac{\Delta P}{P}} = \frac{\Delta Q}{\Delta P} * \frac{Q}{P} < 0$$

Αν η ελαστικότητα ζήτησης σε απολυτή τιμή είναι μεγαλύτερη του ένα τότε λεμέ ότι είναι ελαστική δηλαδή μια αύξηση της τιμής κατά 5% θα έχει ως αποτέλεσμα την μείωση της ζητούμενης ποσότητας κατά 10% ή ανάποδα μια μείωση της τιμής κατά 5% θα έχει ως αποτέλεσμα την αύξηση της ζήτησης κατά 10%, από την άλλη πλευρά αν η ελαστικότητα ζήτησης σε απολυτή τιμή είναι μικρότερη του ένα τότε η ελαστικότητα ζήτησης ορίζεται ως ανελαστική αυτό σημαίνει ότι η ζητούμενη ποσότητα επηρεάζεται λιγότερο από μεταβολές της τιμής για παράδειγμα μια αύξηση της τιμής κατά 10% με ανελαστική ζήτηση θα έχει ως αποτέλεσμα την μείωση της ζητούμενης ποσότητας κατά 5% (ή και λιγότερο ή και περισσότερο ανάλογα με το αποτέλεσμα της πράξης), από την άλλη πλευρά μια μείωση της τιμής κατά 10% θα αυξήσει την ζήτηση μόνο κατά 5%.

Ένα παράδειγμα για να γίνει ακόμα πιο κατανοητή η ελαστικότητα ζήτησης είναι μέσο προϊόντων υψηλής ποιότητας ,για παράδειγμα η ζήτηση για αγορά αυτοκίνητου μάρκας Ferrari δεν θα επηρεαστεί σχεδόν καθόλου με μια μικρή αύξηση στην τιμή της καθώς το αυτοκίνητο δεν το αγοράζει κάποιος για να πηγαίνει απλά από το σημείο α στο σημείο β είναι ένα κοινωνικό σύμβολο , με υψηλές αποδόσεις και μεγάλο brand name, το ίδιο ισχύει και για τα κινητά της apple ή τα ρολόγια της Rolex κλπ. τα οποία έχουν ανελαστική

ζήτηση. Από την άλλη πλευρά προϊόντα όπως τα κινητά τηλεφώνα της Xiaomi ,τα αυτοκίνητα της Citroen που χαρακτηρίζονται κύριος ως value for money δηλαδή ότι προσφέρουν ικανοποιητική ποιότητα σε αντίστοιχα φυσιολογική τιμή με μια αύξηση της τιμής τους θα υπάρξει μεγάλη πτώση στην ζητούμενη ποσότητα τους καθώς η ελαστικότητα ως προς την τιμή είναι ελαστική. Προφανώς υπάρχουν και περιπτώσεις οπού η ελαστικότητα ζήτησης είναι τελείως ανελαστική και η καμπύλη ζήτησης είναι μια καθετή γραμμή που συμβολίζει ότι ανεξάρτητος τιμής οι καταναλωτές θα αγοράσουν το προϊόν σε μια συγκεκριμένη ποσότητα τέτοια προϊόντα είναι είδη πρώτης ανάγκης όπως για παράδειγμα τα φάρμακα. Ενώ από την άλλη πλευρά στην περίπτωση οπού η καμπύλη ζήτησης είναι μια ευθεία οριζόντια γραμμή δηλαδή η ελαστικότητα ζήτησης είναι τέλειος ελαστική αυτό σημαίνει ότι οι καταναλωτές σε μια συγκεκριμένη τιμή χ θα αγοράσουν όσο το δυνατόν μεγαλύτερη ποσότητα από το συγκεκριμένο προϊόν.

Πέρα όμως από την τιμή υπάρχουν και άλλοι προσδιοριστικοί παράγοντες οπού αυτή την φορά μετατοπίζουν πλήρως την καμπύλη ζήτησης είτε προς τα δεξιά είτε προς τα αριστερά. Πιο συγκεκριμένα ένας βασικός παράγοντας που επηρεάζει την ζήτηση είναι το εισόδημα των καταναλωτών, όταν το εισόδημα των καταναλωτών αυξάνεται έχει ως αποτέλεσμα η ζήτηση για κανονικά αγαθά όπως πχ τηλεοράσεις υπολογιστές αυτοκίνητα κλπ. να αυξάνεται ενώ όταν το εισόδημα μειώνεται , μειώνεται και η ζητούμενη ποσότητα για αυτά τα αγαθά, ενώ αυξάνεται η ζήτηση για κατωτέρα αγαθά των παραπάνω για παράδειγμα αντί για χρήση και αγορά νέου αυτοκίνητου οι καταναλωτές προτιμούν να μετακινηθούν με τα μέσα μαζικής μεταφοράς όπως το μετρό και το λεωφορείο , αντί για αγορά ενός νέου κινητού τηλεφώνου οι καταναλωτές προτιμούν την αγορά ενός μεταχειρισμένου. Ενώ από την άλλη πλευρά μια μείωση στο εισόδημα έχει ως αποτέλεσμα την μείωση της ζήτησης για κανονικά αγαθά. Βέβαια σημασία για τον εκάστοτε αναλυτή να βρει εκείνους τους παράγοντες που μεταβάλουν το εισόδημα των καταναλωτών , για παράδειγμα ένας τέτοιος παράγοντας είναι το επίπεδο ανεργίας στην οικονομία όσο περισσότερα άτομα εργάζονται τόσο μεγαλύτερο είναι τοι διαθέσιμο εισόδημα για κατανάλωση. Ένας άλλος παράγοντας που επηρεάζει το εισόδημα είναι το μορφωτικό επίπεδο των ατόμων και η εξειδίκευση τους καθώς όσο υψηλότερη είναι τόσο μεγαλύτερος είναι και ο μισθός των εργαζομένων. Προφανώς οι φόροι εισοδήματος οπού θέτουν τα κράτη μειώνουν το διαθέσιμο εισόδημα των καταναλωτών, Άλλοι μακροοικονομικοί παράγοντες που επηρεάζουν το εισόδημα των καταναλωτών είναι το ΑΕΠ μιας χωρά και

γενικότερα οι δημοσιονομικές δαπάνες του κράτους καθώς αυτές κινούν την οικονομία και δημιουργούνται νέες θέσεις εργασίας ένας άλλος παράγοντας είναι ο πληθωρισμός δηλαδή η ποσοστιαία αύξηση των τιμών του καλαθιού του καταναλωτή σε σχέση με προηγούμενες χρονικές περιόδους. Γενικότερα οι τιμές των προϊόντων συνήθως αυξάνονται λόγο αύξησης των τιμών των παραγωγικών συντελεστών στην εφοδιαστική αλυσίδα παραδείγματος χάρη αύξηση των μισθών των οδηγών των φορτηγών , που έχει ως αποτέλεσμα την αύξηση του μεταφορικού κόστους των εμπορευμάτων , ή η αύξηση της τιμής των πρώτων υλών , ή εφαρμογή δασμών στα εισαγόμενα ημιετιμα προϊόντα κλπ. Επίσης ακόμα και οι προσδοκίες των επιχειρήσεων που παράγουν τα προϊόντα μπορεί να επηρεάσουν την τιμή του προϊόντος για παράδειγμα αν οι επιχειρήσεις θεωρήσουν ότι η τιμή του προϊόντος θα αυξηθεί στο μέλλον τότε θα αυξήσουν την παραγωγική δυναμικότητα τους στο βραχυχρόνιο μέλλον.

Άλλοι δυο προσδιοριστικοί παράγοντες της ζήτησης είναι οι τιμές των υποκατάστατων και συμπληρωματικών αγαθών. Τα υποκατάστατα αγαθά είναι αγαθά που μπορούν να αντικαταστήσουν το ένα το άλλο για παράδειγμα πετρέλαιο θερμαστής και χρήση φυσικού αεριού για θερμαστή αν αυξηθεί η τιμή του ενός τότε θα αυξηθεί η ζήτηση για το άλλο αγαθό. Από την άλλη πλευρά τα συμπληρωματικά αγαθά είναι εκείνα τα οποία οι καταναλωτές όταν καταναλώνουν το ένα καταναλώνουν ταυτόχρονα και το άλλο , ένα από τα πιο τρανταχτά παραδείγματα αυτής της κατηγορίας είναι οι εκτυπωτές και τα μελανιά , οπού συνήθως τα καταστήματα πουλάνε τους εκτυπωτές σε τιμές ίσες ή και μικρότερες του κόστους ενώ στην συνέχεια δημιουργούν κέρδος από την πώληση των μελανιών του συγκεκριμένου μοντέλου εκτυπωτή , αλλά τέτοια παραδείγματα είναι τα κινητά τηλεφωνά και τα ακουστικά , οι τηλεοράσεις και οι ηχομπαρες , ο καφές και η ζάχαρη. Όταν η τιμή των συμπληρωματικών αγαθών αυξάνεται τότε μειώνεται η ζήτηση και για το προϊόν που αναλύουμε, ενώ αντίθετα όταν η τιμή του συμπληρωματικού μειώνεται αυξάνεται η ζήτηση του αλλού προϊόντος.

Επίσης σημαντικός προσδιοριστικός παράγοντας της ζήτησης είναι οι προσδοκίες των καταναλωτών για το μέλλον , αν οι καταναλωτές θεωρήσουν ότι η οικονομία θα με μπει σε ύφεση και κατά επέκταση θα μειωθεί το εισόδημα τους σε αυτήν την περίπτωση προτιμούν να αποταμιεύσουν τα χρήματα τους και να μειώσουν την κατανάλωση. Επίσης

όσον αφορά τις τιμές των προϊόντων αν οι καταναλωτές θεωρούν ότι η τιμή του προϊόντος θα μειωθεί για παράδειγμα υπάρχει το Samsung s24 στην αγορά και αναμένεται να κυκλοφορήσει το Samsung s25 τότε οι καταναλωτές αποφεύγουν αν αγοράσουν το πρώτο σήμερα και θα περιμένουν την κυκλοφορία του δευτέρου για να πέσει η τιμή του Samsung s24. Το ίδιο συμβαίνει και στην περίπτωση οπού οι καταναλωτές αναμένουν ότι θα υπάρξει έλλειψη των προϊόντων (πχ λόγο της πανδημίας covid που είχε ως αποτέλεσμα την διακοπή της λειτουργίας εργοστασίων παραγωγής κινητών τηλέφωνων στην κίνα) στο βραχυχρόνιο μέλλον και αρά η τιμή του προϊόντος αναμένεται να αυξηθεί τότε θα επισπεύσουν την αγορά του προϊόντος.

Ένας άλλος προσδιοριστικός παράγοντας της ζήτησης είναι οι προτιμήσεις των καταναλωτών. Οι προτιμήσεις των καταναλωτών επηρεάζουν σε μεγάλο βαθμό την ζήτηση για τελικά προϊόντα και υπηρεσίες , και μεταβάλλονται κύριος λόγο της τεχνολογικής εξέλιξης για παράδειγμα η ζήτηση για φωτογραφικές μηχανές έχει ελαττωθεί και μειωθεί στην επαγγελματική μόνο χρήση με την είσοδο στην αγορά των κινητών τηλεφώνων. Οι προτιμήσεις τον καταναλωτών επηρεάζονται επίσης από την κοινωνία και την κοινωνική ομάδα που ανήκει ο εκάστοτέ καταναλωτής , αυτό έχει ως αποτέλεσμα οι αγοραστικές του αποφάσεις να επηρεάζονται από την μάρκα προϊόντων που αγοράζει το κοντινό του κοινωνικό περιβάλλον. Επιπρόσθετα η μόδα κάθε έτος αλλάζει μαζί με αυτήν και οι προτιμήσεις του εκάστοτε καταναλωτή. Τέλος ο εκάστοτε καταναλωτής είναι διαφορετικός , δηλαδή διαφορετικές προτιμήσεις έχουν οι άνδρες από τις γυναικείες , διαφορετικές προτιμήσεις έχουν τα άτομα ηλικίας 20 ετών , και διαφορετικές προτιμήσεις τα άτομα μεγαλύτερης ηλικίας , επίσης το κάθε άτομα έχει διαφορετικό lifestyle. Βέβαια πολλές φορές στην σύγχρονη εποχή το lifestyle επηρεάζεται από τα social media τους influencers και συνολικά τις διαφημίσεις ενός προϊόντος. Γενικά μια αύξηση στις δαπάνες για την προβολή ενός προϊόντος έχει ως αποτέλεσμα την αύξηση της ζήτησης του.

Ένας ακόμα προσδιοριστικός παράγοντας της ζήτησης είναι οι καιρικές συνθήκες και η εποχικότητα, για παράδειγμα οι ενοικιάσεις αυτοκίνητων στα νησιά του αιγαίου αυξάνονται κατά την περίοδο του καλοκαιριού ενώ το χειμώνα είναι σχεδόν μηδενικές. Με την ιδιά λογική η κατανάλωση πετρελαίου θερμαστής το χειμώνα είναι υψηλότερη σε σχέση με την κατανάλωση για την ιδιά χρονική περίοδο το καλοκαίρι. Η έννοια της εποχικότητας αναλύεται διεξοδικά στο κεφάλαιο 7.

Βέβαια η ζήτηση για ένα προϊόν δεν είναι πάντα η ιδιά αλλά μεταβάλετέ ανάλογος την φάση του κύκλου ζωής του προϊόντος. Πιο συγκεκριμένα υπάρχουν τέσσερεις φάσεις στο κύκλο ζωής ενός προϊόντος η εισαγωγή, , η ανάπτυξη , η ωρίμανση , και τέλος η παρακμή. Στην φάση της εισαγωγής συνήθως η ζήτηση για το νέο προϊόν είναι περιορισμένη και χαρακτηρίζεται από υψηλή αβεβαιότητα, καθώς το προϊόν είναι άγνωστο στους καταναλωτές της εκάστοτε αγοράς που απευθύνεται. Ο τρόπος με τον οποίο οι επιχειρήσεις μπορούν αν αυξήσουν την ζήτηση για το προϊόν είναι μέσο δαπανών για την προώθηση του , ενώ οι τρόποι ανάλυσης της ζήτησης είναι περιορισμένοι σε ποιοτικές μεθόδους όπως την Delphi καθώς δεν υπάρχουν ιστορικά δεδομένα. Για αυτό το λόγο η επιχείρηση θα πρέπει να χρησιμοποιήσει μια ευέλικτη εφοδιαστική αλυσίδα , που θα μπορεί να ανταποκριθεί και σε μεγάλες αυξήσεις της ζήτησης αλλά ταυτόχρονα δεν θα πρέπει να είναι πολύ κοστοβορα. Στην φάση της ανάπτυξης το προϊόν πλέον είναι γνωστό προς τους καταναλωτές και η ζήτηση για αυτό αυξάνεται με πολύ γρήγορο ρυθμό , σε αυτήν την φάση η επιχείρηση πρέπει να ορίσει την κατάλληλη τιμή , και να αυξήσει την παραγωγή του προϊόντος μέσα από επενδύσεις σε εργοστάσια, μέσα μεταφοράς και αποθήκες, ενώ ταυτόχρονα θα πρέπει να ανταπεξέλθει σε τυχόν εισόδους στην αγορά άλλων αντίστοιχων ανταγωνιστικών προϊόντων. Τέλος σε αυτό το στάδιο η επιχείρηση μπορεί να αναλύσει σε καλύτερο βαθμό την ζήτηση καθώς οι αναλύσεις στηρίζονται σε ποσοτικά δεδομένα. Στο στάδιο της ωρίμασης η ζήτηση πλέον σταθεροποιείται και αυξάνεται με πολύ μικρό ποσοστό σε σχέση με τις πωλήσεις των προηγουμένων περιόδων ,καθώς οι περισσότεροι καταναλωτές έχουν ήδη αγοράσει το προϊόν. Αυτή η φάση χαρακτηρίζεται από έντονο ανταγωνισμό που έχει ως αποτέλεσμα να μειώνονται οι τιμές πωλήσεις και κατά επέκταση να συρρικνώνονται τα κέρδη ένας τρόπος για την αποφυγή του παραπάνω είναι η διαφοροποίηση του προϊόντος , μέσο της τεχνολογικής ανάπτυξης, και του brand name όπως για παράδειγμα γίνεται στην αγορά των κινητών τηλέφωνων και των αυτοκίνητων. Τέλος στην φάση της παρακμής οι καταναλωτές στέφονται σε νέα προϊόντα με αποτέλεσμα η ζήτηση να περιορίζεται σε ένα πολύ μικρό τμήμα πιστών πελατών. Η επιχείρηση σε αυτό το στάδιο θα πρέπει να μειώσει όσο τον δυνατό περισσότερο τα λειτουργικά της κόστη , όπως το κόστος αποθήκευσης , και η ανάλυση της ζήτησης θα πρέπει να εστιαστεί για το αν θα πρέπει να συνεχιστεί η παραγωγή και πώληση του προϊόντος στην αγορά ή θα πρέπει να αποσυρθεί τέλειος από αυτήν.

1.3 ΤΡΟΠΟΙ ΔΗΜΙΟΥΡΓΕΙΑΣ ΠΡΟΒΛΕΨΕΩΝ

Σε αυτήν την ενότητα θα αναλύσουμε εν συντομία τις μεθόδους δημιουργίας προβλέψεων καθώς στο κεφάλαιο 4 αναλύεται η θεωρία πίσω από τα ποσοτικά μοντέλα μηχανικής μάθησης ενδελεχώς , ενώ στο κεφάλαιο 7 και 9 αναπτύσσονται αλγόριθμοι στην python για τον προσδιορισμό τις μελλοντικής τιμής του κινητού Samsung galaxy s24(κεφάλαιο 7) και των εξαμηνιαίων πωλήσεων του ομίλου πλαίσιο(κεφάλαιο 9).

Οι μέθοδοι δημιουργίας προβλέψεων χωρίζονται σε δυο βασικές κατηγορίες τις ποιοτικές και τις ποσοτικές. Από την μια πλευρά οι ποσοτικές μέθοδοι δημιουργούν προβλέψεις βασιζόμενες σε αριθμητικά δεδομένα και μέσο της χρήσης μαθηματικών στατιστικής και αλγορίθμων. Ενώ από την άλλη πλευρά οι ποιοτικές μέθοδοι χρησιμοποιούνται όταν δεν υπάρχουν επαρκή αριθμητικά δεδομένα για αυτό τον λόγο βασίζονται κύριος στην γνώση , εμπειρία διαίσθηση , και γενικότερα την κρητική ικανότητα των στελεχών μιας εταιρίας για την δημιουργία προβλέψεων για το μέλλον.

Η πιο συνήθης ποσοτική μέθοδος δημιουργίας προβλέψεων είναι μέσο από την στατιστική και την επιστήμη των υπολογιστών τα λεγόμενα data analytics. Σε αυτά ο αναλυτής αρχικά συλλεγεί τα δεδομένα μέσα από την βάση δεδομένων της εταιρείας οπού εργάζεται ,το ιντερνέτ και από άλλες τρίτες επιχειρήσεις πχ από κάποιον συνεργάτη προμηθευτή της εταιρείας , εν συνέχεια αναλύει τις μεταβλητές μια προς μια και συμπληρώνει τυχών ελλείπεις τιμές , τέλος τοποθετεί εκείνες της μεταβλητές που θεωρεί σημαντικές στην δημιουργία των προβλέψεων σε διαφορά μοντέλα όπως για παράδειγμα στην γραμμική παλινδρόμηση. Ο τρόπος ανάλυσης που αναφέρθηκε παραπάνω χρησιμοποιείται κύριος στα αιτιατά μοντέλα προβλέψεων όπως τα οικονομετρικά οπού εκεί υπάρχει σύνδεση μεταξύ εξαρτημένης μεταβλητής και ερμηνευτικής για παράδειγμα αν κάποιος θέλει να αναλύσει τις ημερήσιες πωλήσεις των παγωτών μιας εταιρείας (εξαρτημένη μεταβλητή) μπορεί να χρησιμοποιήσει ως ερμηνευτικές μεταβλητές την μέση θερμοκρασία μέσα στην ημέρα , την τιμή του παγωτού , το ΑΕΠ, τον πληθυσμό της χωράς, τις προτιμήσεις των καταναλωτών κλπ. Το θετικό αυτών των μοντέλων είναι ότι υπάρχει ερμηνεία μεταξύ των μεταβλητών , παραδείγματος χάρη μια αύξηση της θερμοκρασίας κατά 1 βαθμό κελσίου αυξάνει την ζήτηση κατά 1000 τεμάχια. Από την άλλη πλευρά μοντέλα όπως τα

αυτοπαλινδρομα , ο κινητός μέσος ορός κλπ. ανήκουν στην κατηγορία των μη αιτιατών μοντέλων καθώς οι προβλέψεις για τις επόμενες τιμές τις μεταβλητής είναι αποτέλεσμα της ανάλυσης των προηγουμένων τιμών αυτής (ιστορικά δεδομένα). Αυτές οι μέθοδοι είναι πολύ χρήσιμες στην εφοδιαστική αλυσίδα καθώς μέσο αυτών μπορεί να γίνει πρόβλεψη για τις μελλοντικές πωλήσεις και κατά επέκταση η εταιρεία μπορεί να διαχειριστεί με τον πλέον βέλτιστο τρόπο τα αποθέματα της αποφεύγοντας ελλείψεις ή υπερ. αποθεματοποίηση. Επιπλέον μέσα από αυτές τις μεθόδους μπορεί να επιτευχθεί η βέλτιστη τιμολόγηση του προϊόντος καθώς αυτοί οι αλγόριθμοι βοηθούν τον αναλυτή να βρει εκείνα τα χαρακτηριστικά των προϊόντων που προτιμούν οι καταναλωτές και κατά επέκταση είναι διαθετιμενοι να πληρώσουν παραπάνω για την απόκτηση τους (αναλυτικό παράδειγμα γίνεται στο κεφαλαίο 8) , και όπως αναφέραμε και στην ενότητα 1.2 η τιμή είναι ένας από τους βασικότερους προσδιοριστικούς παράγοντες της ζήτησης. Στην κατηγορία των data analytics ανήκουν επίσης οι μέθοδοι των νευρωνικων δικτυών εν συντομία καθώς και αυτοί αναλύονται διεξοδικά στο κεφάλαιο 7 είναι αλγόριθμοι οπού λαμβάνουν ως είσοδο την εξαρτημένη και τις ερμηνευτικές μεταβλητές στην συνέχεια τις αναλύουν και εξάγουν προβλέψεις για την εξαρτημένη μεταβλητή στην συνέχεια υπολογίζουν ένα σφάλμα μεταξύ των προβλέψεων και των πραγματικών τιμών , ενημερώνονται , και εκτελούν την ιδιά διαδικασία πολλές φορές μέχρι να ελαχιστοποιηθεί το σφάλμα αυτό ονομάζεται εκπαίδευση του νευρωνικου δικτύου. Το θετικό της χρήσης των νευρωνικων δικτυών είναι ότι μπορούν να αντιληφθούν την τάση και την εποχικότητα που υπάρχει μέσα στα δεδομένα ,ότι μπορούν να χρησιμοποιηθούν σε μη γραμμικά προβλήματα και ότι μπορούν να εισαχθούν σε αυτά πολλές εξωγενείς ερμηνευτικές μεταβλητές. Τα αρνητικά της χρήσης τους είναι ότι απαιτείται πολύς χρόνος και μεγάλη υπολογιστική ισχύ για την εκπαίδευση τους , και ότι οι προβλέψεις που εξάγουν είτε είναι καλές είτε όχι δεν μπορούν να ερμηνευθούν όπως για παράδειγμα στα μοντέλα της γραμμικής παλινδρόμησης.

Μια άλλη μέθοδος δημιουργίας προβλέψεων είναι εκείνη των προσομοιώσεων. Οι προσομοιώσεις είναι τεχνικές οπού μπορεί να γίνει αναπαράσταση πραγματικών συμβάντων σε ψηφιακό χώρο με την χρήση των μαθηματικών και των υπολογιστών για παράδειγμα η tesla εκπαιδεύει τις νέες εκδόσεις του autopilot των αυτοκινήτων της όχι στους πραγματικούς δρόμους αλλά σε ψηφιακούς. Με τον ίδιο ακριβός τρόπο μπορεί ο αναλυτής να δημιουργήσει ένα επιχειρησιακό μοντέλο σε ψηφιακό περιβάλλον και να

πειραματιστεί με διαφορά σενάρια χωρίς να επηρεάζει την πραγματική λειτουργεία της επιχείρησης για παράδειγμα τι θα συμβεί αν σε ένα δρομολόγια παραδοχής προϊόντων το φορτηγό επιλέξει άλλη διαδρομή από την συνηθισμένη δηλαδή κατά ποσό θα μειωθεί ή θα αυξηθεί ο χρόνος παράδοσης , και ποσό καύσιμο θα εξοικονομηθεί ,τι θα συμβεί αν προσδεθεί ένα επιπλέον σημείο παράδοσης ,τι θα συμβεί στα αποθέματα αν χρησιμοποιηθεί μέθοδος fifo αντί για lifo , τι θα συμβεί αν προσληφθούν περισσότεροι pickers δηλαδή ποσό θα αυξηθούν τα κόστη και ποσό θα μειωθεί ο χρόνος συγκέντρωσης των προϊόντων ανά picking list , τι θα συμβεί αν η εταιρεία αγοράσει δυο ή τρία ή και παραπάνω παλετοφορα , πόσους νέους πελάτες θα εξυπηρετήσει το άνοιγμα ενός νέου καταστήματος σε μια άλλη γεωγραφική περιοχή, τι θα συμβεί αν επιλεγεί άλλος προμηθευτής με διαφορετικούς χρόνους παράδοσης δηλαδή αντί για την προμήθεια ενός προϊόντος από την κίνα γίνει η προμήθεια του από έναν προμηθευτή στην Ελλάδα κατά ποσό θα αυξηθούν ή θα μειωθούν τα μεταφορικά κόστη , ποσό θα αυξηθεί ή θα μειωθεί ο χρόνος παράδοσης της παραγγελίας , ποσό καλύτερη η χειρότερη θα είναι η ποιότητα του προμηθευομένου προϊόντος και κατά ποσό αυτή θα επηρεάσει το τελικό προϊόν. Προφανώς οι προσομοιώσεις μπορούν να εφαρμοστούν και για την αποφυγή και διαχείριση κίνδυνων , για παράδειγμα τι θα συμβεί εάν γίνει κάποια απεργία στην χωρά του προμηθευτή πρώτων υλών και κατά ποσό αυτή θα επηρεάσει την παραγωγική διαδικασία , υπάρχει περίπτωση να ξεμείνει από αποθέματα η εταιρεία και να χάσει πωλήσεις , τι θα συμβεί αν χαλάσει ένα φορτηγό ή ένα μηχάνημα παραγωγής. Όλα τα παραπάνω ερωτήματα μπορούν να απαντηθούν μέσα από τις προσομοιώσεις. Ένας από τους πιο διαδεδομένους αλγορίθμους προσομοίωσης είναι ο monte Carlo ο οποίος βασίζεται στην χρήση πιθανοτήτων και τυχαίων αριθμών για την επίλυση πραγματικών προβλημάτων. η συγκεκριμένη μέθοδος επιλύει τα προβλήματα μέσο της επανάληψής της δειγματοληψίας , για παράδειγμα έστω ότι κάποιος αναλυτής δεν γνωρίζει ότι η πιθανότητα εμφάνισης του αριθμού 4 από την ρίψη ενός ζαριού είναι ίση με 1/6 (μια είναι η πλευρά του ζαριού που έχει τον αριθμό 4 ενώ συνολικά το ζάρι μπορεί να λάβει 6 διαφορετικές τιμές), θα μπορούσε χρησιμοποιώντας ένα πρόγραμμα όπως το excel ή η python και να ζητήσει μια μεταβλητή να παίρνει τυχαίες ακέραιες τιμές από το 1 έως το 6 στην συνέχεια να δημιουργήσει μια συνάρτηση if δηλαδή εάν ο τυχαίος αριθμός πάρει την τιμή 4 να εισαχθεί μέσα σε μια άλλη μεταβλητή με αθροίσματα ο αριθμός 1 διαφορετικά εάν η τυχαία μεταβλητή πάρει μια τιμή διαφορετική από το 4 να μπει στην μεταβλητή με τα αθροίσματα η τιμή 0 , αυτή η διαδικασία επαναλαμβάνεται πολλές φορές πχ 10.000

φορές στην συνέχεια ο αναλυτής πρέπει να πάρει την μεταβλητή με τα αθροίσματα και να την διαιρέσει με τον αριθμό των ρίψεων το αποτέλεσμα που θα προκύψει είναι πολύ κοντά στην πραγματική πιθανότητα 1/6. Ένα παράδειγμα του monde Carlo στα logistics θα μπορούσε να είναι η προσομοίωση του χρόνου καθυστέρησης των φορτηγών κατά την επιστροφή τους στην εταιρεία , πιο συγκεκριμένα θα μπορούσαν να επιλέγουν μεταβλητές όπως οι καιρικές συνθήκες ,η πιθανότητα να γίνει κάποιο ατύχημα στον δρόμο , και άλλες όπως ο αριθμός των χρηστών του δρόμου , αυτές οι μεταβλητές μπορούν να πάρουν τυχαίες τιμές ή να προσδιοριστούν με βάση ιστορικά δεδομένα της επιχείρησης από προηγούμενα δρομολόγια με τις αντίστοιχες πιθανότητες εμφάνισης του καθενός. Στην συνέχεια εκτελείται ο αλγόριθμος πολλές χιλιάδες φορές και ο αναλυτής καταλήγει σε ένα μέσο χρόνο καθυστέρησης και στην συνέχεια μπορεί να εξάγει συμπεράσματα για τα έξοδα όπως για παράδειγμα η αυξημένη κατανάλωση καυσίμου.

Μια ακόμα ποσοτική μέθοδος για την δημιουργία προβλέψεων είναι εκείνη των προδρόμων δεικτών. Οι πρόδρομοι δείκτες είναι μεταβλητές οι οποίες μεταβάλλονται ανοδικά ή πτωτικά και έχουν ως αποτέλεσμα την μεταβολή της μεταβλητής που ο αναλυτής εξετάζει , μοιάζουν αρκετά με τις γραμμικές παλινδρομήσεις καθώς και αυτό το μοντέλο βασίζεται στα ιστορικά δεδομένα αλλά κύριος στην εμπειρική παρατήρηση , η διαφορά τους με την γραμμική παλινδρόμηση είναι ότι οι μεταβλητές δεν κινούνται ταυτόχρονα αλλά πρώτα επηρεάζεται η ερμηνευτική μεταβλητή και στην συνέχεια συνήθως μεταβάλετέ και η εξαρτημένη μεταβλητή. Για παράδειγμα αν για κάποιον λόγο το εισόδημα των καταναλωτών μειωθεί αυτό θα έχει ως συνέπεια την μείωση της κατανάλωσης και κατά επέκταση της ζήτησης για προϊόντα. Επίσης αν για μια χρονική περίοδο αυξηθούν οι παραγγελίες για ένα προϊόν αναμένεται αυτές να συνεχίσουν να αυξάνονται , με τον ίδιο τρόπο αν παρατηρηθεί ότι το απόθεμα ενός προϊόντος μειώνεται με γρήγορο ρυθμό τότε αυτό αναμένεται να συνεχιστεί και αρά η εταιρεία προβαίνει σε κατασκευή νέας παρτίδας. Ένας άλλος πρόδρομος δείκτης είναι εκείνος του κόστους μεταφορών και των πρώτων υλών αν για παράδειγμα το κόστος μεταφοράς ενός προϊόντος από την κίνα προς την Ελλάδα αυξηθεί τότε αυτό θα έχει ως αποτέλεσμα την αύξηση της τιμής του προϊόντος που αυτή στην συνέχεια θα οδηγήσει σε μείωση της συνολικής ζήτησης.

Όσον αφορά τις ποιοτικές μεθόδους για την δημιουργία προβλέψεων η πιο ευρέος γνωστή είναι η μέθοδος Delphi. Αυτή η μέθοδος χρησιμοποιείται κυρίως για την δημιουργία μακροπροθέσμων προβλέψεων σε μεταβλητές οπού δεν υπάρχουν προηγούμενα ιστορικά δεδομένα και ως αποτέλεσμα οι απόψεις των στελεχών και συνεργατών της εταιρίας διαδραματίζουν τον σημαντικότερο ρολό για την λήψη μιας απόφασης. Πιο συγκεκριμένα η μέθοδος Delphi είναι μια δομημένη διαδικασία επαναλαμβανομένων ερωτήσεων προς μια ομάδα στελεχών με στόχο την προσέγγιση μια κοινής γνώμης (εκτίμησης) για την μελλοντική έκβαση μιας μεταβλητής. Αναλυτικά τα βήματα τις διαδικασίας είναι τα παρακάτω , αρχικά πρέπει να επιλεχθούν εκείνα τα άτομα τα οποία έχουν γνώσεις και εμπειρία πάνω στο θέμα που θέλουμε να αναλύσουμε ή να προβλέψουμε για παράδειγμα αν θέλουμε να προσδιορίσουμε αν η ζήτηση των αυτοκίνητων της Toyota θα αυξηθεί στην Ελλάδα θα πρέπει να επιλέγουν άτομα τα οποία έχουν γνώσεις πάνω στην ελληνική οικονομία και στις καταναλωτικές συνήθεις των Ελλήνων δηλαδή τι είδους αυτοκίνητα προτιμούν οι Έλληνες αγοραστές SUV, coupe ,sedan , μικρά SUV κλπ. Στην συνέχεια αποστέλλεται στους ειδικούς ένα ερωτηματολόγιο με διάφορες ερωτήσεις αναφορικά με τις μελλοντικές τάσεις και τα ενδεχόμενα σενάρια έκβασης του αποτελέσματος. Οι απαντήσεις των ερωτηματολογίων συλλέγονται και αναλύονται από άτομα τρίτα που δεν ανήκουν στην ομάδα των ειδικών , εν συνέχεια οι απαντήσεις του κάθε ειδικού δίνονται ανώνυμα στα υπόλοιπα μέλη των ειδικών οπού τους δίνεται η δυνατότητα να αλλάξουν την απάντηση τους με βάση τις απαντήσεις των άλλων μέλλων. Αυτή η διαδικασία επαναλαμβάνεται μέχρις ότου οι απαντήσεις των ειδικών να συγκλίνουν μεταξύ τους. Η μεθάς Delphi έχει αποδειχτεί πολύ χρήσιμη για την πρόβλεψη της ζήτησης σε προϊόντα που κάνουν την πρώτη τους εμφάνιση στην αγορά και αρά δεν υπάρχουν προηγούμενα δεδομένα πωλήσεων , για την χρήση ή μη νέων τεχνολογιών μέσα στην εφοδιαστική αλυσίδα για παράδειγμα αντικατάσταση των picker με robot , και για τον σχεδιασμό μακροχρόνιων στρατηγικών παραδείγματος χάρη άνοιγμα νέου εργοστασίου συναρμολόγησης του τελικού προϊόντος στην κίνα.

Μια μέθοδος που συνδικάζει τις ποιοτικές και τις ποσοτικές μεθόδους είναι εκείνη της ερευνάς αγοράς. Οι έρευνες αγοράς συνηθώς εφαρμόζονται για την δημιουργία προβλέψεων για την ζήτηση ενός νέου προϊόντος που πρόκειται να κυκλοφορήσει στην αγορά. Υπάρχουν διάφοροι τρόποι εκτέλεσης αυτής της μεθόδου όπως για παράδειγμα μέσο δημοσκοπήσεων στο διαδίκτυο , η μέσο ανάλυσης στατιστικών από κοινωνικά

δίκτυα, αλλά ο ποιο γνωστός τρόπος είναι μέσο ερωτηματολογίων. Στο ερωτηματολόγιο τις περισσότερες φορές υπάρχουν ερωτήσεις αναφορικά με το φύλο , την ηλικία , το εισόδημα, το μορφωτικό επίπεδο κλπ. στην συνέχεια υπάρχουν ερωτήσεις για τις προτιμήσεις του καταναλωτή , συνήθως αυτές οι ερωτήσεις έχουν απαντήσεις σε κλίμακα ή είναι διχοτομημένες δηλαδή υπάρχουν μόνο δυο απαντήσεις ναι και όχι . στην συνέχεια πρέπει να επιλεχθούν εκείνες οι ομάδες ατόμων που προσδοκάει η εταιρεία να πουλήσει το νέο της προϊόν για παράδειγμα αν το ερωτηματολόγιο έχει ερωτήσεις αναφορικά με την γεύση και την προτίμηση ενός αναψυκτικού όπως τα red bull η ομάδα των ατόμων που θα επιλεγεί θα πρέπει να είναι άτομα μικρής και μέσης ηλικίας , και οι χώροι που μπορεί να απαντηθούν τα ερωτηματολόγια θα μπορούσαν να είναι εκδηλώσεις με extreme sports ή να αποστέλλεται σε συνδρομητές του newsletter της εταιρείας και τα άτομα που θα το απαντήσουν να τους αποστέλλεται ένας κωδικός έκπτωσης για την επόμενη αγορά τους ώστε να υπάρχει κίνητρο συμπλήρωσης του. Τέλος εφόσον τα ερωτηματολόγια έχουν απαντηθεί το τμήμα μάρκετινγκ εξάγει διαφορά στατιστικά διαγράμματα όπως πίτες και ιστογράμματα και με βάση αυτά προκύπτει η απόφαση για την κυκλοφορία του νέου προϊόντος.

2. ΕΙΣΑΓΩΓΗ ΣΤΗΝ ΣΤΑΤΙΣΤΙΚΗ

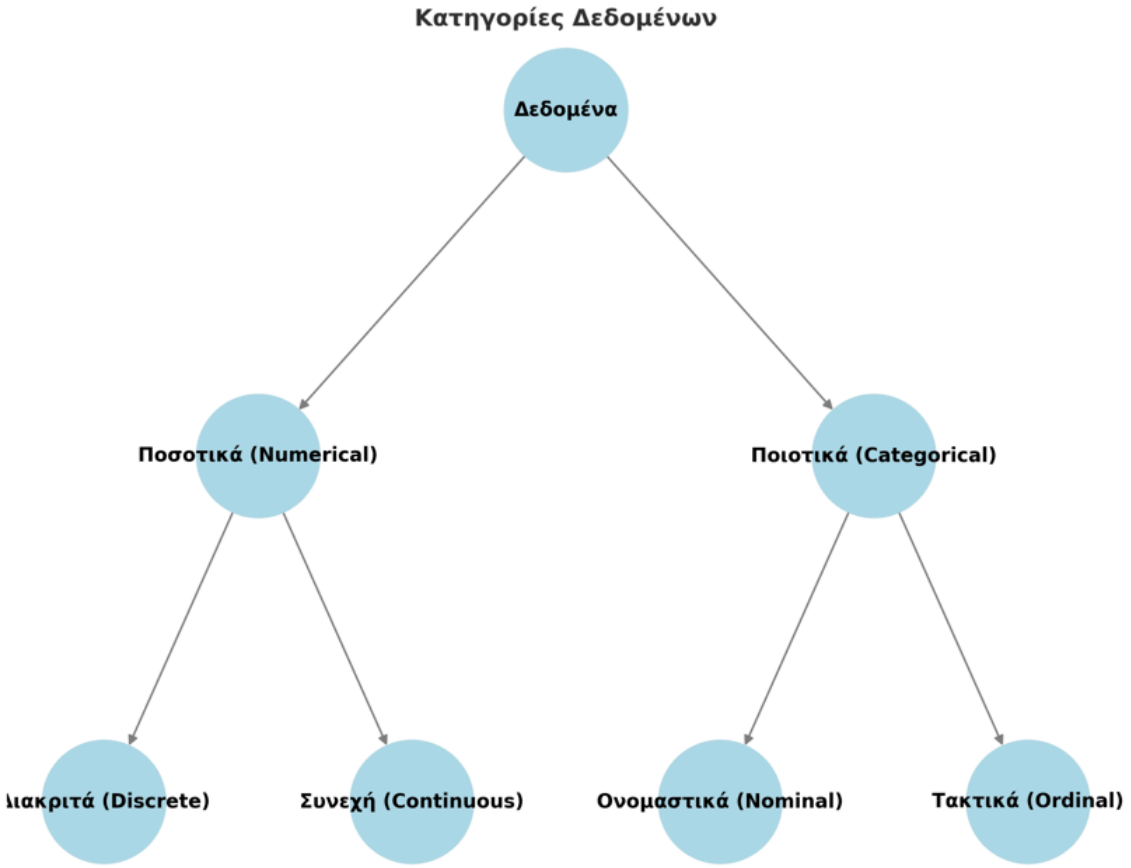
Σε αυτό το κεφάλαιο γίνεται σύντομη αναφορά σε βασικές μεθόδους περιγραφικής στατιστικής όπως για παράδειγμα τους τύπους των δεδομένων , στατιστικά μετρά όπως τον μέσο ορό την διακύμανση , την τυπική απόκλιση το skewness ,επιπλέον γίνεται ανάλυση στα είδη των διαστημάτων εμπιστοσύνης , και στους ελέγχους υποθέσεων . Όλες οι παραπάνω μέθοδοι παρότι φαίνονται απλοί και εύκολοι στην κατανόηση είναι πολύ χρήσιμοι στην ανάλυση της ζήτησης στην εφοδιαστική αλυσίδα παραδείγματος χάρη με την χρήση διαστημάτων εμπιστοσύνης μπορεί ο εργαζόμενος στο τμήμα αγορών μιας εταιρείας λιανικών πωλήσεων να αποφασίσει ποσά προϊόντα να παραγγείλει με βάση τα ιστορικά δεδομένα των πωλήσεων του συγκεκριμένου προϊόντος το προηγούμενο έτος. Επίσης οι ελέγχει υποθέσεων είναι πολύ σημαντικοί για θέματα μάρκετινγκ και των πωλήσεων καθώς ο αναλυτής μπορεί να ελέγξει αν μια ομάδα προϊόντων έχει περισσότερές πωλήσεις σε σχέση με μια άλλη ή να κάνει αναλύσεις με βάση της εξαμηνιαίες πωλήσεις , ή αν ένα κατάστημα λιανικών πωλήσεων της εταιρίας έχει περισσότερες πωλήσεις σε σύγκριση με ένα άλλο της ιδίας εταιρίας σε μια άλλη γεωγραφική περιοχή.

2.1 ΚΑΤΗΓΟΡΙΕΣ ΔΕΔΟΜΕΝΩΝ

Τα στατιστικά δεδομένα χωρίζονται σε δυο μεγάλες κατηγορίες τα ποιοτικά δεδομένα και τα ποσοτικά δεδομένα , και κάθε μια από αυτές τις δυο κατηγορίες δεδομένων έχει και τις αντίστοιχες δίκες της υποκατηγορίες. Πιο συγκεκριμένα μια κατηγορία των ποιοτικών δεδομένων είναι τα ονομαστικά δεδομένα τα οποία δεν έχουν κάποια ιεραρχία μεταξύ τους για παράδειγμα στα δεδομένα που θα αναλύσουμε στο κεφάλαιο 6 υπάρχουν δεδομένα όπως το φύλο των πελατών που παίρνει τιμές άνδρας και γυναικά αντίστοιχα , άλλες τέτοιες κατηγορίες δεδομένων είναι το χρώμα ενός κινητού τηλεφώνου που πωλείται σε ένα κατάστημα λιανικών πωλήσεων πχ πράσινο άσπρο μπλε , λόγο της φύσης των δεδομένων δεν μπορούμε να τα ταξινομήσουμε δηλαδή ότι το κόκκινο χρώμα είναι καλύτερο από το μπλε χρώμα παρόλα αυτά μπορούμε να τις ομαδοποιήσουμε και να τις μετατρέψουμε σε ψευδομεταβλητες δηλαδή σε τιμές που όταν ισχύει κάτι πχ έστω στα δεδομένα με το φύλο αν είναι άνδρας θα μπει τιμή 1 και αν είναι γυναικά τιμή 0 με αυτό τον τρόπο αυτά τα δεδομένα μπορούν να μελετηθούν σε αναλύσεις όπως ελέγχους αποφάσεων(πχ οι γυναίκες πελάτες αγοράζουν περισσότερα προϊόντα ομορφιάς από ότι οι

άντρες πελάτες) και να μοντελοποιηθούν σε μοντέλα μηχανικής μάθησης για παλινδρόμηση ή για ταξινόμηση. Στην κατηγορία των ποιοτικών δεδομένων ανήκουν επίσης τα διατακτικά δεδομένα δηλαδή τιμές οι οποίες έχουν μεταξύ τους μια ιεραρχία αλλά δεν μπορεί αυτή να αποδοθεί αριθμητικά , ένα απλό παράδειγμα αυτής της κατηγορίας δεδομένων είναι το μέγεθος μιας μπλούζας για παράδειγμα μικρό μεσαίο μεγάλο μέγεθος ή το ποσό ικανοποιητική ήταν η εμπειρία ενός πελάτη μέσα στο κατάστημα για παράδειγμα άσχημη ,μέτρια ,πάρα πολύ καλή , άλλο παράδειγμα είναι οι διάφοροι επεξεργαστές φορητών υπολογιστών intel I5 6500 , intel I5 7500, intel I5 8500, προφανώς ο τελευταίος έχει περισσότερη υπολογιστική ισχύ από τον πρώτο που αναφέρθηκε αλλά πρέπει να χρησιμοποιηθούν και αλλά δεδομένα για να φανερωθεί η διαφορά που ίσως δημιουργείτε στην τιμή του εκάστοτε υπολογιστή. Και αυτά τα δεδομένα παρότι μπορούμε να τα ιεραρχήσουμε για να εισέρθουν σε ένα μοντέλο μηχανικής μάθησης πρέπει πρώτα να μετατραπούν σε ψευδομεταβλητες.

Όσον αφορά τα ποσοτικά δεδομένα αυτά είναι αριθμοί οι οποίοι μπορούν να μετρηθούν με ακρίβεια και χωρίζονται σε δυο υποκατηγορίες. Η μια εξ αυτών είναι τα διακριτά δεδομένα τα οποία περνούν συγκεκριμένες (το εύρος τιμών είναι συνήθως γνωστό) και ακέραιες τιμές για παράδειγμα τα παιδεία που έχει ένας πελάτης 1,2,3 , ο βαθμός ενός φοιτητή στις εξετάσεις από 1 έως 10 , οι αριθμοί στο κινο κλπ. Η δεύτερη υποκατηγορία η οποία είναι και η πιο συνήθης είναι τα συνεχή αριθμητικά δεδομένα τα οποία περνούν και ακέραιες και δεκαδικές τιμές χωρίς κάποιο εύρος και είναι συνηθώς αποτέλεσμα μέτρησης για παράδειγμα οι παλμοί ενός αθλητή που καταγράφονται από ένα έξυπνο ρολόι κατά την διάρκεια της προπόνησης του ή θερμοκρασία ενός φορτηγού ψυγείου , οι μισθοί των πελατών ενός καταστήματος , το βάρος μιας παλέτας , ενός φορτιού, η τιμή ενός κινητού τηλεφώνου , η τιμή μιας μετοχής κλπ., προφανώς υπάρχουν διάφορες μεταξύ των παραπάνω καθώς μερικά δεδομένα είναι χρονοσειρες όπως η θερμοκρασία του φορτηγού ψυγείου σε σχέση με το βάρος μιας παλέτας το οποίο δεν είναι χρονοσειρα. Εν συντομία (καθώς οι χρονοσειρες και τα χαρακτηριστικά τους αναλύονται διεξοδικά στο κεφάλαιο 7) χρονοσειρα είναι ένα σύνολο δεδομένων οπού οι τιμές του είναι διατεταγμένες χρονικά και η σειρά με την οποία είναι τοποθετημένες έχει σημασία , επίσης η κάθε τιμή έχει κάποια σχέση με την προηγουμένη ή τις προηγούμενες τιμές της χρονοσειρας και συνήθως είναι πολύ σημαντικό μετρό εκτίμησης της επομένης τιμής.



Εικόνα 1: κατηγορίες δεδομένων

2.2 ΠΕΡΙΓΡΑΦΙΚΗ ΣΤΑΤΙΣΤΙΚΗ

Σε αυτή την ενότητα θα αναλύσουμε τις βασικές μεθόδους της περιγραφικής στατιστικής όπως την διαφορά μεταξύ πληθυσμού και δείγματος στα δεδομένα , μετρά κεντρικής τάσης , την συμμετρία των δεδομένων, μετρά διασποράς , την σχέση μεταξύ των δεδομένων και θα αναφερθούμε σε μερικά βασικά χαρακτηριστικά της κανονικής κατανομής.

Σχετικά με την διαφορά μεταξύ δείγματος και πληθυσμού αυτή προκύπτει από το γεγονός ότι ο πληθυσμός αναφέρεται στο σύνολο των δεδομένων που θέλουμε να αναλύσουμε ,ενώ το δείγμα είναι συνήθως ένα μικρό υποσύνολο του πληθυσμού για παράδειγμα αν θέλουμε να κάνουμε μια ανάλυση για τους μισθούς των εργαζομένων μιας εταιρίας μεταφορών και αναφερόμαστε στον πληθυσμό θα πρέπει να συλλέξουμε όλους τους μισθούς των εργαζομένων αυτών που είναι σε άδεια, που έχουν πάει να παραδώσουν πακέτα ,που εργάζονται στην βραδινή βάρδια κλπ. ενώ από την άλλη αν οι εργαζόμενοι είναι συνολικά 100 και ο αναλυτής συλλέξει 20 μισθούς την ώρα που θα επισκεφθεί τις

εγκαταστάσεις της εταιρίας τότε αυτό είναι ένα δείγμα του πληθυσμού. Η παραπάνω διάκριση είναι πολύ σημαντική καθώς από αυτήν επηρεάζεται η τελική μορφή διάφορων στατιστικών τύπων όπως της διακύμανσης. Προφανώς επειδή είναι δύσκολη η συλλογή του συνόλου των δεδομένων στις αναλύσεις χρησιμοποιείται ένα μικρό μέρος αυτών δηλαδή δείγματα, βέβαια με την πάροδο του χρόνου και την εξέλιξη της τεχνολογίας (big data) είναι πιο εύκολο πλέον να συλλεχθεί το σύνολο των δεδομένων, για παράδειγμα οι μισθοί των εργαζομένων τις παραπάνω εταιρίας μπορούν να αποκτηθούν ανεξάρτητος της παρουσίας των εργαζομένων μέσο τις βάσης δεδομένων και πιο συγκεκριμένα με την χρήση του προγράμματος hr (human resources) της εταιρείας.

Συνεχίζουμε με την ανάλυση των μέτρων κεντρικής τάσης . Το πρώτο μετρό που θα αναλύσουμε είναι ο μέσος ορός οπού είναι το άθροισμα των τιμών των παρατηρήσεων ενός συνόλου δεδομένων διαιρεμένο με το πλήθος των παρατηρήσεων. Ο μέσος ορός είναι ένα από τα σημαντικότερα μετρά ανάλυσης των δεδομένων καθώς είναι μια αντιπροσωπευτική τιμή του συνόλου των δεδομένων, και μπορεί να εξηγήσει πολλές λειτουργίες σχετικά με την αποδοτικότητα μιας εφοδιαστικής αλυσίδας όπως για παράδειγμα το μέσο ορό πληρότητας των φορτηγών μιας εταιρείας logistic τον μέσο ορό ταχύτητας του μεταφορικού μέσου (πχ van) κατά την παράδοση και επίσης επιτρέπει την άμεση σύγκριση δυο διαφορετικών συνόλων δεδομένων πχ μέσος χρόνος παράδοσης 10 παλετών σε 5 διαφορετικά γεωγραφικά σημεία το πρωί σε σχέση με τον μέσο χρόνο παράδοσης των ιδίων ποσοτήτων στα ιδιά σημεία με το ίδιο όχημα αλλά το βραδύ. Το αρνητικό της χρήσης του μέσου ορού είναι ότι επηρεάζεται πολύ από ακραίες τιμές στο δείγμα (outliers) έστω για παράδειγμα ότι θέλουμε να υπολογίσουμε το μέσο εισόδημα των κάτοικων της Γαλλίας και έχουμε συλλέξει μόνο 10 τιμές(δείγμα του πληθυσμού) οπού οι 9 από αυτές είναι μεταξύ 20000 και 30000 ευρώ το έτος και η 10 τιμή είναι ενός κάτοικου του Μονακό με εισόδημα 300.000 ευρώ αυτή η τιμή αλλοιώνει σε μεγάλο βαθμό το αποτέλεσμα της ανάλυσης ωθώντας τον μέσο ορό πολύ πιο πάνω από τις φυσιολογικές τιμές.

$$\bar{\chi} = \frac{sum(xi)}{n}$$

Οπού χι είναι η κάθε παρατήρηση και n το μέγεθος του δείγματος.

Λύση στο παραπάνω πρόβλημα δίνει η χρήση της διάμεσου (median) η οποία για να βρεθεί θα πρέπει πρώτα να έχουν ταξινομηθεί τα δεδομένα σε φθίνουσα ή αύξουσα σειρά. Η διάμεσος είναι η τιμή η οποία βρίσκεται στην μεσαία θέση της σειράς των δεδομένων.

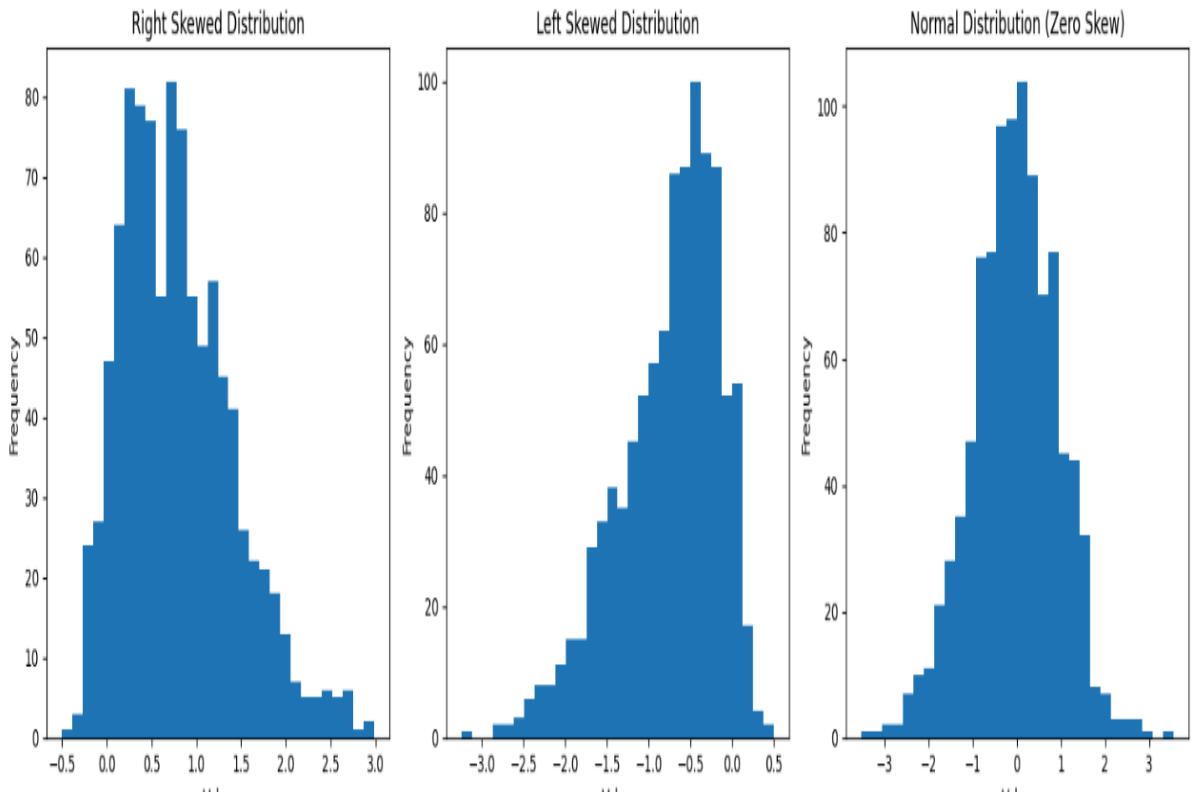
Αν ο αριθμός των παρατηρήσεων είναι άρτιος για να βρεθεί η διάμεσος πρέπει να υπολογιστεί ο μέσος ορός των δυο μεσαίων τιμών το θετικό της χρήσης του όπως αναφέρθηκε και παραπάνω είναι ότι δεν επηρεάζεται από ακραίες τιμές αλλά το βασικό μειονέκτημα του είναι ότι επειδή υπολογίζεται μόνο από τις μεσαίες τιμές του δείγματος αλλαγές στο σύνολο των δεδομένων ίσως να μην την επηρεάσουν με αποτέλεσμα να χαθεί η συνολική πληροφορία των δεδομένων.

Το τρίτο μετρό είναι το mode δηλαδή η τιμή που εμφανίζεται περισσότερες φορές μέσα στο συνολο των δεδομένων. Το συγκεκριμένο μετρό είναι πολύ χρήσιμο για την ανάλυση ποιοτικών μεταβλητών όπως για παράδειγμα πιο μέγεθος μπλούζας εμφανίζεται τις περισσότερές φορές , με αυτό τον τρόπο ο αναλυτής μπορεί να δει σε ένα βαθμό την ζήτηση του προϊόντος σε σύγκριση με τα αλλά μεγέθη, ή για παράδειγμα στην μεταβλητή εμπειρίας του πελάτη από την εξυπηρέτηση του μέσα στο κατάστημα. Το θετικό της χρήσης του mode είναι ότι και αυτό δεν επηρεάζεται από ακραίες τιμές αλλά στα μειονεκτήματα της χρήσης του συγκαταλέγεται το γεγονός ότι μπορεί να μην υπάρχει μόνο μια τιμή που εμφανίζεται τις περισσότερές φορές μέσα στο δείγμα , επίσης οι χρονοσειρες συνήθως δεν περνούν τις ιδίες τιμές καθώς δεν μένουν σταθερές για μεγάλο χρονικό διάστημα και αρά δεν μπορεί να υπολογιστεί το mode πχ τιμή μέτοχων και τέλος όπως και η διάμεσος δεν δίνει την πλήρη εικόνα των δεδομένων.

Έχοντας αναλύσει τα παραπάνω είμαστε πλέον σε θέση να αναφερθούμε στην ασσυμετρία (skewness) των δεδομένων. Η ασσυμετρία δείχνει κατά ποσό η κατανομή των δεδομένων είναι συμμετρική δηλαδή αν τα δεδομένα <<γέρνουν>> προς μια πλευρά. Αν ο μέσος ορός είναι μεγαλύτερος από την τιμή της διάμεσου καθώς υπάρχουν μεγάλες τιμές στο dataset πχ εισόδημα κατοίκου του Μονακό τότε λεμέ ότι υπάρχει θετική ασσυμετρία (right skew). Όταν η διάμεσος είναι μεγαλύτερη από τον μέσο ορό δηλαδή οι μεγάλες τιμές είναι περισσότερές από τις μικρές τιμές οι οποίες γίνονται ακραίες τιμές για το συγκεκριμένο σύνολο δεδομένων τότε υπάρχει αρνητική ασσυμετρία (left skew). Και τέλος αν ο μέσος ορός το mode και η διάμεσος έχουν την ιδιά τιμή τότε η κατανομή είναι συμμετρική και συνήθως τείνει να είναι παρόμοια με την κανονική κατανομή.

$$skew = \frac{n}{(n-1)*(n-2)} * sum(\frac{xi-\bar{x}}{s})\text{^}3$$

Οπού χι είναι η κάθε μια παρατήρηση $\bar{\chi}$ ο μεσος ορος του δειγματος s τυπική απόκλιση του δείγματος και n ο αριθμός των παρατηρήσεων του δείγματος.



Εικονα 2: ασυμετρια

Συνεχίζουμε με τα μετρά διασποράς το βασικότερο είναι η διακύμανση και η τυπική απόκλιση. Η διακύμανση είναι ένα στατιστικό μετρό που δείχνει την απόσταση των δεδομένων από τον μέσο ορό. Για τον υπολογισμό της υπολογίζουμε την διαφορά κάθε παρατήρησης από τον μέσο ορό των δεδομένων και στην συνέχεια αθροίζουμε αυτές τις αποστάσεις και τις διαιρούμε με τον αριθμό των παρατηρήσεων εάν πρόκειται για δεδομένα που είναι το σύνολο του πληθυσμού ενώ εάν πρόκειται για δεδομένα που αποτελούν δείγμα τότε η διαίρεση γίνεται αφαιρώντας μια μονάδα από τον αριθμό των παρατηρήσεων μικραίνοντας με αυτόν τον τρόπο τον παρονομαστή και αυξάνοντας την τελική τιμή του κλάσματος προσδίδοντας με αυτό τον τρόπο αυξημένη μεταβλητότητα (καθώς δεν γνωρίζουμε τον πληθυσμό). Όσο πιο μικρή είναι η διακύμανση τόσο πιο κοντά είναι οι παρατηρήσεις στο μέσο ορό ενώ όσο πιο μεγάλη είναι η διακύμανση τόσο πιο απομακρυσμένες είναι οι τιμές από τον μέσο ορό. Επειδή το αποτέλεσμα της διακύμανσης είναι στο τετράγωνο για να μετατραπούν στην αρχική μορφή των δε δεδομένων περνούμε την ριζά του κλάσματος το αποτέλεσμα της πράξης είναι η τυπική απόκλιση.

$$std\ \text{πλυθησμου} = \sqrt{\frac{sum(xi-\bar{x})\text{^}2}{N}}$$

$$std\ \text{δειγματος} = \sqrt{\frac{sum(xi-\bar{x})\text{^}2}{N-1}}$$

Οπού χι είναι κάθε μια παρατήρηση του δείγματος, $\bar{\chi}$ ο μεσος ορος του δειγματος και N ο αριθμός των παρατηρήσεων. Σε αυτό το σημείο θα κάνουμε ένα παράδειγμα για την κατανόηση της τυπικής απόκλισης καθώς χρησιμοποιείται σε όλες σχεδόν τις στατιστικές μεθόδους.

Έστω ότι ο υπεύθυνος σε μια αποθήκη θέλει να αποφασίσει σε ποιον από τους δυο υπάλληλους του θέλει να δώσει τα κλειδιά για το άνοιγμα της αποθήκης την επόμενη μέρα καθώς δεν θα μπορέσει να έρθει στην δουλειά λόγο αδείας, έχει μετρήσει 10 ημέρες (δείγμα καθώς οι υπάλληλοι εργάζονταν πάνω από 2 έτη στην εταιρεία) τον χρόνο σε λεπτά που καθυστερούν να έρθουν στην δουλειά. Ο πρώτος που έρχεται περπατώντας εκ πρώτης όψεως έχει αργήσει περισσότερα λεπτά από τον δεύτερο που έρχεται με το αυτοκίνητο του παρόλα αυτά ο μέσος ορός και των δυο είναι ίσος με 6,4 λεπτά , η διαφορά εγγυείται στην τυπική απόκλιση καθώς ο πρώτος υπάλληλος καθυστερεί σταθερά 5 με 8 λεπτά ενώ ο δεύτερος οπού δεν καθυστερεί σχεδόν καθόλου λόγο κίνησης στον κιφήσο μια ημέρα από τις 10 άργησε 60 λεπτά με αποτέλεσμα η τυπική απόκλιση να είναι πολύ μεγαλύτερη σε σχέση με αυτή του πρώτου υπαλλήλου. Με βάση αυτά τα δεδομένα η απόφαση θα παρθεί σε σχέση με το ποσό αποδεκτός ή όχι είναι ο υπεύθυνος της αποθήκης στον κίνδυνο , αν θέλει να είναι σίγουρος ότι η αποθήκη θα ανοίξει με μια μικρή καθυστέρηση θα δώσει τα κλειδιά στον πρώτο υπάλληλο, εάν από την άλλη θεωρεί ότι τα 60 λεπτά καθυστέρηση είναι μια ακραία τιμή (outlier) και ότι δεν πρόκειται να ξανασυμβεί θα δώσει τα κλειδιά στον δεύτερο υπάλληλο οπού τις περισσότερές ημέρες ερχόταν στην ώρα του. Το συμπέρασμα από τα παραπάνω είναι ότι για να έχει μια καλύτερη εικόνα ο αναλυτής αναφορικά με την τυπική απόκλιση θα πρέπει να έχει όσο το δυνατόν μεγαλύτερο δείγμα(στο παράδειγμα μας ο υπεύθυνος να κατέγραφε κάθε μέρα για 2 χρονιά τον χρόνο που καθυστερούσαν να έρθουν οι υπάλληλοι) έτσι ώστε να ελαττωθεί η επίδραση των ακραίων τιμών.

παρατήρηση	υπάλληλος 1 περπάτημα	υπάλληλος 2 αυτοκίνητο
1	5	0

2	7	1
3	6	0
4	8	1
5	5	60
6	7	0
7	6	1
8	7	0
9	6	1
10	7	0
μέσος ορός	6,4	6,4
τυπική απόκλιση	0,9660917831	18,83967445

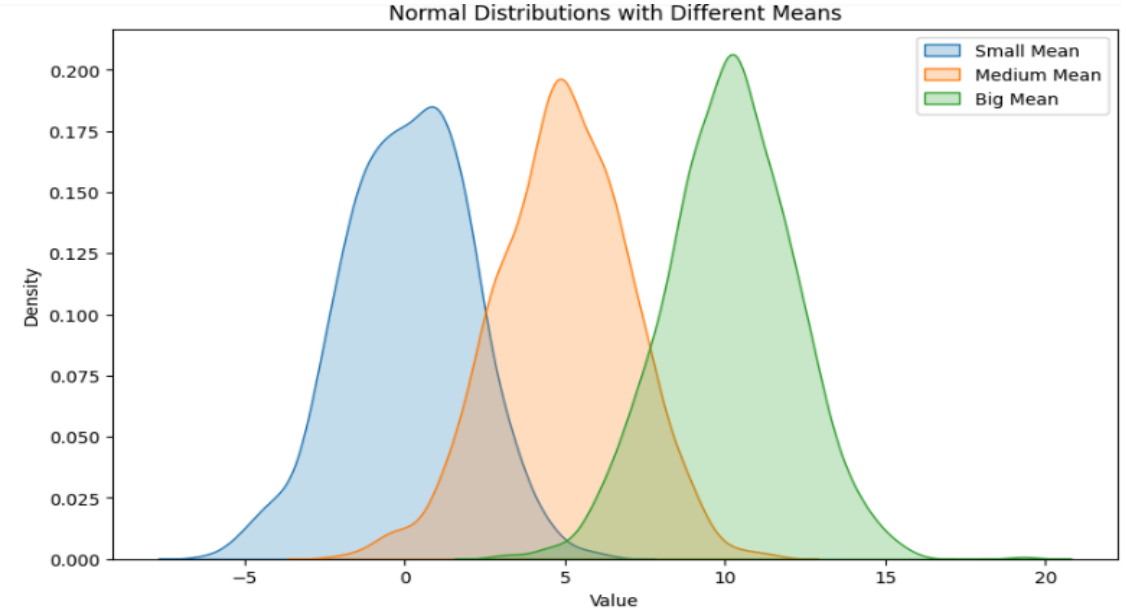
Συνεχίζουμε την ενότητα με την έννοια της συνδιακύμανσης και της συσχέτισης, και τα δυο αυτά στατιστικά μετρά είναι πολύ σημαντικά στην θεωρεία της μηχανικής μάθησης αλλά και των χρονοσειρων καθώς από αυτά πηγάζουν έννοιες όπως η αυτοσυσχετιση και η μερική αυτοσυσχετιση που θα αναλύσουμε στο κεφάλαιο 7. Η συσχέτιση (correlation) που είναι η κανονικοποιημενη μορφή της συνδιακύμανσης παίρνει τιμές από το -1 έως το 1 και δείχνει την κατεύθυνση που κινούνται δυο μεταβλητές. Εάν η συσχέτιση παίρνει τιμές μεταξύ ενός νούμερου μεγαλύτερου του μηδενός και του ένα τότε λεμέ ότι υπάρχει θετική συσχέτιση μεταξύ των δυο μεταβλητών δηλαδή όταν αυξάνεται η μια αυξάνεται και η άλλη όταν μειώνεται η μια μειώνεται και η άλλη ανάλογα με το ποσό μεγάλος είναι ο συντελεστής συσχέτισης ,παράδειγμα θετικής συσχέτισης είναι η αύξηση του αποθέματος που έχει ως συνέπεια την αύξηση του κόστους αποθέματος (περισσότερα αποθέματα περισσότεροι εργαζόμενοι για την διαχείριση τους αύξηση εξόδων για ενοικίαση αποθηκών κλπ.). Αν ο συντελεστής πάρει τιμή μικρότερη του μηδενός έως το -1 τότε λεμέ ότι υπάρχει αρνητική συσχέτιση μεταξύ των δυο μεταβλητών δηλαδή όταν αυξάνεται η μια μεταβλητή η άλλη μειώνεται , όταν η μια μειώνεται η άλλη αυξάνεται για παράδειγμα όταν μια εταιρεία logistics συνεργάζεται με αξιόπιστους προμηθευτές τότε μπορεί να κρατάει μικρότερα αποθέματα ασφάλειας καθώς γνωρίζει ότι ο προμηθευτής είναι αξιόπιστος και ότι την προμηθεύει πάντα στις συμφωνημένες χρονικές περιόδους(υψηλή αξιοπιστία του προμηθευτή οδηγεί σε μείωση των αποθεμάτων και χρήσης τεχνικών just in time). Και τέλος όταν ο συντελεστής συσχέτισης πάρει τιμή ίση με 0 τότε δεν υπάρχει συσχέτιση μεταξύ των δυο μεταβλητών για παράδειγμα το αν νίκησε μια ποδοσφαιρική ομάδα σε σχέση με τις τιμές που χρεώνει ένας προμηθευτής της πρώτες ύλες. Οι τύποι εμφανίζονται παρακάτω

$$Cov(X,Y)\text{δειγματος} = \frac{sum((xi-\bar{x})*(yi-\bar{y}))}{N-1}$$

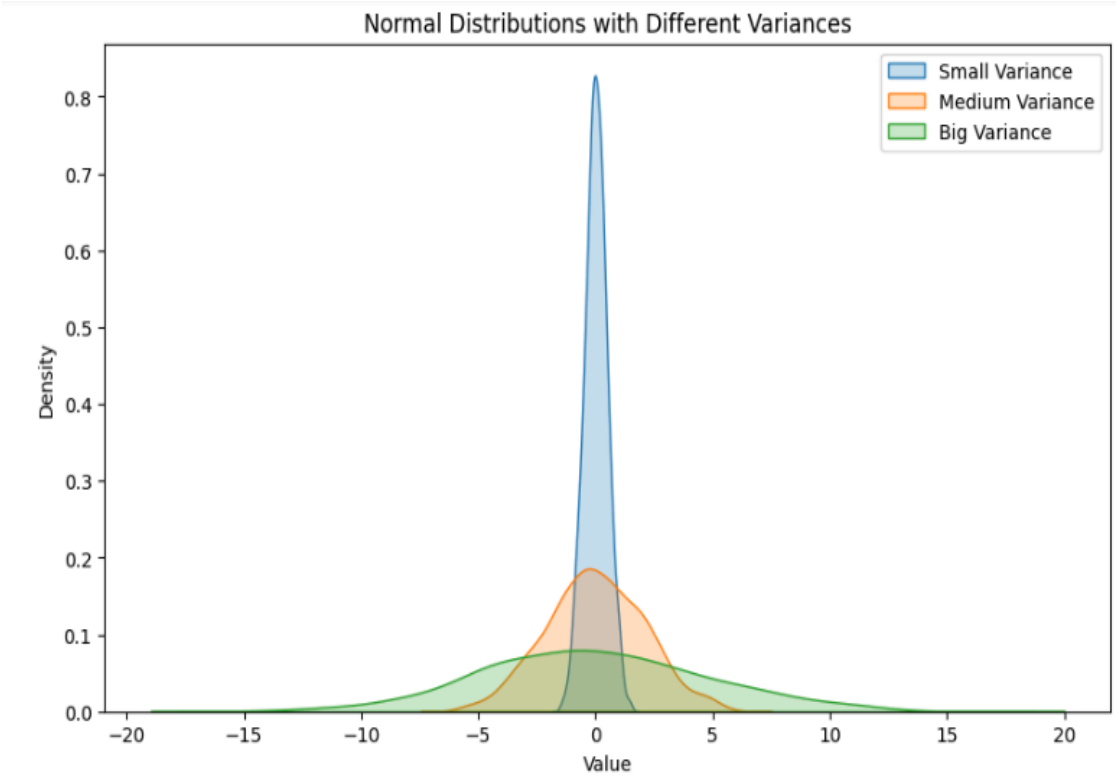
$$correlation = \frac{covariance(x,y)}{sx*sy}$$

Οπού xi είναι η κάθε τιμή του δείγματος χ , $\bar{\chi}$ είναι ο μεσος ορος του δειγματος χ ,sx η δειγματική τυπική αποκλείσει του δείγματος χ και οι αντίστοιχες τιμές του δείγματος y.

Τέλος θα ολοκληρώσουμε την ενότητα με τον ορισμό της κανονικής κατανομής και της κατανομής student t. Λεμέ ότι τα δεδομένα ακολουθούν την κανονική τιμή όταν είναι συμμετρικά μοιρασμένα γύρω από την μέση τιμή του δείγματος και ο μέσος ορός η διάμεσος και το mode του δείγματος ισούνται μεταξύ τους. Τα παραπάνω έχουν ως αποτέλεσμα το σχήμα της κατανομής να είναι σαν καμπάνα. Όταν ο μέσος ορός του δείγματος αυξάνεται η κατανομή μετατοπίζεται προς τα δεξιά όταν μειώνεται η κατανομή μετατοπίζεται προς τα αριστερά. Επίσης όταν η διακύμανση αυξάνεται και οι τιμές απέχουν μεγαλύτερη απόσταση από την μέση τιμή τότε η κατανομή γίνεται πιο πλατιά ενώ αν συμβαίνει το αντίθετο η κατανομή είναι πιο λιγνή. Αναφορικά με την κατανομή student t αυτή έχει πιο παχιές ουρές σε σχέση με την κανονική κατανομή που σηματοδοτεί ότι τα δεδομένα έχουν μεγάλη διασπορά μεταξύ τους δηλαδή απέχουν αρκετά από την μέση τιμή αυξάνοντας την αβεβαιότητα, όταν το δείγμα αποτελείται από λίγες παρατηρήσεις είναι καλύτερο να χρησιμοποιηθεί η student t κατανομή.



Εικονα 3: κανονικες κατανομες με διαφορετικους μεσους ορους



Εικονα 4: κανονικες κατανομες με διαφορετικες διακυμανσεις

2.3 ΔΙΑΣΤΗΜΑΤΑ ΕΜΠΙΣΤΟΣΥΝΗΣ

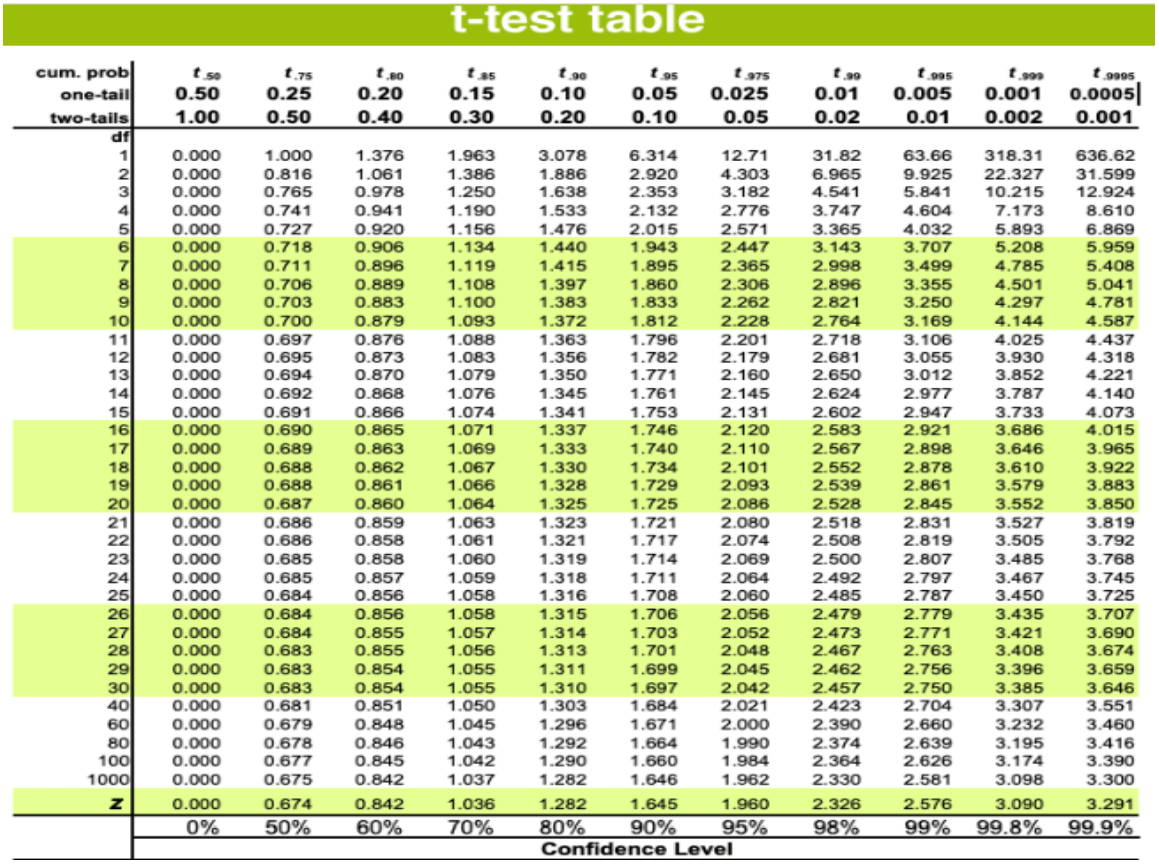
Σε αυτή την ενότητα θα περιγράψουμε μερικά από τα σημαντικότερα διάστημα εμπιστοσύνης για την ανάλυση των εφοδιαστικών αλυσίδων. Αρχικά να αναφέρουμε ότι τα διαστήματα εμπιστοσύνης είναι στατιστικά εργαλεία τα οποία χρησιμοποιούνται για να δώσουν ένα εύρος τιμών γύρω από μια στατιστική παράμετρο όπως τον μέσο ορό ενός δείγματος ,την τυπική απόκλιση του και αλλά. Μια από τις σημαντικότερες παραμέτρους των διαστημάτων εμπιστοσύνης είναι ο συντελεστής α δηλαδή το επίπεδο σημαντικότητας το οποίο είναι η πιθανότητα να κάνει ο αναλυτής σφάλμα τύπου 1(δηλαδή να απορρίψει την μηδενική απόφαση ενώ κανονικά δεν θα έπρεπε να την είχε απορρίψει). Με βάση αυτό το α καθορίζει και το επίπεδο εμπιστοσύνης των διαστημάτων εμπιστοσύνης δηλαδή εάν το α είναι 0,05 τότε το επίπεδο εμπιστοσύνης είναι 95% εάν το α είναι 0,01 τότε το επίπεδο εμπιστοσύνης είναι 99%. Όσο πιο μικρό είναι το α τόσο πιο μεγάλο είναι το εύρος

του διαστήματος εμπιστοσύνης προφανώς υπάρχουν και άλλοι παράγοντες που επηρεάζουν το εύρος του διαστήματος εμπιστοσύνης όπως για παράδειγμα η τυπική απόκλιση του δείγματος οπού όσο μεγαλύτερη είναι τόσο πιο εύρη γίνεται και το διάστημα εμπιστοσύνης επίσης σημαντικό ρολό παίζει και το μέγεθος του δείγματος όταν είναι μικρό η αβεβαιότητα και κατά επέκταση το εύρος του διαστήματος εμπιστοσύνης αυξάνεται το αντίθετο συμβαίνει όταν έχουμε πολλές παρατηρήσεις, επίσης πρέπει να επισημανθεί ότι όταν οι παρατηρήσεις είναι λιγότερες από 30 το στατιστικό που χρησιμοποιείται για την δημιουργία του διαστήματος εμπιστοσύνης προέρχεται από την κατανομή student t και όχι από την κανονική κατανομή το ίδιο ισχύει και όταν αναλύουμε ένα δείγμα και όχι τον πληθυσμό.

Το πρώτο διάστημα εμπιστοσύνης και το πιο σύνηθες είναι αυτό που δημιουργείτε για να αναλύσει το μέσο ορό ενός δείγματος με άγνωστες διακυμάνσεις (δηλαδή το δείγμα αποτελείται από λίγες παρατηρήσεις), για την κατασκευή του χρειάζεται ο αναλυτής να υπολογίσει τον μέσο ορό($\bar{\chi}$) και την τυπική αποκλήσου(σ) του δείγματος(ή του πληθυσμού) και το t statistic το οποίο μπορεί κάνεις να βρει χρησιμοποιώντας συναρτήσεις του excel , βιβλιοθήκες της python αλλά και μέσο του t-table οπού είναι αναρτημένο στο διαδίκτυο ή στην τελευταία σελίδα βιβλίων στατιστικής ανάλογα με το α και τους βαθμούς ελευθέριας. Οι βαθμοί ελευθέριας δεν υπολογίζονται πάντα με τον ίδιο τρόπο, αλλά τις περισσότερές φορές βασίζονται στο αριθμό των παρατηρήσεων του δείγματος(n).

$$\text{διαστημα εμπιστοσυνης για τον μεσο ορο ενος πλυθισμου} = \bar{\chi} \pm z_{a/2} * \frac{\sigma}{\sqrt{n}}$$

$$\text{διαστημα εμπιστοσυνης για τον μεσο ορο ενος δειγματος} = \bar{\chi} \pm \tau_{n-1,a/2} * \frac{s}{\sqrt{n}}$$



t-test table

cum. prob	$t_{.50}$	$t_{.75}$	$t_{.80}$	$t_{.85}$	$t_{.90}$	$t_{.95}$	$t_{.975}$	$t_{.99}$	$t_{.995}$	$t_{.999}$	$t_{.9995}$
one-tail	0.50	0.25	0.20	0.15	0.10	0.05	0.025	0.01	0.005	0.001	0.0005
two-tails	1.00	0.50	0.40	0.30	0.20	0.10	0.05	0.02	0.01	0.002	0.001
df											
1	0.000	1.000	1.376	1.963	3.078	6.314	12.71	31.82	63.66	318.31	636.62
2	0.000	0.816	1.061	1.386	1.886	2.920	4.303	6.965	9.925	22.327	31.599
3	0.000	0.765	0.978	1.250	1.638	2.353	3.182	4.541	5.841	10.215	12.924
4	0.000	0.741	0.941	1.190	1.533	2.132	2.776	3.747	4.604	7.173	8.610
5	0.000	0.727	0.920	1.156	1.476	2.015	2.571	3.365	4.032	5.893	6.869
6	0.000	0.718	0.906	1.134	1.440	1.943	2.447	3.143	3.707	5.208	5.959
7	0.000	0.711	0.896	1.119	1.415	1.895	2.365	2.998	3.499	4.785	5.408
8	0.000	0.706	0.889	1.108	1.397	1.860	2.306	2.896	3.355	4.501	5.041
9	0.000	0.703	0.883	1.100	1.383	1.833	2.262	2.821	3.250	4.297	4.781
10	0.000	0.700	0.879	1.093	1.372	1.812	2.228	2.764	3.169	4.144	4.587
11	0.000	0.697	0.876	1.088	1.363	1.796	2.201	2.718	3.106	4.025	4.437
12	0.000	0.695	0.873	1.083	1.356	1.782	2.179	2.681	3.055	3.930	4.318
13	0.000	0.694	0.870	1.079	1.350	1.771	2.160	2.650	3.012	3.852	4.221
14	0.000	0.692	0.868	1.076	1.345	1.761	2.145	2.624	2.977	3.787	4.140
15	0.000	0.691	0.866	1.074	1.341	1.753	2.131	2.602	2.947	3.733	4.073
16	0.000	0.690	0.865	1.071	1.337	1.746	2.120	2.583	2.921	3.686	4.015
17	0.000	0.689	0.863	1.069	1.333	1.740	2.110	2.567	2.898	3.646	3.965
18	0.000	0.688	0.862	1.067	1.330	1.734	2.101	2.552	2.878	3.610	3.922
19	0.000	0.688	0.861	1.066	1.328	1.729	2.093	2.539	2.861	3.579	3.883
20	0.000	0.687	0.860	1.064	1.325	1.725	2.086	2.528	2.845	3.552	3.850
21	0.000	0.686	0.859	1.063	1.323	1.721	2.080	2.518	2.831	3.527	3.819
22	0.000	0.686	0.858	1.061	1.321	1.717	2.074	2.508	2.819	3.505	3.792
23	0.000	0.685	0.858	1.060	1.319	1.714	2.069	2.500	2.807	3.485	3.768
24	0.000	0.685	0.857	1.059	1.318	1.711	2.064	2.492	2.797	3.467	3.745
25	0.000	0.684	0.856	1.058	1.316	1.708	2.060	2.485	2.787	3.450	3.725
26	0.000	0.684	0.856	1.058	1.315	1.706	2.056	2.479	2.779	3.435	3.707
27	0.000	0.684	0.855	1.057	1.314	1.703	2.052	2.473	2.771	3.421	3.690
28	0.000	0.683	0.855	1.056	1.313	1.701	2.048	2.467	2.763	3.408	3.674
29	0.000	0.683	0.854	1.055	1.311	1.699	2.045	2.462	2.756	3.396	3.659
30	0.000	0.683	0.854	1.055	1.310	1.697	2.042	2.457	2.750	3.385	3.646
40	0.000	0.681	0.851	1.050	1.303	1.684	2.021	2.423	2.704	3.307	3.551
60	0.000	0.679	0.848	1.045	1.296	1.671	2.000	2.390	2.660	3.232	3.460
80	0.000	0.678	0.846	1.043	1.292	1.664	1.990	2.374	2.639	3.195	3.416
100	0.000	0.677	0.845	1.042	1.290	1.660	1.984	2.364	2.626	3.174	3.390
1000	0.000	0.675	0.842	1.037	1.282	1.646	1.962	2.330	2.581	3.098	3.300
z	0.000	0.674	0.842	1.036	1.282	1.645	1.960	2.326	2.576	3.090	3.291
	0%	50%	60%	70%	80%	90%	95%	98%	99%	99.8%	99.9%
	Confidence Level										

Εικονα 5: πινακας με τις student t critical value

Ένα παράδειγμα χρίσης του συγκεκριμένου διαστήματος εμπιστοσύνης στην εφοδιαστική αλυσίδα είναι για την εύρεση του διαστήματος εμπιστοσύνης του μέσου ορού των πωλήσεων ενός προϊόντος. Έστω οι παρακάτω μηνιαίες πωλήσεις με βάση την παραπάνω συνάρτηση υπολογίζουμε τον μέσο ορό του δείγματος την τυπική απόκλιση και βρίσκουμε το κατάλληλο τ στατιστικό το οποίο υπολογίζεται με βάση το α ίσο 5% και οι βαθμοί ελευθέριας είναι ίσοι με 11 καθώς το σύνολο των παρατηρήσεων είναι 12 - 1 =11 και αρά είναι ίσο με 2,201. Το αποτέλεσμα της μεθόδου είναι ότι υπάρχει 95% πιθανότητα ο πραγματικός μέσος ορός των πωλήσεων του προϊόντος να είναι μεταξύ των 74 και 105 τεμαχίων , με βάση αυτό μπορούν να παρθούν κάποια συμπεράσματα για την επόμενη παραγγελία του συγκεκριμένου προϊόντος , για παράδειγμα αν η εταιρεία θέλει να είναι 95% σίγουρη ότι δεν θα ξεμείνει από το συγκεκριμένο προϊόν θα πρέπει να παραγγείλει 105 τεμάχια τον επόμενο μηνά αν από την άλλη μπορεί να ανεχτεί το ρίσκο της έλλειψης αποθεμάτων θα μπορούσε να παραγγείλει 74 τεμάχια τον επόμενο μηνά.

μήνας	πωλήσεις σε τεμάχια

1	90
2	80
3	85
4	75
5	73
6	60
7	68
8	74
9	90
10	110
11	128
12	138
μέσος ορός	89,25
τυπική απόκλιση	24,15903746
άνω οροί	104,6000236
κάτω οροί	73,89997643

Πριν συνεχίσουμε την ανάλυση των διαστημάτων εμπιστοσύνης του μέσου ορού μεταξύ δυο δειγμάτων θα πρέπει να αναφερθούμε σε έναν έλεγχο υποθέσεων ο οποίος δείχνει εάν οι διακυμάνσεις των δειγμάτων είναι ίσες ή άνισες καθώς μέσα από αυτόν καθορίζεται ο τρόπος υπολογισμού των διαστημάτων εμπιστοσύνης αυτής της μορφής. Αυτός ο έλεγχος γίνεται μέσο της κατανομής f (η οποία είναι right-skewed και παίρνει μόνο θετικές τιμές) και έχει ως μηδενική υπόθεση ότι οι διακυμάνσεις των δειγμάτων είναι ίσες $\sigma_{\alpha}^{2} = \sigma_{\beta}^{2}$ και ως εναλακτικη υποθεση ότι οι δικυμασεις είναι ανισες $\sigma_{\alpha}^{2} \neq \sigma_{\beta}^{2}$.Στην συνέχεια ο αναλυτής υπολογίζει τις διακυμάνσεις των δειγμάτων ώστε να βρει το f στατιστικό οπού θα χρησιμοποιηθεί στον έλεγχο της υπόθεσης, ο τρόπος υπολογισμού του είναι μέσο της διαίρεσης τοποθετώντας στον αριθμητή την μεγαλύτερη από τα δυο δείγματα διακύμανση και τοποθετώντας στον παρανομαστεί αυτή με την μικρότερη διακύμανση . Στην συνέχεια συγκρίνεται αυτό το στατιστικό με το critical value της f κατανομής ανάλογα με το επίπεδο σημαντικότητας και τους βαθμούς ελευθέριας.

$$f\ statistic = \frac{s_a\text{^}2}{s_b\text{^}2}$$

Τότε εάν το f στατιστικό είναι μεγαλύτερο από την κρίσιμη τιμή $f_{a/2,df1,df2}$ (οπού df1=nα−1 και df2=nβ−1 και n ο αριθμός των παρατηρήσεων του εκάστοτε δείγματος)που

προέρχεται από τον αντίστοιχο πίνακα της κατανομής f ο αναλυτής απορρίπτει την μηδενική υπόθεση και οι διακυμάνσεις των δειγμάτων είναι άνισες αντίθετος εάν το f στατιστικό είναι μικρότερο από την κριτική τιμή f τότε πρέπει να αποδεχτεί την μηδενική υπόθεση και οι διακυμάνσεις θεωρούνται ίσες.

Έχοντας αναφέρει τα παραπάνω τα διαστήματα εμπιστοσύνης για την διαφορά των μέσων ορών δυο δειγμάτων είναι τα παρακάτω.

Αν την ανάλυση την κάνουμε με δεδομένα που είναι το σύνολο του πληθυσμού τότε η critical value προέρχεται από την κανονική κατανομή:

$$\bar{\chi}_{\alpha} - \bar{\chi}_{b} \pm z_{a/2} * \sqrt{\frac{\sigma\alpha\text{^}2}{na} + \frac{\sigma b\text{^}2}{nb}}$$

Αν η ανάλυση γίνεται με δε δεδομένα που αποτελούν ένα δείγμα του συνόλου του πληθυσμού και οι διακυμάνσεις μεταξύ των δυο αυτών δειγμάτων είναι ίσες τότε το critical value προέρχεται από την student t κατανομή:

$$\bar{\chi}_{\alpha} - \bar{\chi}_{b} \pm \tau_{\frac{a}{2},na+nb-2} * sp\sqrt{\frac{1}{na} + \frac{1}{nb}}$$

Οπού το sp είναι η συνεκτιμημένη διακύμανση των δειγμάτων και υπολογίζεται από τον παρακάτω τύπο:

$$sp = \sqrt{\frac{(na-1)*sa^{2}+(nb-1)*sb\text{^}2}{na+nb-2}}$$

Οπού na είναι ο αριθμός των παρατηρήσεων του δείγματος α και nb του δείγματος β αντίστοιχα , επίσης sa και sb είναι η τυπική απόκλιση του εκάστοτε δείγματος.

Εάν η ανάλυση γίνεται και αυτή την φορά με δεδομένα που αποτελούν ένα υποσύνολο του πληθυσμού (δείγματα) αλλά οι διακυμάνσεις μεταξύ των δειγμάτων είναι άνισες με βάση τον έλεγχο υποθέσεων τότε και αυτή την φορά χρησιμοποιείται το critical value από την student t κατανομή για τον υπολογισμό του παρακάτω διαστήματος εμπιστοσύνης.

$$\bar{\chi}_{\alpha} - \bar{\chi}_{b} \pm \tau_{\frac{a}{2},\nu} * \sqrt{\frac{sa\text{^}2}{na} + \frac{sb\text{^}2}{nb}}$$

Ο τρόπος υπολογισμού των βαθμών ελευθέριας προκύπτει από τον τύπο του welch Satterthwaite οπού

$$\text{βαθμοι ελευθεριας}(v) = \frac{(\frac{sa^2}{na}+\frac{sb^2}{nb})\text{^}2}{\frac{(\frac{sa^2}{na})\text{^}2}{na-1}+\frac{(\frac{sb^2}{nb})\text{^}2}{nb-1}}$$

Έχοντας αναλύσει τα παραπάνω θα κάνουμε ένα παράδειγμα υπολογισμού διαστήματος εμπιστοσύνης για τον μέσο ορό δυο δειγμάτων. Πιο συγκεκριμένα έστω ότι έχουμε δώδεκα μηνιαίες πωλήσεις ενός κοινού προϊόντος δυο καταστημάτων μιας αλυσίδας λιανικών πωλήσεων το ένα κατάστημα είναι στο περιστέρι και το άλλο στον Άλιμο και θέλουμε να ελέγξουμε πόσες παραπάνω κατά μέσο ορό είναι οι πωλήσεις του πρώτου έναντι του δευτέρου καταστήματος ώστε να αποφασίσει ο υπεύθυνος της εφοδιαστικής αλυσίδας εάν πρέπει να αυξηθούν τα δρομολόγια των φορτηγών τροφοδοσίας σε ένα από τα δυο κατάστημα. Αρχικά υπολογίζουμε τον μέσο ορό την διακύμανση και την τυπική απόκλιση των δυο δειγμάτων, στην συνέχεια κάνουμε τον έλεγχο ισότητας των διακυμάνσεων για τον υπολογισμό του f στατιστικού βάζουμε ως αριθμητή την μεγαλύτερη μεταξύ των δυο διακύμανση δηλαδή του καταστήματος του Αλίμου. Το αποτέλεσμα της πράξης είναι 1,1 στην συνέχεια χρησιμοποιώντας τον τύπο finv στο excel βάζοντας ως επίπεδο σημαντικότητας το α=0,05 αρά α/2=0,025 και βαθμούς ελευθέριας 11 και 11 όσες οι παρατηρήσεις του εκάστοτε δείγματος μείον μια με βάση τον τύπο που δείξαμε παραπάνω περνούμε ως κριτική τιμή το 3,47 το οποίο είναι μεγαλύτερο από το φ στατιστικό οπότε δεν μπορούμε να απορρίψουμε σε επίπεδο εμπιστοσύνης 95% το γεγονός ότι οι διακυμάνσεις είναι ίσες οπότε για τον υπολογισμό του διαστήματος εμπιστοσύνης χρησιμοποιείται ο τύπος με άγνωστο πληθυσμό και ίσες διακυμάνσεις. Οπότε στην συνέχεια εφαρμόζεται ο αντίστοιχος τύπος οπού πρέπει να υπολογιστεί το sp για την εύρεση της τ κριτικής τιμής χρησιμοποιήθηκε ο τύπος tinv του excel με επίπεδο σημαντικότητας α=0,05 αρά α/2=0,025 με 22 (nα=12+nβ=12=24-2=22) βαθμούς ελευθέριας.

Το αποτέλεσμα είναι ότι το κατάστημα του Αλίμου πουλάει 3 έως 43 περισσότερα προϊόντα αυτού του είδους μέσα στο έτος 95% των περιπτώσεων αρά ο υπεύθυνος εφοδιασμού θα πρέπει να έχει υπόψη του πιθανή αύξηση των δρομολογίων για μεταφορά του προϊόντος σε αυτό το κατάστημα.

μήνας	κατάστημα περιστέρι	κατάστημα Αλίμου
1	120	155
2	130	145
3	125	160
4	140	150
5	138	165
6	150	175
7	155	160
8	160	180
9	162	185
10	170	195
11	175	200
12	180	210
μέσος ορός	150,4166667	173,3333333
διακύμανση	396,4469697	437,8787879
τυπική απόκλιση	19,91097611	20,92555347
f statistic	1,104507844	
f critical	3,473699051	
sp	20,42456557	
τ critical value	2,405472746	
ανω οριο	-2,859127752	
κατω οριο	-42,97420558	

Υπάρχουν και αλλά διαστήματα εμπιστοσύνης όπως αυτό για την διακύμανση ενός δείγματος για τις διακυμάνσεις δυο δειγμάτων για πιθανότητες κλπ. αλλά δεν θα αναλυθούν στην παρούσα εργασία καθώς δεν χρησιμοποιήθηκαν σε κάποιο από τα dataset.

2.4 ΕΛΕΝΧΟΙ ΥΠΟΘΕΣΕΩΝ

Όπως τα διαστήματα εμπιστοσύνης ένα άλλο πολύ σημαντικό εργαλείο ανάλυσης δεδομένων είναι οι έλεγχοι υποθέσεων , καθώς μπορούν να χρησιμοποιηθούν σε δεδομένα πωλήσεων πχ να ελεγχθεί εάν οι πωλήσεις ενός καταστήματος είναι περισσότερές έναντι ενός αλλού καταστήματος , εφοδιαστικών αλυσίδων για την διαχείριση αποθεμάτων πρόβλεψη ζήτησης ανάλυση παραδόσεων , marketing ,ιατρικής , οικονομίων, χημείας και άλλων τομέων της οικονομίας. Έχουμε ήδη αναφερθεί στον έλεγχο υποθέσεων για την ισότητα των διακυμάνσεων δυο δειγμάτων , με τον ίδιο ακριβός τρόπο λειτουργούν και οι υπόλοιποι έλεγχοι υποθέσεων αναφορικά με τον μέσο ορό ενός δείγματος , τον μέσο ορό δυο δειγμάτων , για την διακύμανση ενός δείγματος για ποσοστά πιθανοτήτων και αλλά

ουσιαστικά οι έλεγχοι υποθέσεων ξεκινούν θέτοντας μια υπόθεση ως μηδενική και στην συνέχεια ο αναλυτής προσπαθεί με βάση τα δεδομένα που έχει συλλέξει να απορρίψει αυτήν την υπόθεση.

Ο πρώτος έλεγχος υποθέσεων που θα αναλύσουμε είναι για το μέσο ορό του πληθυσμού και του δείγματος. Για να πραγματοποιηθεί ο έλεγχος υποθέσεων θα πρέπει πρώτα ο αναλυτής να υπολογίσει το t,z,f,x^2 στατιστικό και να το συγκρίνει με την αντίστοιχη κρίσιμη τιμή της κατανομής που χρησιμοποιήθηκε για τον υπολογισμό του στατιστικού. Επίσης θα πρέπει να δοθεί έμφαση στον τρόπο με τον οποίο διατυπώνεται η μηδενική υπόθεση καθώς μέσο αυτής καθορίζεται ο έλεγχος των υποθέσεων. Εάν θέλουμε να ελέγξουμε εάν ο μέσος ορός ενός δείγματος είναι διαφορετικός από μια τιμή τότε ο έλεγχος είναι δίπλευρος Η0: Χ(ΜΕΣΟΣ ΔΕΙΓΜΑΤΟΣ) = ΜΙΑ ΤΥΧΑΙΑ ΤΙΜΗ

Η1: Χ(ΜΕΣΟΣ ΔΕΙΓΜΑΤΟΣ) ≠ ΜΙΑ ΤΥΧΑΙΑ ΤΙΜΗ στην συνέχεια υπολογίζεται το στατιστικό με βάση τον παρακάτω τύπο ανάλογα εάν τα δεδομένα αποτελούν πληθυσμό ή αν αποτελούν μέρος αυτού:

$$z\ \text{στατιστικο(για πλυθισμο)} = \frac{\bar{\chi} - \mu 0}{\frac{\sigma}{\sqrt{n}}}$$

$$t\ \text{στατιστικο(για δειγμα)} = \frac{\bar{\chi} - \mu 0}{\frac{s}{\sqrt{n}}}$$

Οπού $\overline{\chi}$ είναι ο μεσος του δειγματος μ0 μια τυχαια τιμη(που θέλουμε να συγκρίνουμε με τον μέσο ορό του δείγματος αν είναι μεγαλύτερη ή μικρότερη) σ η τυπική απόκλιση του πληθυσμού και s η τυπική απόκλιση του δείγματος και n ο αριθμός των παρατηρήσεων.

Στην συνέχεια γίνεται σύγκριση του στατιστικού με την κρίσιμη τιμή της αντίστοιχης κατανομής για τον πληθυσμό η σύγκριση γίνεται με βάση τον παρακάτω τύπο:

$$|z\ \text{στατιστικο}| > z\ critical\ value\ a/2$$

Ενώ η αντίστοιχη ανισότητα για το δείγμα είναι η παρακάτω:

$$|\tau\ \text{στατιστικο}| > \tau\ critical\ value \frac{a}{2}, n - 1\ \text{βαθμοι ελευθεριας}$$

Εάν όμως θέλουμε να ελέγξουμε ένα η μέση τιμή του δείγματος ή του πληθυσμού είναι μικρότερη από μια τυχαία τιμή τότε ο έλεγχος υποθέσεων γράφεται με τον παρακάτω τρόπο:

Η0:Χ(ΜΕΣΟΣ ΔΕΙΓΜΑΤΟΣ)≥ ΜΙΑ ΤΥΧΑΙΑ ΤΙΜΗ

Η1:Χ(ΜΕΣΟΣ ΔΕΙΓΜΑΤΟΣ)< ΜΙΑ ΤΥΧΑΙΑ ΤΙΜΗ

Και οι ανισότητες εκφράζονται με τον παρακάτω τρόπο καθώς ο έλεγχος είναι μονόπλευρος προς την αριστερή <<ουρά>> της κατανομής

$$\zeta \text{ στατιστικο} < - \ z \ critical \ value \ a$$

εάν ισχύει το παραπάνω τότε απορρίπτουμε την η0 και δεχόμαστε την η1 στο αντίστοιχο επίπεδο σημαντικότητας εάν δεν ισχύει το παραπάνω τότε δεν μπορούμε να απορρίψουμε την μηδενική απόφαση. Η αντίστοιχη ανισότητα για τον δειγματικό έλεγχο υποθέσεων παρουσιάζεται παρακάτω:

$$\tau \text{ στατιστικο} < - \ \tau \ critical \ value \ a, n \ - \ 1 \text{ βαθμοι ελευθεριας}$$

Εάν όμως θέλουμε να ελέγξουμε εάν η μέση τιμή του δείγματος ή του πλεύσιμου είναι μεγαλύτερη από μια τυχαία τιμή τότε ο έλεγχος υποθέσεων γράφεται με τον παρακάτω τρόπο:

Η0:Χ(ΜΕΣΟΣ ΔΕΙΓΜΑΤΟΣ)≤ ΜΙΑ ΤΥΧΑΙΑ ΤΙΜΗ

Η1:Χ(ΜΕΣΟΣ ΔΕΙΓΜΑΤΟΣ)> ΜΙΑ ΤΥΧΑΙΑ ΤΙΜΗ

Και οι ανισότητες εκφράζονται με τον παρακάτω τρόπο καθώς ο έλεγχος είναι μονόπλευρος προς την δεξιά <<ουρά>> της κατανομής

$$\zeta \text{ στατιστικο} > \ z \ critical \ value \ a$$

Εάν ισχύει το παραπάνω τότε απορρίπτουμε την η0 και δεχόμαστε την η1 στο αντίστοιχο επίπεδο σημαντικότητας εάν δεν ισχύει το παραπάνω τότε δεν μπορούμε να απορρίψουμε την μηδενική απόφαση. Η αντίστοιχη ανισότητα για τον δειγματικό έλεγχο υποθέσεων παρουσιάζεται παρακάτω:

$$\tau \text{ στατιστικο} > \ \tau \ critical \ value \ a, n \ - \ 1 \text{ βαθμοι ελευθεριας}$$

Για την καλύτερη κατανόηση των παραπάνω θα δούμε μια εφαρμογή στην εφοδιαστική αλυσίδα. Ένα πολύ σημαντικό πρόβλημα της εφοδιαστικής αλυσίδας είναι η αυξημένη κατανάλωση καυσίμων των μεταφορικών οχημάτων στις μεταφορές προϊόντων εντός της πόλης λόγο της αυξημένης κίνησης , και των συνέχων στάσεων των οχημάτων σε πολλά

σημεία παράδοσης. Για τον παραπάνω λόγο η μεταφορική εταιρία <<αζρ>> αγόρασε ένα πρόγραμμα δρομολόγησης οπού επιλεγεί τον καλύτερο τρόπο μεταφοράς των προϊόντων σε όλα τα σημεία παράδοσης βρίσκοντας τον συντομότερο και με την λιγότερη κίνηση δρόμο , ταυτόχρονα επανεκπαίδευσε τους οδηγούς των φορτηγών και των μικρών βαν με σεμινάρια οικολογικής οδήγησης(πχ σβήσιμο του κινητήρα του οχήματος την ώρα της στάθμευσης κατά την παράδοση στις αποθήκες των πελάτων). Ο υπεύθυνος δρομολόγησης κατέγραψε την μηνιαία κατανάλωση τον προηγούμενο μηνά 10 οδηγών στα ιδιά δρομολόγια με τα ιδιά οχήματα πριν την εφαρμογή του προγράμματος δρομολόγησης και των σεμιναρίων οικολογικής οδήγησης και θέλει να τα συγκρίνει για να δει εάν η κατανάλωση μετα από αυτές τις αλλαγές των 10 οδηγών μειώθηκε αυτόν τον μηνά.

Για την επίλυση του παραπάνω προβλήματος αρχικά υπολογίζουμε την διαφορά της κατανάλωσης πριν την εφαρμογή των μέτρων αφαιρώντας από αυτήν την κατανάλωση, στην συνέχεια στην στήλη με τις διάφορες υπολογίζουμε τον μέσο ορό και την δειγματική τυπική απόκλιση καθώς οι παραιτήσεις είναι μόνο ένα υποσύνολο του συνόλου των οδηγών οι υποθέσεις του ελέγχου είναι οι παρακάτω

Η0:ΜΕΣΟΣ ΟΡΟΣ ΚΑΤΑΝΑΛΩΣΗΣ ΠΡΙΝ<ΜΕΣΟΣ ΟΡΟΣ ΚΑΤΑΝΑΛΩΣΗΣ ΜΕΤΑ

Η1:ΜΕΣΟΣ ΟΡΟΣ ΚΑΤΑΝΑΛΩΣΗΣ ΠΡΙΝ> ΜΕΣΟΣ ΟΡΟΣ ΚΑΤΑΝΑΛΩΣΗΣ ΜΕΤΑ

Αρά η Η1 μπορεί να γραφτεί και ως εξής: μέσος ορός κατανάλωσης μετα - μέσος ορός κατανάλωσης πριν>0 κατά επέκταση η Η1 γράφεται: ο μέσος ορός της διαφοράς>0

Με βάση τα παραπάνω υπολογίζεται το τ στατιστικό το οποίο είναι ίσο με 6,15 και το τ critical value με α=0,05 καθώς ο έλεγχος είναι μονόπλευρος με 9 βαθμούς ελευθέριας ίσος με 1,83 και αρά εφόσον το τ στατιστικό είναι μεγαλύτερο από το τ critical value λεμέ ότι σε επίπεδο εμπιστοσύνης 95% τα σεμινάρια οικολογικής οδήγησης και το πρόγραμμα δρομολόγησης μείωσαν την κατανάλωση καυσίμου των οδηγών της εταιρείας.(η τυχαία τιμή που αναφέρθηκε παραπάνω στη θεωρία στο παράδειγμα μας είναι μηδέν εάν όμως θέλαμε να δούμε εάν η κατανάλωση πριν με μετα μειώθηκε κατά 10 λίτρα ανά μέσο ορό τότε η Η1 θα γραφόταν μέσος ορός της διαφοράς>10)

οδηγός	κατανάλωση πριν	κατανάλωση μετα	διαφορά
1	300	280	20
2	305	290	15

3	170	165	5
4	327	300	27
5	425	401	24
6	364	350	14
7	280	269	11
8	274	250	24
9	245	240	5
10	298	272	26
μέσος ορός			17,1
τυπική απόκλιση			8,33266664
t στατιστικό			6,156492539
t critical			1,833

Στην συνέχεια θα παρουσιάσουμε τους ελέγχους υποθέσεων για την εύρεση της διαφοράς μεταξύ των μέσων ορών δυο δειγμάτων , όπως είχαμε εξετάσει και για τα διαστήματα εμπιστοσύνης.

Ο τρόπος με τον οποίο υπολογίζεται το ζ στατιστικό για τον έλεγχο υποθέσεων των μέσων ορών δυο πληθυσμών είναι μέσο του παρακάτω τύπου:

$$z\ \text{στατιστικο(για πλυθισμο)} = \frac{\overline{\chi 1} - \overline{x2}}{\sqrt{\frac{\sigma 1\hat{}2}{n1} + \frac{\sigma 2\hat{}2}{n2}}}$$

Οπού χ1 είναι ο μέσος ορός του πρώτου πληθυσμού, χ2 ο μέσος ορός του δευτέρου πληθυσμού σ1,σ2 είναι η πληθυσμιακή τυπική απόκλιση του αντίστοιχου πληθυσμού όπως και το n1 και n2 οπού είναι αντίστοιχα ο αριθμός των παρατηρήσεων του εκάστοτε πληθυσμού.

Οι ανισότητες του συγκεκριμένου ελέγχου υποθέσεων γράφονται με τους παρακάτω τρόπους ανάλογα με το εάν ο έλεγχος είναι δίπλευρος ή μονόπλευρος.

H0: x1(ο μέσος ορός του πρώτου πληθυσμού) = x2(ο μέσος ορός του δευτέρου πληθυσμού)

H1: x1(ο μέσος ορός του πρώτου πληθυσμού) ≠ x2(ο μέσος ορός του δευτέρου πληθυσμού)

Στον παραπάνω δίπλευρο έλεγχο η ανισότητα της σύγκρισης του ζ στατιστικού με την ζ κρίσιμη τιμή της κατανομής είναι η παρακάτω

$$|z\ \text{στατιστικο}| > z\ critical\ value\ a/2$$

Εάν ισχύει το παραπάνω τότε απορρίπτουμε την μηδενική υπόθεση και κάνουμε αποδεκτή την η1, εάν δεν ισχύει τότε δεχόμαστε την μηδενική απόφαση.

Εάν ο έλεγχος είναι μονόπλευρος δηλαδή ο αναλυτής θέλει να ελέγξει εάν ο μέσος ορός ενός από τους δυο πληθυσμούς είναι μικρότερος ή μεγαλύτερος του αλλού.

H0: x1(ο μέσος ορός του πρώτου πληθυσμού) ≥ x2(ο μέσος ορός του δευτέρου πληθυσμού)

H1: x1(ο μέσος ορός του πρώτου πληθυσμού) < x2(ο μέσος ορός του δευτέρου πληθυσμού)

είτε

H0: x1(ο μέσος ορός του πρώτου πληθυσμού) ≤ x2(ο μέσος ορός του δευτέρου πληθυσμού)

H1: x1(ο μέσος ορός του πρώτου πληθυσμού) > x2(ο μέσος ορός του δευτέρου πληθυσμού)

Τότε στην πρώτη περίπτωση η ανισότητα είναι η παρακάτω

$$\zeta\ \text{στατιστικο} <-\ z\ critical\ value\ a$$

Ενώ στην δεύτερη περίπτωση η ανισότητα γράφεται ως εξής:

$$\zeta\ \text{στατιστικο} >\ z\ critical\ value\ a$$

Εάν ισχύουν οι παραπάνω ανισότητες τότε στον αντίστοιχο έλεγχο υποθέσεων ο αναλυτής πρέπει να απορρίψει την μηδενική υπόθεση και να κανείς αποδεκτή την η1.

Όσον αφορά τους ελέγχους υποθέσεων των μέσων ορών δυο δειγμάτων αρχικά ο αναλυτής πρέπει να ελέγξει εάν οι διακυμάνσεις των δειγμάτων είναι ίσες ή άνισες με βάση τον έλεγχο υποθέσεων που αναλύθηκε στην προηγουμένη ενότητα με τα διαστήματα εμπιστοσύνης με την χρήση της κατανομής f.

Αναφορικά με τον έλεγχο υποθέσεων των μέσων ορών δυο δειγμάτων με ίσες διακυμάνσεις η κατανομή που χρησιμοποιείται είναι η student t και για τον υπολογισμό του στατιστικού τ πρέπει πρώτα να έχει υπολογιστεί η pooled διακύμανση των δειγμάτων

με βάση τον παρακάτω τύπο οπού είδαμε και στην ενότητα με τα διαστήματα εμπιστοσύνης

$$sp = \sqrt{\frac{(n1-1)*s1^{2}+(n2-1)*s2\hat{}2}{n1+n2-2}}$$

Και το τ στατιστικό υπολογίζεται με τον παρακάτω τύπο

$$\tau\ \text{στατιστικο(για δειγματα με ισες διακυμανσεις)} = \frac{\overline{\chi 1}-\overline{x2}}{sp*\sqrt{\frac{1}{n1}+\frac{1}{n2}}}$$

Οι ανισότητες για τον παραπάνω έλεγχο υποθέσεων γράφονται με τους παρακάτω τρόπους ανάλογα με το εάν ο έλεγχος είναι μονόπλευρος ή δίπλευρος.

H0: x1(ο μέσος ορός του πρώτου δείγματος) = x2(ο μέσος ορός του δευτέρου δείγματος)

H1: x1(ο μέσος ορός του πρώτου δείγματος) ≠ x2(ο μέσος ορός του δευτέρου δείγματος)

Στον παραπάνω δίπλευρο έλεγχο η ανισότητα της σύγκρισης του τ στατιστικού με την τ κρίσιμη τιμή της κατανομής είναι η παρακάτω

$$|\tau\ \text{στατιστικο}| > \tau\ critical\ value\ n1\ +\ n2\ -\ 2,\ a/2$$

Οπού n1+n2-2 είναι ο αριθμός των βαθμών ελευθέριας της κρίσιμης τιμής οπού n1 και n2 είναι ο αριθμός των παρατηρήσεων του εκάστοτε δείγματος.

Εάν ισχύει το παραπάνω τότε απορρίπτουμε την μηδενική υπόθεση και κάνουμε αποδεκτή την η1, εάν δεν ισχύει τότε δεχόμαστε την μηδενική απόφαση.

Εάν ο έλεγχος είναι μονόπλευρος δηλαδή ο αναλυτής θέλει να ελέγξει εάν ο μέσος ορός ενός από τα δυο δείγματα είναι μικρότερος ή μεγαλύτερος του αλλού τότε οι υποθέσεις γράφονται με τον παρακάτω τρόπο:

H0: x1(ο μέσος ορός του πρώτου δείγματος) ≥ x2(ο μέσος ορός του δευτέρου δείγματος)

H1: x1(ο μέσος ορός του πρώτου δείγματος) < x2(ο μέσος ορός του δευτέρου δείγματος)

είτε

H0: x1(ο μέσος ορός του πρώτου δείγματος) ≤ x2(ο μέσος ορός του δευτέρου δείγματος)

H1: x1(ο μέσος ορός του πρώτου δείγματος) > x2(ο μέσος ορός του δευτέρου δείγματος)

Τότε στην πρώτη περίπτωση η ανισότητα είναι η παρακάτω:

$$\tau\ \text{στατιστικο} < -\ \tau\ critical\ value\ n1 + n2 - 2,\ a$$

Ενώ στην δεύτερη περίπτωση η ανισότητα γράφεται ως εξής:

$$\tau\ \text{στατιστικο} > \tau\ critical\ value\ n1 + n2 - 2,\ a$$

Τέλος στην περίπτωση οπού οι διακυμάνσεις των δειγμάτων είναι άνισες ο τρόπος με τον οποίο υπολογίζεται το τ στατιστικό για τον συγκεκριμένο έλεγχο υποθέσεων είναι μέσο του παρακάτω τύπου:

$$\tau\ \text{στατιστικο(για δειγματα με ανισες διακυμανσεις)} = \frac{\overline{\chi 1} - \overline{x2}}{\sqrt{\frac{s1\text{^}2}{n1} + \frac{s2\text{^}2}{n2}}}$$

Οπού χ1 και χ2 είναι ο μέσος ορός του εκάστοτε δείγματος, n1 και n2 ο αριθμός των παρατηρήσεων αντίστοιχα και τέλος s1 και s2 είναι η δειγματική τυπική απόκλιση των δειγμάτων 1 και 2.

Οι ανισότητες και οι αντίστοιχοι δίπλευροι και μονόπλευροι έλεγχοι υποθέσεων είναι ίδιοι με εκείνους των δειγμάτων με ίσες διακυμάνσεις η μονή διαφορά είναι ότι για τον υπολογισμό των βαθμών ελευθέριας της κρίσιμης τιμής τ βρίσκονται με τον υπολογισμό του παρακάτω τύπου :

$$\text{βαθμοι ελευθεριας(δειγματων με ανισες διακυμανσεις)} = \frac{(\frac{s1^2}{n1} + \frac{s2^2}{n2})\text{^}2}{\frac{(\frac{s1^2}{n1})\text{^}2}{n1-1} + \frac{(\frac{s2^2}{n2})\text{^}2}{n2-1}}$$

Έχοντας αναφέρει τα παραπάνω θα δούμε για άλλη μια φορά ένα απλό παράδειγμα εφαρμογής αυτών στην εφοδιαστική αλυσίδα. Πιο συγκεκριμένα ένα συχνό φαινόμενο στις εταιρείες λιανικών πωλήσεων αλλά και εταιρειών άλλων κλάδων είναι το πρόβλημα επιλογής προμηθευτή. Στο παράδειγμα που θα δούμε παρακάτω η εταιρεία λιανικών πωλήσεων <<χαρ>> έχει συλλέξει τον χρόνο παράδοσης των τελευταίων 15 παραγγελιών της από δυο προμηθευτές της. Πιο αναλυτικά και οι 15 παραγγελίες στάλθηκαν στους προμηθευτές της εταιρείας την ιδιά ημέρα και ώρα οι τιμές και η ποσότητα των προϊόντων ανά παραγγελία ήταν ιδίες και για τους δυο προμηθευτές και στις 15 παραγγελίες το μόνο

που αλλάζει είναι ο χρόνος παράδοσης των προϊόντων από τους προμηθευτές στόχος του στελέχους εφοδιαστικής αλυσίδας είναι με βάση αυτές τις 15 δειγματικές τιμές καθώς η εταιρεία συνεργάζεται με τους προμηθευτές πάνω από 10 έτη και έχει πραγματοποιήσει χιλιάδες παραγγελίες να επιλέξει με ποιον από τους δυο προμηθευτές να συνεχίσει την συνεργασία (θα πρέπει να αναφέρουμε ότι το προϊόν δεν είναι σπάνιο και ότι υπάρχουν πόλοι άλλοι προμηθευτές παγκόσμιος που μπορούν να το προμηθεύσουν στην εταιρεία επίσης δεν υπάρχει καμία έκπτωση στην περίπτωση που αγοραστεί μια μόνο παλέτα σε σύγκριση με την αγορά 20 παλετών από τους προμηθευτές προφανώς σε ένα πραγματικό σενάριο τα παραπάνω δεδομένα κατά πασά πιθανότητα δεν ισχύουν καθώς όσο πιο μεγάλη είναι η παραγγελία ενός ιδίου προϊόντος (κωδικού) και αυτό μπορεί και καλύπτει την καρότσα του φορτηγού,container κλπ. (full truck loaded) τόσο πιο μεγάλη θα είναι και η έκπτωση από τους προμηθευτές καθώς εξοικονομούν χρόνο και χρήματα κατά την διαχείριση της παραγγελίας στην αποθήκη τους αλλά και κατά την πραγματοποίηση του δρομολογίου καθώς το φορτηγό είναι πληροίς φορτωμένο με προϊόντα ενός πελάτη και αρά αποφεύγονται ενδιάμεσες στάσεις σε άλλους πελάτες Επίσης καλό είναι να αναφέρουμε ότι στην περίπτωση που το προϊόν ήταν σπάνιο δηλαδή δεν υπήρχαν πόλοι προμηθευτές που να το παράγουν και δεν υπήρχαν αλλά παρόμοια υποκατάστατα του και ταυτόχρονος ήταν σημαντικό κομμάτι για την παραγωγή ενός τελικού προϊόντος από το οποίο προέρχεται μεγάλο μέρος των συνολικών εσοδών της εταιρείας τότε είναι καλύτερο να μην διακοπεί η συνεργασία με κανέναν προμηθευτή λόγο των παραπάνω , ένα παράδειγμα τέτοιου προϊόντος είναι οι επεξεργαστές φορητών υπολογιστών και οι κάρτες γραφικών καθώς παράγονται κυρίως από τρεις εταιρείες την amd την intel και την Nvidia αρά ο υπεύθυνος εφοδιαστικής αλυσίδας παραδείγματος χάρη της dell θα πρέπει να κρατήσει καλές επαφές και με τους τρεις αυτούς προμηθευτές ώστε να αποφευχθεί πιθανή έλλειψη αυτών των προϊόντων που θα έχει ως αποτέλεσμα χαμένες πωλήσεις για την dell.)

Τα δεδομένα που συνέλεξε ο υπεύθυνος της εφοδιαστικής αλυσίδας του συγκεκριμένου προϊόντος φαίνονται στον παρακάτω πίνακα, όπως αναφέραμε τα δεδομένα είναι δυο δείγματα 15 παρατηρήσεων το καθένα με τους χρόνους παράδοσης της εκάστοτε παραγγελίας η οποία αποτελείται από τον ίδιο όγκο προϊόντων και τιμή. Αρχικά ο υπεύθυνος υπολογίζει τον μέσο ορό και την διακύμανση των ημέρων παράδοσης και των δυο προμηθευτών και παρατηρεί ότι ο προμηθευτής 1 έχει μικρότερο χρόνο παράδοσης κατά μέσο ορό σε σχέση με τον προμηθευτή 2 αλλά ταυτόχρονος έχει μεγαλύτερη διακύμανση σε σχέση με τον δεύτερο οπότε δεν μπορεί να οδηγηθεί σε ξεκάθαρο

συμπέρασμα για το ποιος ολοκληρώνει τις παραγγελίες γρηγορότερα. Οπότε αποφασίζει να κάνει έναν μονόπλευρο έλεγχο υποθέσεων έχοντας ως μηδενική απόφαση ότι ο προμηθευτής 1 παραδίδει τις παραγγελίες σε περισσότερές ημέρες σε σχέση με τον 2 και εναλλακτική υπόθεση ότι ο προμηθευτής 1 είναι γρηγορότερος(οι ημέρες παράδοσης των παραγγελιών είναι λιγότερες) στην ολοκλήρωση των παραγγελιών σε σχέση με τον 2.

H0: $\chi 1 > \chi 2$ (ο μέσος χρόνος αποστολής των παραγγελιών του προμηθευτή 1 είναι μεγαλύτερος σε σχέση με τον αντίστοιχο του προμηθευτή 2)

H0: $\chi 1 < \chi 2$ (ο μέσος χρόνος αποστολής των παραγγελιών του προμηθευτή 1 είναι μικρότερος σε σχέση με τον αντίστοιχο του προμηθευτή 2)

Για να επιλέξει όμως τον τρόπο υπολογισμού του τ στατιστικού πρέπει πρώτα να κάνει τον έλεγχο διακυμάνσεων. Στην συνέχεια πρέπει να υπολογίσει το f στατιστικό(1,49) και να το συγκρίνει με την κρίσιμη τιμή της κατανομής f οπού είναι ίση με 2,48(α=0,05 και οι βαθμοί ελευθέριας είναι 14 και 14 αντίστοιχα).

H0: s1^2=s2^2 (Οι διακυμάνσεις των δύο προμηθευτών είναι ίσες)

H1: s1^2≠s2^2 (Οι διακυμάνσεις των δύο προμηθευτών είναι διαφορετικές)

Και εφόσον 1,49<2,48 τότε δεν απορρίπτει την μηδενική απόφαση και αρά οι διακυμάνσεις είναι ίσες. Οπότε για τον υπολογισμό του τ στατιστικού πρέπει να βρεθεί το sp. Το τ στατιστικό είναι ίσο με -4,47 και η κρίσιμη τιμή από την κατανομή τ είναι ίση με -1,701 (28 βαθμοί ελευθέριας και α=0,05) και ο έλεγχος είναι ο παρακάτω -4,47<-1,701 το οποίο ισχύει και αρά ο υπεύθυνος μπορεί να απορρίψει την μηδενική απόφαση και αρά να σταματήσει την συνεργασία της εταιρείας με τον προμηθευτή 2 (καθώς όπως είχαμε πει το μόνο κριτήριο διαχωρισμού των δυο είναι ο χρόνος παράδοσης των παραγγελιών).

παραγγελίες	προμηθευτής 1(ημέρες αποστολής παραγγελίας)	προμηθευτής 2 (ημέρες αποστολής παραγγελίας)
1	10	12
2	12	14
3	9	13
4	11	12
5	13	15
6	14	16
7	10	14
8	12	13

9	11	12
10	10	14
11	12	13
12	13	15
13	9	12
14	10	14
15	12	13
αριθμός παρατηρήσεων	15	15
μέσος ορός	11,2	13,46666667
διακύμανση	2,314285714	1,552380952
τυπική απόκλιση	1,521277659	1,245945806
f στατιστικό	1,490797546	
f κρίσιμη τιμή	2,48	
sp	1,39	
βαθμοί ελευθέριας	28	
τ στατιστικό	-4,47	
τ κρίσιμη τιμή	-1,701	

Ολοκληρώνοντας αυτή την ενότητα θα πρέπει να αναφέρουμε ότι οι παραπάνω υπολογισμοί έγιναν μέσο του google sheets (το αντίστοιχο πρόγραμμα υπολογιστικών φύλλων της google σαν το excel της Microsoft) παρόλα αυτά είναι ευκολότερη και γρηγορότερη η εκτέλεση των παραπάνω ελέγχων μέσο προγραμμάτων όπως η python και η R , στα συγκεκριμένα όμως προγράμματα υπολογίζεται αυτόματα για τον εκάστοτε έλεγχο υποθέσεων αντί για την κρίσιμη τιμή της κάθε κατανομής το p-value. Το p-value εκφράζει την πιθανότητα να παρατηρήσουμε μια τιμή του στατιστικού ελέγχου ίση ή πιο ακραία από αυτή που προκύπτει από τα δεδομένα του δείγματος, υπό την προϋπόθεση ότι η μηδενική υπόθεση H0 είναι αληθής. Το p value υπολογίζεται με βάση την κατανομή του στατιστικού ελέγχου πχ την f αν πρόκειται για έλεγχο διακυμάνσεων, και αντιστοιχεί στο εμβαδόν της περιοχής κάτω από την καμπύλη της κατανομής που βρίσκεται πέρα από την παρατηρούμενη τιμή του στατιστικού δείκτη, εάν το p value είναι μικρότερο από το επιλεγμένο α τότε ο αναλυτής απορρίπτει την μηδενική απόφαση εάν όμως το p value είναι μεγαλύτερο από το α που έχει επιλέξει ο αναλυτής τότε πρέπει να αποδεχτεί την μηδενική υπόθεση.

3. ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ

1.1 ΧΡΙΣΕΙΣ ΒΑΣΕΩΝ ΔΕΔΟΜΕΝΩΝ ΣΤΗΝ ΕΦΟΔΙΑΣΤΙΚΗ ΑΛΥΣΙΔΑ

Σε αυτό το κεφάλαιο θα αναπτύξουμε συνοπτικά τον ορισμό των σχεσιακών βάσεων δεδομένων και τα βασικά τους χαρακτηριστικά, στην συνέχεια θα αναλύσουμε την χρησιμότητα τους στην εφοδιαστική αλυσίδα και στην τελευταία ενότητα θα αναπτύξουμε μια απλή βάση δεδομένων για ένα ηλεκτρονικό κατάστημα πώλησης hardware desktop υπολογιστών.

Αναφορικά με τον ορισμό των βάσεων δεδομένων βάση δεδομένων είναι μια συλλογή από συσχετιζόμενα δεδομένα εύκολα προσπελάσιμα και με δυνατότητες ανανέωσης, διαγραφής και τροποποίησης. Πιο συγκεκριμένα οι σχεσιακές βάσεις δεδομένων βασίζονται στην θεωρία των μαθηματικών σχέσεων και πρώτοεμφανιστήκαν την δεκαετία του 1970 από τον προγραμματιστή Edgar f Codd. Σε μια σχεσιακή βάση δεδομένων τα δεδομένα αποθηκεύονται σε πίνακες , οπού κάθε πίνακας αποτελείται από γραμμές και στήλες (attributes), κάθε μια γραμμή αντιπροσωπεύει μια εγγραφή ενώ κάθε στήλη αντιστοιχεί σε ένα χαρακτηριστικό αυτής της εγγραφής. Κάθε πίνακας μπορεί να ονομαστεί και ως οντότητα , οι οντότητες αντιπροσωπεύουν πραγματικές καταστάσεις στον φυσικό κόσμο για παράδειγμα σε μια εταιρεία logistics υπάρχουν πολλές οντότητες για παράδειγμα η οντότητα πελάτες οπού έχει ως γνωρίσματα το όνομα του πελάτη την διεύθυνση του το τηλέφωνο του κλπ. μια άλλη οντότητα είναι οι παραγγελίες που κάνουν οι πελάτες οπού έχει μέσα γνωρίσματα όπως η κατάσταση της παραγγελίας ,το συνολικό ποσό ,την ημερομηνία της παραγγελίας και αλλά , άλλη οντότητα θα μπορούσε να είναι η αποστολή δηλαδή ένας πίνακας με χαρακτηριστικά όπως η ημερομηνία αποστολής ,η ημερομηνία παράδοσης ,η κατάσταση της παραγγελίας και ο οδηγός που κάνει την παράδοση ακόμα και ο τύπος της μεταφοράς για παράδειγμα αν είναι με αεροπλάνο, με φορτηγό ή με πλοίο, επίσης μια ακόμα σημαντική οντότητα είναι οι εργαζόμενοι στην εταιρεία οπού θα είχαν γνωρίσματα όπως το ,τηλέφωνο ,το όνομα ,το επώνυμο τους ,τον μισθό τους ,το υπόλοιπο των αδειών τους , το τι είδους εργασία ή σε πιο τμήμα ανήκουν για παράδειγμα οδηγός, λογιστής, γραμματέας picker κλπ. ,το ωράριο των εργαζομένων και αλλά , επίσης ένα σημαντικό κομμάτι των εταιρειών logistics είναι τα οχήματα τα οποία έχουν γνωρίσματα όπως ο αριθμός κυκλοφορίας τους ,την χωρητικότητα τους σε

κιλά ή σε τόνους ,την μάρκα τους ,την κατηγορία που ανήκουν πχ ένα φορτηγό μπορεί να διαθέτει ψυγείο και αρά όταν υπάρξει ένα αίτημα μεταφοράς κρεάτων δηλαδή ευαίσθητων προϊόντων η βάση δεδομένων να εμφανίζει μνήμα ή να αναγκάζει τον packer να τοποθετήσει τα προϊόντα σε φορτηγό ψυγείο, άλλες οντότητές θα μπορούσαν να είναι η αποθήκες με γνωρίσματα όπως την γεωγραφική τους τοποθεσία ,την χωρητικότητα τους σε όγκο ,η οποία θα ενημερώνεται αυτόματα όταν τοποθετούνται νέα αποθέματα , το όνομα του υπευθύνου της αποθήκης οι θέσεις των ραφιών , ή γενικότερα των αποθηκευτικών θέσεων . Επίσης η πιο σημαντική οντότητα σε μια εταιρεία logistics είναι τα προϊόντα και γενικότερα τα αποθέματα τα οποία μπορούν να έχουν χαρακτηριστικά όπως το όνομα, την περιγραφή ,το βάρος ,τις διαστάσεις τους, την κατηγορία που ανήκουν για παράδειγμα αν είναι εύθραυστα αν είναι προϊόντα ψυγείου όπως τρόφιμα ,την αξία τους ,την θέση τους μέσα στην αποθήκη ,το lot ,η ημερομηνία εισαγωγής τους στην αποθήκη , η ημερομηνία εξόδου τους από την αποθήκη ,το barcode η διαθέσιμη ποσότητα κλπ. και τέλος μια ακόμα σημαντική οντότητα είναι εκείνη των προμηθευτών της εταιρείας οπού συνήθως έχει γνωρίσματα όπως το όνομα της εταιρείας του προμηθευτή ,την διεύθυνση του ,το τηλέφωνο το email το ΑΦΜ τα ,προϊόντα που προμηθεύει και αλλα. Η εκάστοτε εταιρεία τροποποιεί ή κατασκευάζει την βάση δεδομένων της με βάση τις ανάγκες που η ιδιά έχει. Η διαδικασία κατασκευής των βάσεων δεδομένων είναι πολύ απαιτητική και οι περισσότερες εταιρείες απευθύνονται σε άλλες εταιρείες για την κατασκευή τους όπως για παράδειγμα την oracle, καθώς οι περισσότερες βάσεις δεδομένων διαχωρίζονται σε υποκατηγορίες δηλαδή η βάση δεδομένων για την αποθήκη ονομάζεται wms (Wherehouse management system) για την διαχείριση των πελατών υπάρχει άλλο σύστημα που ονομάζεται crm (customer relationship management) , υπάρχουν βάσεις δεδομένων που εστιάζουν αποκλειστικά και μόνο για την διαχείριση της μισθοδοσίας των εργαζομένων και του λογιστήριού , ή εστιάζουν στην παραγωγή ενός προϊόντος με προμήθειες αποθεμάτων ημικατεργασμενα ύλες κλπ. Όταν πολλές βάσεις δεδομένων ‘’ενωθούν’’ μεταξύ τους δηλαδή υπάρξει δικτύωση όλων των παραπάνω συστημάτων τότε το λογισμικό πλέον ονομάζεται ERP (enterprise resource planning) ένα από τα πιο δημοφιλή λογισμικά ERP είναι το sap.

Ένα σημαντικό κομμάτι των βάσεων δεδομένων και πιο συγκεκριμένα των γνωρίσματων των οντοτήτων είναι τα πρωτεύον κλειδιά. Το πρωτεύον κλειδί είναι ένα γνώρισμα του πίνακα (ή και συνδυασμός γνωρίσματων) οπού προσδιορίζει με μοναδικό τρόπο μια εγγραφή και σε καμία περίπτωση δεν μπορεί να μην έχει τιμή(δηλαδή να είναι null). Για

παράδειγμα το πρωτεύον κλειδί της οντότητας οχήματα θα μπορούσε να είναι ο αριθμός πινακίδας του οχήματος , της οντότητας εργαζόμενοι θα μπορούσε να είναι ο αριθμός ταυτότητας του εργαζομένου, αυτό που ξεχωρίζει το πρωτεύον κλειδί από τα υπόλοιπα γνωρίσματα είναι η μοναδικότητα δηλαδή δεν θα μπορούσε να ήταν πρωτεύον κλειδί το όνομα ενός εργαζομένου καθώς μπορεί να υπήρχαν πόλοι εργαζόμενοι με το ίδιο όνομα ή ακόμα και επώνυμο. Όταν ένα πρωτεύον κλειδί τοποθετείται ως γνώρισμα σε έναν άλλο πίνακα που έχει ήδη πρωτεύον κλειδί το πρώτο ονομάζεται ξένο κλειδί και υπάρχει ώστε να διασφαλίζεται η ακεραιότητα των δεομένων μεταξύ των δυο ή και περισσοτέρων οντοτήτων, ένα απλό παράδειγμα ξένου κλειδιού είναι το πρωτεύον κλειδί ενός προϊόντος στον πίνακα αποθήκη.

Όσον αφορά τους τύπους συσχετίσεων των πινάκων αυτοί είναι τρεις η πρώτη κατηγορία είναι η ένα προς ένα οπού κάθε εγγραφή του πρώτου πίνακα αντιστοιχεί σε ακριβώς μια εγγραφή του δευτέρου πίνακα και προφανώς ισχύει και το αντίστροφο , τέτοιου είδους συσχέτιση δεν εφαρμόζεται συχνά καθώς θα μπορούσαν οι πληροφορίες του δευτέρου πίνακα να τοποθετηθούν απευθείας ως γνωρίσματα του πρώτου πίνακα. Η δεύτερη κατηγορία είναι η συσχέτιση ένα προς πολλά όπου κάθε εγγραφή του πρώτου πίνακα συσχετίζεται με πολλές εγγραφές του δευτέρου πίνακα αλλά οι εγγραφές του δευτέρου πίνακα αντιστοιχούν μόνο σε μια εγγραφή του πρώτου, για παράδειγμα στον πίνακα αποθήκη μπορεί να υπάρχουν πολλά αποθέματα αλλά κάθε απόθεμα ανήκει σε μια μόνο αποθήκη αρά το πρωτεύον κλειδί του πίνακα αποθέματα έχει τοποθετηθεί ως γνώρισμα στον πίνακα αποθήκη. Και τέλος η τρίτη συσχέτιση μεταξύ των πινάκων είναι εκείνη του πολλά προς πολλά όπου κάθε εγγραφή του πρώτου πίνακα συσχετίζεται με πολλές εγγραφές του δευτέρου πίνακα και το αντίστροφο, για να υλοποιηθεί όμως η παραπάνω σχέση θα πρέπει να δημιουργηθεί ένας τρίτος ενδιάμεσος πίνακας όπου θα έχει το δικό του πρωτεύον κλειδί και θα έχει ως γνωρίσματα τα πρωτεύοντα κλειδιά των άλλων δυο πινάκων. Ένα παράδειγμα στην εταιρεία logistics είναι οι σχέση μεταξύ προμηθευτή και αποθεμάτων καθώς πόλοι προμηθευτές μπορούν να προμηθεύουν με πολλά διαφορετικά προϊόντα την εταιρεία και ταυτόχρονα πολλά διαφορετικά αποθέματα μπορούν να προέρχονται από πολλούς διαφορετικούς προμηθευτές.

Στην συνέχεια θα αναλύσουμε τις κατηγορίες των δεδομένων που μπορούν να εισαχθούν σε μια βάση δεδομένων. Αν ο αναλυτής θέλει ένα γνώρισμα να παίρνει αποκλειστικά και μόνο ακεραίες θετικές ή αρνητικές τιμές μπορεί να χρησιμοποιήσει την κατηγορία TINYINT, SMALLINT, MEDIUMINT , INT και BIGINT . Εάν το γνώρισμα χρειάζεται

να έχει δεκαδικά ψηφιά τότε πρέπει να επιλέγουν κατηγορίες δεδομένων όπως το float , double και decimal. Για γνωρίσματα που περνούν διάφορους χαρακτήρες όπως γράμματα και σύμβολα , χρησιμοποιούνται οι τύποι δεδομένων όπως ο char , varchar tinytext , text , mediumtext και longtext. Για γνωρίσματα τα οποία θέλουμε να εισέρχονται ημερομηνίες χρησιμοποιούνται οι τύποι δεδομένων όπως date όπου δέχεται της ημερομηνίες ως χρονιά/ μήνας / μέρα, η datetime όπου δέχεται τα δεδομένα ως χρονιά/μήνας/μέρα ώρες/λεπτά/δευτερόλεπτα , και υπάρχει και η timestamp όπου παίρνει την ημερομηνία τις χρονικής στιγμής που γίνεται η εγγραφή των δεδομένων. Η διαφορά μεταξύ των ιδίων τύπων δεδομένων έγκειται στον χώρο που θα δεσμεύσει η βάση δεδομένων για την αποθήκευση της πληροφορίας, δηλαδή αν όντως ο δημιουργός της βάσης δεδομένων γνωρίζει ότι ένα γνώρισμα δεν θα λάβει τιμή πολύ μεγάλη για παράδειγμα η ηλικία ενός ατόμου πχ από 0 έως 120 τότε μπορεί να επιλέξει να την αποθήκευση σε τύπο δεδομένων TINYINT αντί για INT το ίδιο μπορεί να συμβεί και με τους χαρακτήρες. Προφανώς όταν κατασκευάζονται μικρές βάσεις για μικρές επιχειρήσεις δεν υπάρχει τόσο μεγάλο πρόβλημα αποθήκευσης των δεδομένων όμως για πολύ μεγάλες επιχειρήσεις όπως η google η κάθε επιλογή αυτή μετράει καθώς αποθηκεύεται άπειρος όγκος πληροφορίας καθημερινά.

Έχοντας αναλύσει τα βασικά χαρακτηριστικά των βάσεων δεδομένων θα εστιάσουμε σε κάποιες απλές εντολές που μπορεί να χρησιμοποιήσει ο χρηστής μέσα σε αυτές. Αρχικά η εντολή για την δημιουργία μιας βάσης δεδομένων είναι η create database αν θέλουμε να την διαγράψουμε χρησιμοποιούμε την εντολή drop database και το όνομα που της έχουμε δώσει , για την δημιουργία των πινάκων χρησιμοποιούμε την εντολή create table και μέσα σε αυτή ορίζουμε το όνομα και την κατηγορία των δεδομένων που θα δέχεται το κάθε γνώρισμα του πίνακα. Η εισαγωγή εγγράφων μέσα σε έναν πίνακα γίνεται μέσο της εντολής insert into (όνομα πίνακα) values , ενώ για την διαγραφή τους χρησιμοποιείται η εντολή delete εάν επίσης ο χρηστής θέλει να αλλάξει τα γνωρίσματα ενός πίνακα για παράδειγμα να προσθέσει στον πίνακα εργαζόμενοι το γνώρισμα τηλέφωνο μπορεί να το πραγματοποιήσει με την εντολή alter table. Στην συνέχεια έχοντας δημιουργήσει την βάση δεδομένων με τους αντίστοιχους πίνακες και έχοντας εισάγει σε αυτούς μερικές εγγραφές ο χρηστής μπορεί να αποκτήσει πρόσβαση στις εγγραφές του εκάστοτε πίνακα μέσο της εντολής select, επίσης τοποθετώντας και την εντολή where μπορεί να επιλέξει συγκείμενες εγγραφές από ένα πίνακα με βάση κάποιο κριτήριο πχ την εμφάνιση των ονομάτων των

εργαζομένων που έχουν μισθό άνω των 1000 ευρώ , μέσα στην εντολή where μπορούν να μπουν και τελεστές όπως το or(να ισχύει η μια συνθήκη ή άλλη) και το and (δηλαδή να ισχύουν παραπάνω από μια συνθήκη ταυτόχρονα). Αν ο αναλυτής θέλει να ομαδοποιήσει τα δεδομένα και να κάνει πράξεις όπως το sum , count , avg με τις εγγραφές ενός ή παραπάνω πινάκων θα πρέπει να χρησιμοποιήσει την εντολή group by και αν μέσα σε αυτήν θέλει να ορίσει κάποιες συνθήκες με τις οποίες θα εμφανίζονται οι ομαδοποιημένες εγγραφές θα πρέπει να κάνει χρήση της εντολής having (λειτουργεί με τον ίδιο ακριβός τρόπο όπως η where αλλά για ομαδοποιημένες εγγραφές). Στην περίπτωση όπου ο αναλυτής θέλει να πάρει πληροφορίες μεταξύ δυο πινάκων για παράδειγμα θέλει να δει τα ονόματα των πελατών που έκαναν παραγγελία εχθές θα πρέπει να ενώσει αυτούς τους δυο πίνακες . για να επιτευχθεί το παραπάνω θα πρέπει πρώτα να έχουν χρησιμοποιηθεί σωστά οι συνδέσεις μεταξύ των πινάκων και ταυτόχρονα να έχουν τοποθετηθεί σωστά τα πρωτεύοντα κλειδιά του κάθε πίνακα σε έναν τρίτο πίνακα αν η σύνδεση είναι πολλά προς πολλά ή ως γνώρισμα σε τον άλλον πίνακα αν είναι ένα προς πολλά. Η εντολή που χρησιμοποιείται για την ένωσή των πινάκων είναι η join , η οποία έχει 4 βασικές εκδοχές η πρώτη είναι το inner join το οποίο επιστρέφει μόνο τις εγγραφές όπου υπάρχει αντιστοιχία στις σχετικές στήλες και των δύο πινάκων ταυτόχρονα. Η επόμενη είναι η left outer join η οποία επιστρέφει όλες τις εγγραφές από τον αριστερό πίνακα και τις αντιστοιχισμένες εγγραφές από τον δεξιό πίνακα , στην περίπτωση που δεν υπάρχουν αντιστοιχημένες εγγραφές στις στήλες του δεξιού πίνακα εμφανίζεται null τιμές , το ίδιο συμβαίνει αλλά στον αριστερό πίνακα στην περίπτωση του right outer join. Και η τελευταία ένωση είναι αυτή του full outer join όπου επιστέφει όλες τις εγγραφές και του δεξιού και του αριστερού πίνακα και απλά στην περίπτωση που δεν έχουν αντιστοιχημένες τις εγγραφές τους τοποθετείται null τιμή.

Σε αυτό το σημείο έχοντας αναλύσει τις βασικές μεθοδολογίες για την ανάπτυξη των βάσεων δεδομένων θα εστιάσουμε σε μερικά από τα πλεονεκτήματα της χρήσης τους στην εφοδιαστική αλυσίδα. Ένα από τα βασικότερα πλεονέκτημα χρήσης βάσεων δεδομένων στην εφοδιαστική αλυσίδα είναι ότι βελτιώνει την διαχείριση των πληροφοριών στην επιχείρηση καθώς πλέον όλα τα δεδομένα είναι αποθηκευμένα μέσα σε αυτή και όχι σε χαρτιά τα οποία είναι πολύ πιθανό να χαθούν. Επίσης μέσο της χρήσης των πρωτευόντων κλειδιών και κατά επέκταση των ξένων κλειδιών εξασφαλίζεται η ακεραιότητα και η συνοχή των δεδομένων, επίσης μέσα από απλά ερωτήματα μπορεί κάνεις να έχει

πρόσβαση σε μεγάλης ποσότητας δεδομένα μέσα σε πολύ μικρό χρονικό διάστημα για παράδειγμα το τι είδους προϊόντα μεταφέρονται με το χ φορτηγό και πια είναι η γεωγραφική του θέση και τρέχουσα ταχύτητα του. Από εκεί και πέρα μέσο της χρήσης ενός crm η επιχείρηση μπορεί να συλλέξει διαφορά χαρακτηριστικά των πελατών όπως τα email ,τα προϊόντα που προμηθευτήκαν στις προηγούμενες αγορές τους και γενικότερα τις καταναλωτικές τους συνήθεις με αποτέλεσμα η επιχείρηση να μπορέσει να προβλέψει καλυτέρα την αναμενομένη ζήτηση και με αυτό τον τρόπο να βελτιστοποιηθεί η εφοδιαστική αλυσίδα καθώς η επιχείρηση θα γνωρίσει ποσά προϊόντα να παράγει ή να παραγγείλει από τους προμηθευτές της και κατά επέκταση να αποφασίσει εάν θέλει να κάνει μεγάλες παραγγελίες όπου θα τις εξασφαλίσουν εκπτώσεις από τους προμηθευτές ή μια στρατηγική just in time με μικρές και συχνές παραγγελίες , προφανώς το ίδιο συμβαίνει και στην περίπτωση όπου παράγει το προϊόν η ιδιά η εταιρεία όπου και εκεί μπορεί να διαχειριστεί καλυτέρα τις προμήθειες πρώτων υλών και να οργανώσει με βέλτιστο τρόπο την παραγωγή της ανά εργοστάσιο. Στην συνέχεια μέσο της ανάλυσης των καταναλωτικών συνήθειων μπορούν να βγουν συμπεράσματα για το που ακριβός εκθηλύνεται υψηλή ζήτηση και κατά επέκταση να οργανωθεί καλυτέρα η αποθήκη επιλέγοντας μεγαλύτερα οχήματα μεταφοράς , τοποθετώντας τα προϊόντα που κινούνται πιο συχνά σε θέσης κοντά στους χώρους φόρτωσης των προϊόντων(χρήση abc ανάλυσης) και επιλέγοντας τον βέλτιστο τρόπο μεταφοράς τους .Επιπρόσθετα ένα σύστημα crm βοηθάει στην ταυτόχρονη αποστολή ενημερωτικών φυλλαδίων και εκπτώσεων μέσο email στους πελάτες επίσης μέσο της ανάλυσης των καταναλωτικών τους συνήθειων το πρόγραμμα μπορεί να προτείνει εξειδικευμένες εκπτώσεις για τον εκάστοτε πελάτη. Τέλος μετά την ολοκλήρωση της αγοράς του πελάτη το crm βοηθάει στο after sales marketing και γενικότερα στην εξυπηρέτηση και στην αντιμετώπιση πιθανών προβλημάτων που ίσως προκύψουν. Ταυτόχρονα βάσεις δεδομένων όπου εστιάζουν στην διαχείριση αποθεμάτων όπως τα wms προσφέρουν επίσης πολλά πλεονεκτήματα στης εταιρίες βελτιώνοντας τον τρόπο λειτουργείας των αποθηκών πιο συγκεκριμένα μέσα από ένα wms ο διαχειριστής της αποθήκης μπορεί να αποκτήσει πληροφορίες όπως την ποσότητα του κάθε αποθέματος μέσα στην αποθήκη αποφεύγοντας με αυτό τον τρόπο την πιθανότητα ελλείψεων ή από την άλλη πλευρά υπερ αποθεματοποίησης. Επίσης μέσο του wms μπορεί να γίνει διαχείριση του κάθε ένα εργαζομένου ξεχωριστά δηλαδή ο κάθε picker (εργαζόμενος που συλλεγεί τα προϊόντα παραγγελιών) έχει την δική του λίστα συλλογής προϊόντων με βάση τα χαρακτηριστικά του και τα δεδομένα που έχει συλλέξει για αυτόν το wms για

παράδειγμα εάν το wms εκτιμήσει ότι με βάση προηγούμενες παρατηρήσεις ότι ένας εργαζόμενος είναι πιο αποδοτικός στην συλλογή μικρών τεμαχίων από έναν άλλο θα επιλέξει τον πρώτο όταν στην παραγγελία πρέπει να συλλεχθούν προϊόντα μικρού όγκου. Επίσης το wms συντονίζει τους εργαζομένους για παράδειγμα αν ένα προϊόν είναι τοποθετημένο σε μια παλέτα την οποία δεν μπορεί να προσεγγίσει ένας picker τότε ενημερώνει έναν χειριστή παλετοφορου ώστε εκείνος να κατεβάσει την παλέτα. αποτέλεσμα των παραπάνω είναι να εξοικονομείται χρόνος καθώς το wms επιλεγεί τον βέλτιστο τρόπο συλλογής των προϊόντων , η μείωση των λαθών και ατυχημάτων. Επιπρόσθετα μέσα από ένα wms μπορεί να γίνει διαχείριση του στόλου των μεταφορικών μέσων της εταιρείας και να επιλεγεί το καταλληλότερο για την εκάστοτε μεταφορά , πχ επιλογή φορτηγού ψυγείου για μεταφορά τροφίμων , επιλογής μικρών van για παραδόσεις μικρού όγκου παραγγελιών μέσα στην πόλη. Τέλος όσον αφορά τα αποθέματα το wms γνωρίζει τα χαρακτηριστικά τους επιτρέποντας την αποδοτικότερη διαχείριση τους για παράδειγμα γνωρίζοντας το μέγεθος και το βάρος τους τα τοποθετεί στα καταλληλά ράφια , γνωρίζοντας την ημερομηνία λήξης πχ των τροφίμων μπορούν να επιλέγουν στρατηγικές διαχείρισης όπως fifo (first in first out) και lifo(last in first out).

1.2 ΑΝΑΠΤΥΞΗ ΑΠΛΗΣ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ ΓΙΑ ΜΙΚΡΟ ΚΑΤΑΣΤΗΜΑ ΛΙΑΝΙΚΩΝ ΠΩΛΗΣΕΩΝ

Σε αυτήν την ενότητα θα αναπτύξουμε μια απλή βάση δεδομένων στην access για την διαχείριση των παραγγελιών των αποθεμάτων ,των προμηθευτών και των πελατών μιας μικρής επιχείρησης εμπορίας hardware συστημάτων desktop υπολογιστών. Ξεκινάμε δημιουργώντας την οντότητα πελάτες. Οι πελάτες έχουν ως πρωτεύον κλειδί το γνώρισμα id το οποίο αυξάνεται αυτόματα με την εισαγωγή νέων εγγράφων στην συγκεκριμένη οντότητα , επίσης έχει γνώρισμα το όνομα , το επώνυμο του πελάτη ,το κινητό τηλέφωνο ,το email και στοιχεία της διεύθυνσης κατοικίας του όπως την πόλη στην οποία κατοικεί και τον ταχυδρομικό κώδικα επιπλέον διακρατατε και η ημερομηνία που ο πελάτης αγόρασε πρώτη φορά από το κατάστημα, παρακάτω παρουσιάζεται ο κώδικας δημιουργίας της οντότητας.

customers

Όνομα πεδίου	Τύπος δεδομένων
id	Αυτόματη Αρίθμηση
first_name	Σύντομο κείμενο
last_name	Σύντομο κείμενο
phone	Σύντομο κείμενο
email	Σύντομο κείμενο
city	Σύντομο κείμενο
address	Σύντομο κείμενο
zip_code	Σύντομο κείμενο
customer_since	Ημερομηνία/Ώρα
gender	Σύντομο κείμενο

```
create table customers (
id serial ,
first_name varchar(50) not null,
last_name varchar(50) not null,
phone varchar(50) not null unique,
email varchar(50) not null unique,
city varchar(50) not null,
address varchar(100) not null,
zip_code varchar(30) not null,
primary key (id)
);
alter table customers add column customer_since date;
```

customers

ID	first_name	last_name	phone	email	city	address	zip_code	customer_s	gender
1	kosats	ninos	23232323	kostas@yahoo.gr	athens	agios dimitrios	99999	11/5/2020	male
2	nikos	papas	232334343	nikos@yahoo.gr	patra	agias triados	54567	11/5/2020	male
3	Michal	Jereatt	894-697-2618	mjereatt2@utexas.ed	Wuyang	776 Acker Alley	454542	11/5/2020	male
4	Barb	Comoletti	607-187-8242	bcomoletti3@upenn.e	Saray	728 Meadow Vall	98887876	11/5/2020	male
5	bill	papas	2109318389	pap@yahoo.gr	athens	agios dimitrios	123213	24/5/2020	male
6	alex	kapas	2109344211	kap@yahoo.gr	athens	abelokipoi	12123	24/5/2020	male
7	nikos	papanastasiou	6937371212	nikpap@hotmail.com	athens	agia paraskeui	121344	24/5/2020	male
8	maria	nikolopoulou	6943824832	marianick@gmail.cor	patra	agiou trifonos	256000	24/5/2020	female
9	giorgos	disketas	6932331223	gdisc@hotmail.com	patra	aigiou	264123	24/5/2020	male
10	nikoleta	papaueodorou	6923344511	nickpapade8eodorou	thessaloniki	dragoumi ionos	2134355	24/5/2020	female
11	vaso	kipriou	2107463745	vasokip@hotmail.cor	thessaloniki	larisis 25	3234343	24/5/2020	female
12	mixalis	poupakis	21094394934	mixpoup@gmail.com	thessaloniki	neapoli	323166	24/5/2020	male
13	kostas	nikolaou	692343432	k.nick@hotmail.com	thessaloniki	skra	3232377	24/5/2020	male
14	nikoleta	aggelopoulou	6934737423	aggnick@hotmail.cor	thessaloniki	skra	898978	24/5/2020	female
15	ilias	papandreou	2104546567	ilias2000@hotmail.cc	thessaloniki	neapoli	343289	24/5/2020	male
16	anna	vixou	23121434	annavix@hotmail.cor	patra	akratas	43442	24/5/2020	female
17	eleni	xristou	21047897664	helenxr@yahoo.gr	athens	palio faliro	21344	24/5/2020	female
18	xara	argirou	213323345	xaraar@hotmail.com	athens	nea smirni	45780	24/5/2020	female
19	stelios	papanikolaou	2104565657	steliospap@yahoo.gr	athens	ilioupoli	17769	24/5/2020	male
20	sofia	pissa	210887669	sofip@gmail.com	athens	petralona	3434667	24/5/2020	female

Εικόνα 6: οντότητα πελάτες

Εν συνέχεια δημιουργούμε τον πίνακα που θα περιλαμβάνει τα προϊόντα που θα πουλάει και θα εμπορεύεται το κατάστημα. Για άλλη μια φορά το πρωτεύον κλειδί ονομάζεται id και αυξάνεται κάθε φορά που εισάγεται ένα νέο προϊόν προς πώληση, τα υπόλοιπα γνωρίσματα του είναι το όνομα του προϊόντος , η περιγραφή του ,η τιμή του προϊόντος η κατηγορία στην οποία ανήκει το συγκεκριμένο γνώρισμα είναι το πρωτεύον κλειδί του πίνακα κατηγορίες και το τελευταίο γνώρισμα είναι το χαρακτηριστικό του εκάστοτε προϊόντος.

products

Όνομα πεδίου	Τύπος δεδομένων
ID	Αυτόματη Αρίθμηση
product_name	Σύντομο κείμενο
product_description	Μεγάλο κείμενο
selling_price	Νομισματική μονάδα
trait	Σύντομο κείμενο
product_category_id	Αριθμός

```
create table products (
id serial,
product_name varchar(50) not null unique,
product_description text,
selling_price real default 0.0,
category int,
primary key(id),
foreign key (category) references categories(id)
);
alter table products add column trait varchar(20);
```

products

ID	product_name	product_de	selling_pric	trait	product_ca
1	intel i5 9500		200,00 €	1151	1
2	intel d3-s4510 480 gb		110,00 €	480GB	4
3	msi geforce rtx 2080 super		750,00 €	2080	2
4	msi mpg z390 gaming pro carbon ac		210,00 €	1151	3
5	intel i5 7500		100,00 €	1151	1
6	intel i5 8500		140,00 €	1151	1
7	intel i7 9900k		500,00 €	1151	1
8	intel i5 9600k		220,00 €	1151	1
9	amd ryzen 5 2600		112,00 €	am4	1
10	amd ryzen 5 3600x		205,00 €	am4	1
11	amd ryzen 5 3600		170,00 €	am4	1
12	amd ryzen 7 3700x		300,00 €	am4	1
13	amd ryzen 7 3800x		330,00 €	am4	1
14	asus rog stix z390-f		205,00 €	1151	3
15	asus rog strix x570-f gaming		300,00 €	am4	3
16	aorus z390 master		300,00 €	1151	3
17	aorus x570 pro		270,00 €	am4	3
18	msi mpg x570 gaming endge wifi		205,00 €	am4	3

Εικόνα 7 : οντότητα προϊόντα

Ο πίνακας κατηγορίες που αναφέρθηκε παραπάνω αποτελείται από το πρωτεύον του κλειδί και από το όνομα της κατηγορίας όπως για παράδειγμα κάρτα γραφικών ,επεξεργαστής ,τροφοδοτικό ,μνήμη ραμ κλπ.

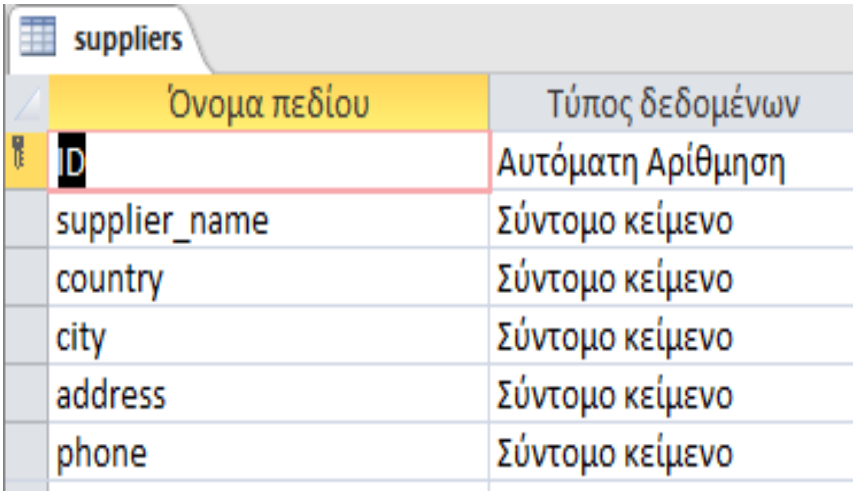
categories

Όνομα πεδίου	Τύπος δεδομένων
ID	Αυτόματη Αρίθμηση
category_name	Σύντομο κείμενο

```
create table categories (
id int,
category_name varchar(50)
);
alter table categories add primary key (id);
```

Εικόνα 8: οντότητα κατηγορίες προϊόντων

Επίσης ένας άλλος βασικός πίνακας είναι αυτός στον οποίο αποθηκεύονται πληροφορίες για τους προμηθευτές των προϊόντων. Το πρωτεύον κλειδί του πίνακα είναι ένας ακέραιος αριθμός ο οποίος αυξάνεται με κάθε νέα εγγραφή , ενώ τα υπόλοιπα γνωρίσματα του είναι το όνομα του προμηθευτή πχ intel η χωρά και η πόλη στην οποία εδρεύει , η διεύθυνση και το τηλέφωνο επικοινωνίας του κτλπ.



```
create table suppliers (
id serial ,
supplier_name varchar(50) unique,
country varchar(50),
city varchar(50),
address varchar(50),
phone varchar(50),
primary key (id)
);
```

suppliers

ID	supplier_na	country	city	address	phone
1	intel	usa	california	santa clara	32313223
2	msi	china	taiwan	new taipei city 231	657452412
3	AMD	USA	california	Santa Clara	123292810
4	NVIDIA	USA	california	2788 San Tomas Expy	83292910
5	GIGABYTE	China	New Taipei City 231	Taiwan	44544454545
6	Kingston	United Kingdo	Brooklands CloseSunbury-on-Thames	Middlesex TW16 7EP	83292810
7	Western Digit	USA	San Jose	5601 Great Oaks Parkv	8329710
8	Corsair	USA	California	Fremont	83292810
9	be quiet	Germany	Glinde	Biedenkamp 3A21509	836592810
10	ASUS	China	New Taipei City 231	Taiwan	832939990

Εικόνα 9: οντότητα προμηθευτές

Η επόμενη οντότητα είναι εκείνη των παραγγελιών των πελατών , όπου έχει ως πρωτεύον κλειδί έναν αύξοντα αριθμό, και γνωρίσματα το πρωτεύον κλειδί του πίνακα πελάτες, την ώρα και την ημερομηνία όπου έγινε η παραγγελία και το αν ο πελάτης θέλει να παραλάβει το προϊόν στο σπίτι του ή όχι.

orders

Όνομα πεδίου	Τύπος δεδομένων
order_id	Αυτόματη Αρίθμηση
customer_id	Αριθμός
time_of_the_order	Ημερομηνία/Ώρα
home_delivery	Ναι/Όχι

```
create table orders (
id serial ,
customer_id int,
time_of_the_order timestamp,
primary key(id),
foreign key (customer_id) references customers(id)
);
```

orders

order_id	customer_id	time_of_the	home_deliv
1	1	27/3/2020	☑
2	2	27/3/2020	☐
3	3	23/5/2020	☑
4	4	23/5/2020	☑
6	5	24/5/2020	☐
8	6	24/5/2020	☐
9	7	24/5/2020	☐
10	8	24/5/2020	☐
11	9	24/5/2020	☐
12	10	24/5/2020	☐
13	11	24/5/2020	☐
14	12	24/5/2020	☐
15	13	24/5/2020	☐
16	14	24/5/2020	☐
17	15	24/5/2020	☐
18	16	24/5/2020	☐
19	17	24/5/2020	☐
20	18	24/5/2020	☐
21	19	24/5/2020	☐
22	20	24/5/2020	☐

Εικόνα 10: οντότητα παραγγελίες

Επίσης δημιουργούμε έναν ενδιάμεσο πίνακα μεταξύ των παραγγελιών και των προϊόντων με όνομα order_products ο οποίος έχει ως πρωτεύον κλειδί το συνδυασμό των

πρωτευόντων κλειδιών του πίνακα παραγγελιών και προϊόντων και ως γνώρισμα την ποσότητα που θέλει ο κάθε πελάτης ανά προϊόν που έχει στην παραγγελία του.

order_products

Όνομα πεδίου	Τύπος δεδομένων
order_id	Αριθμός
product_id	Αριθμός
quantity	Αριθμός

```
create table order_products (
order_id int,
product_id int,
quantity int,
primary key (order_id,product_id),
foreign key (order_id) references orders(id),
foreign key (product_id) references products(id)
);
```

order_products

order_id	product_id	quantity
1	1	2
1	3	2
2	3	4
2	4	10
3	13	2
4	7	2
4	15	2
4	16	3
6	1	1
6	3	1
6	4	1
6	28	1
6	45	1
6	47	1
6	54	1
8	4	1
8	5	1
8	28	1
8	33	2
8	46	1

Εικόνα 11: πίνακας μεταξύ των οντοτήτων παραγγελίες και προϊόντα

Κάνουμε την ιδιά διαδικασία για την δημιουργία του πίνακα όπου το κατάστημα θα κάνει τις παραγγελίες του προς τους προμηθευτές. Ο πίνακας ονομάζεται refill και έχει ως πρωτεύον κλειδί μια αύξουσα τιμή ,ως ξένο κλειδί το πρωτεύον κλειδί του πίνακα των προμηθευτών και ως γνώρισμα την ώρα που έγινε η παραγγελία.

refill

Όνομα πεδίου	Τύπος δεδομένων
ID	Αυτόματη Αρίθμηση
supplier_id	Αριθμός
time_of_the_refill	Ημερομηνία/Ώρα

```
create table refill (
id serial,
supplier_id int,
time_of_the_refill timestamp,
primary key(id),
foreign key (supplier_id) references suppliers(id)
);
```

refill

ID	supplier_id	time_of_the
1	1	10/5/2020
2	2	10/5/2020
3	3	11/5/2020
4	5	11/5/2020
5	6	11/5/2020
6	7	11/5/2020
7	8	11/5/2020
8	9	11/5/2020
9	10	11/5/2020
10	1	25/5/2020

Εικόνα 12: οντότητα παραγγελιών του καταστήματος προς τους προμηθευτές

Η σύνδεση μεταξύ των παραγγελιών του καταστήματος από τους προμηθευτές με τον πίνακα των αποθεμάτων (προϊόντων) επιταχύνεται μέσο ενός ενδιάμεσου πίνακα με όνομα refill_product όπου έχει ως πρωτεύον κλειδί τον συνδυασμό των πρωτευόντων κλειδιών του πίνακα προϊόντα και των παραγγελιών και έχει ως γνώρισμα την ποσότητα της παραγγελίας και την τιμή όπου τα προϊόντα αγοραστήκαν.

refill_products

Όνομα πεδίου	Τύπος δεδομένων
refill_id	Αριθμός
product_id	Αριθμός
quantity	Αριθμός
purchase_price	Νομισματική μονάδα

```
create table refill_products (
refill_id int,
product_id int,
quantity int,
purchase_price real default 0.0,
primary key (refill_id,product_id),
foreign key (refill_id) references refill(id),
foreign key (product_id) references products(id)
);
```

refill_products

refill_id	product_id	quantity	purchase_pric
1	1	100	100,00 €
1	2	40	75,00 €
1	5	40	50,00 €
1	6	80	70,00 €
1	7	150	300,00 €
1	8	70	100,00 €
1	40	20	60,00 €
2	3	30	400,00 €
2	4	45	100,00 €
2	18	30	100,00 €
2	20	20	350,00 €
2	24	80	200,00 €
2	51	5	40,00 €
3	9	30	50,00 €
3	10	110	100,00 €
3	11	40	90,00 €
3	12	120	150,00 €
3	13	200	160,00 €
4	16	40	170,00 €

Εικόνα 13: πίνακας μεταξύ της οντότητας παραγγελίες προς τους προμηθευτές και της οντότητας προϊόντα

Και η τελευταία οντότητα είναι εκείνη των χρεωστικών καρτών των πελατών ο οποίος έχει αύξον πρωτεύον κλειδί , και τα γνωρίσματα του είναι τα 16 ψηφιά της κάρτας, το όνομα του κάτοχου της ,η ημερομηνία λήξης και το cvv.

credit_cards

Όνομα πεδίου	Τύπος δεδομένων
ID	Αυτόματη Αρίθμηση
customer_id	Αριθμός
number16	Σύντομο κείμενο
cardholder_name	Σύντομο κείμενο
expire_date	Ημερομηνία/Ώρα
cvv	Σύντομο κείμενο

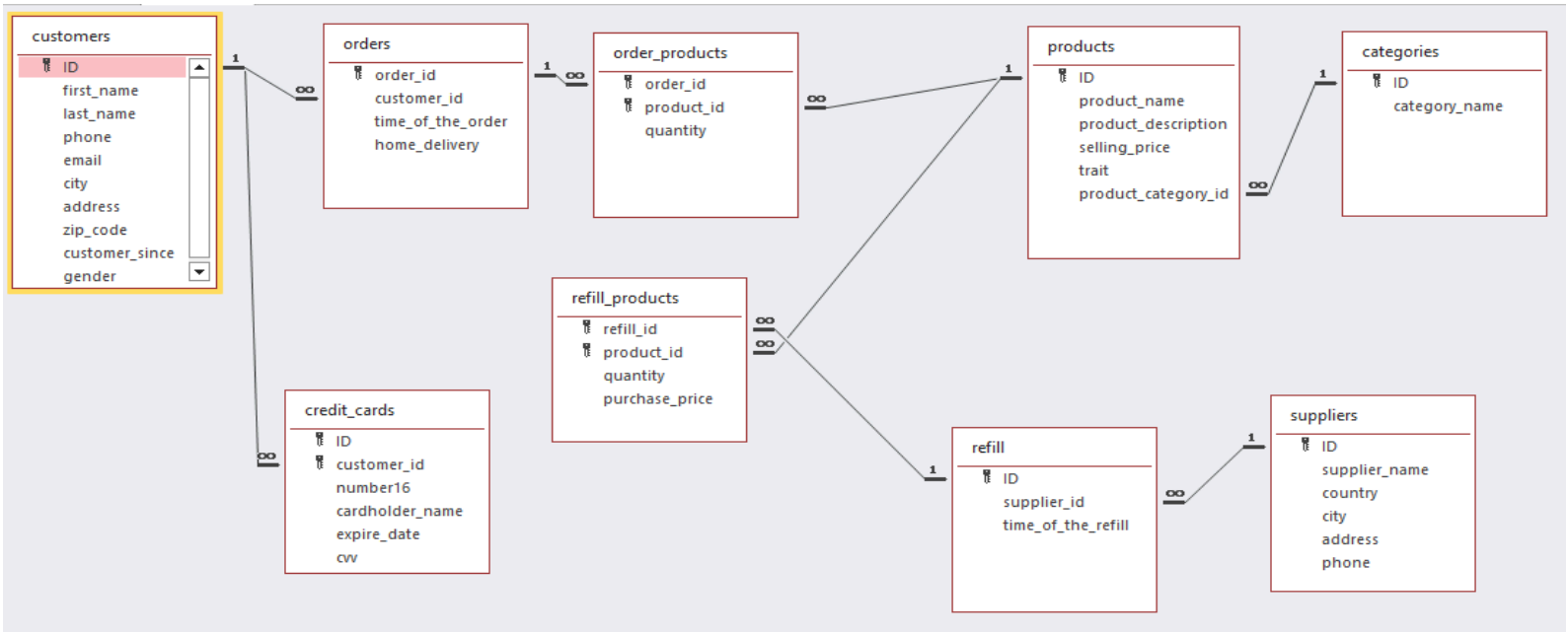
```
create table creditcards (
id  serial,
numbers16 varchar(19),
cardholder_name varchar(50),
expire_date date,
cvv varchar(3),
primary key (id)
);
```

credit_cards

ID	customer_i	number16	cardholder_	expire_date	cvv
1	1	11111111111	kkkkkkkkkkkk	05/25	333
2	1	22222222222	ooooooooooo	06/25	999
3	1	33333333333	lllllllllllllllllllllll	08/25	777
4	2	77777777777	mmmmmmm	09/25	444
5	2	55555555555	ppppppppppp	09/25	666
6	3	11111111111	kkkkkkkkkkkk	05/25	777
7	4	54454545454	ppppppppppp	08/25	777

Εικόνα 14: οντότητα πιστωτικές κάρτες

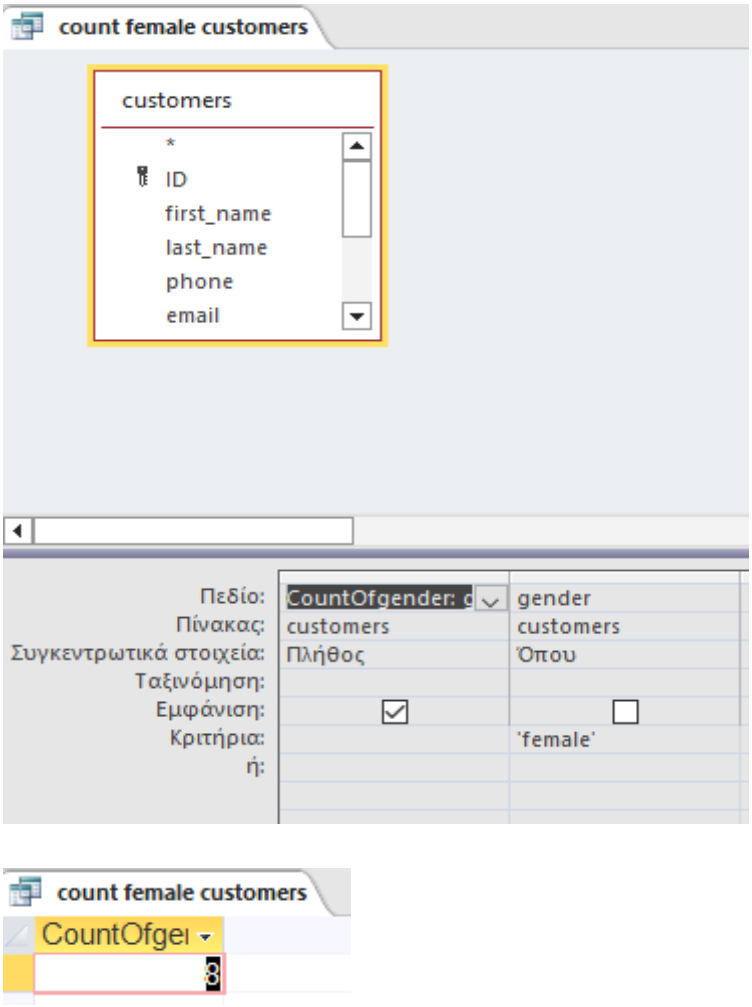
Έχοντας δημιουργήσει της οντότητες με τα γνωρίσματα τους, και εισάγει μερικά δεδομένα μπορούμε να δούμε το διάγραμμα οντοτήτων συσχετίσεων.



Εικόνα 15: διάγραμμα οντοτήτων συσχετίσεων

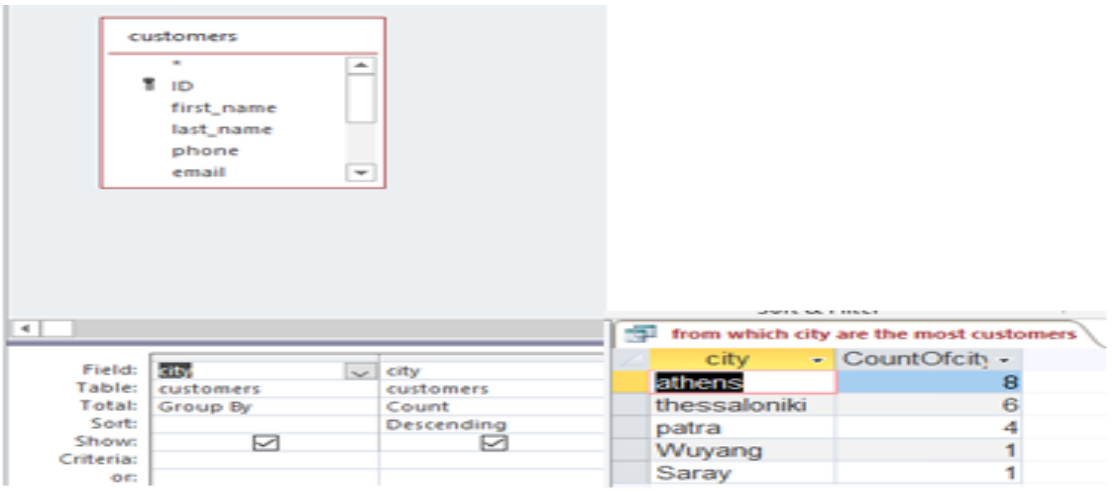
Πιο αναλυτικά η σχέση μεταξύ πελτέδων και παραγγελιών είναι πολλά προς πολλά δηλαδή πόλοι πελάτες μπορούν να πραγματοποιήσουν πολλές παραγγελίες και αυτό ισχύει και αντίστροφα. Επίσης σε πολλές παραγγελίες μπορούν να υπάρξουν πολλά διαφορετικά προϊόντα οπότε η σχέση μεταξύ προϊόντων και παραγγελιών είναι επίσης πολλά προς πολλά. Η σχέση μεταξύ πελατών και τραπεζικών καρτών είναι ένα προς πολλά δηλαδή ένας πελάτης μπορεί να έχει πολλές κάρτες πληρωμής και από την άλλη πολλές κάρτες μπορούν να ανήκουν σε έναν μόνο πελάτη. Επίσης μεταξύ των προϊόντων και των κατηγοριών τους υπάρχει σχέση πολλά προς ένα δηλαδή πολλά προϊόντα ανήκουν μόνο σε μια κατηγορία και το ίδιο ισχύει αντίστροφα δηλαδή μια κατηγορία μπορεί να ανήκει σε πολλά προϊόντα ταυτόχρονα. Από την άλλη πλευρά της προμήθειας των προϊόντων υπάρχει σχέση πολλά προς πολλά μεταξύ των προμηθευτών και των προϊόντων που αποστέλλουν (πουλάνε) στο κατάστημα , δηλαδή πολλοί προμηθευτές προμηθεύουν το κατάστημα με πολλά προϊόντα , αντίστοιχα πολλά διαφορετικά προϊόντα μπορούν να πωληθούν στην επιχείρηση από πολλούς διαφορετικούς προμηθευτές για παράδειγμα ο supplier amd πουλάει στην εταιρεία και κάρτες γραφικών και επεξεργαστές.

Έχοντας ολοκληρώσει τα παραπάνω συνεχίζουμε με την δημιουργία απλών ερωτημάτων για την ανάλυση των δεδομένων. Ένα από αυτά είναι για παράδειγμα η εύρεση του συνόλου των γυναικών που αγοράσαν από το κατάστημα.



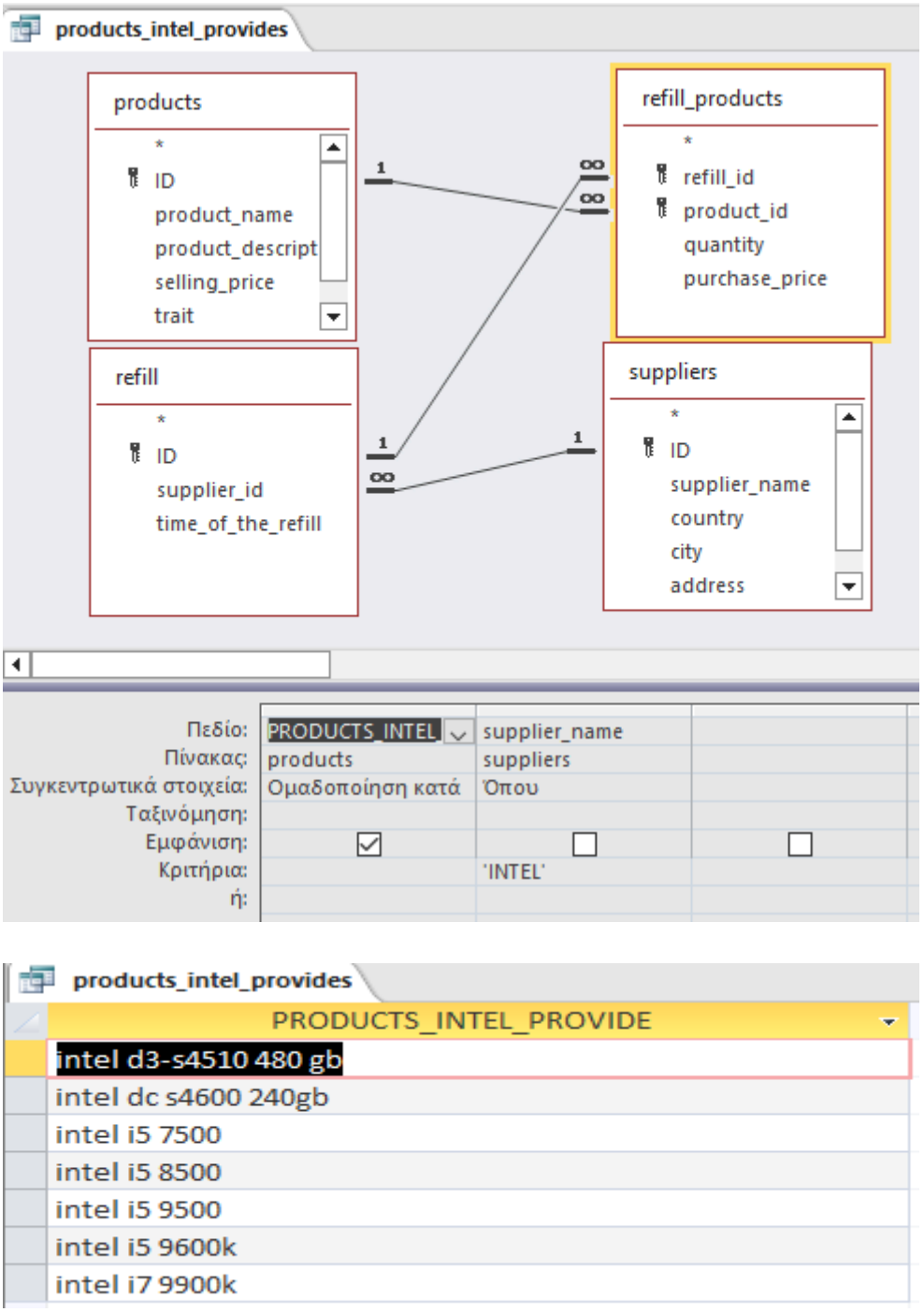
Εικόνα 16: ερώτημα για την εύρεση του αριθμού των γυναικών πελατών

Επίσης μπορεί να συνταχθεί ένα ερώτημα για την εύρεση της πόλης από τη οποία έγιναν οι περισσότερες παραγγελίες.



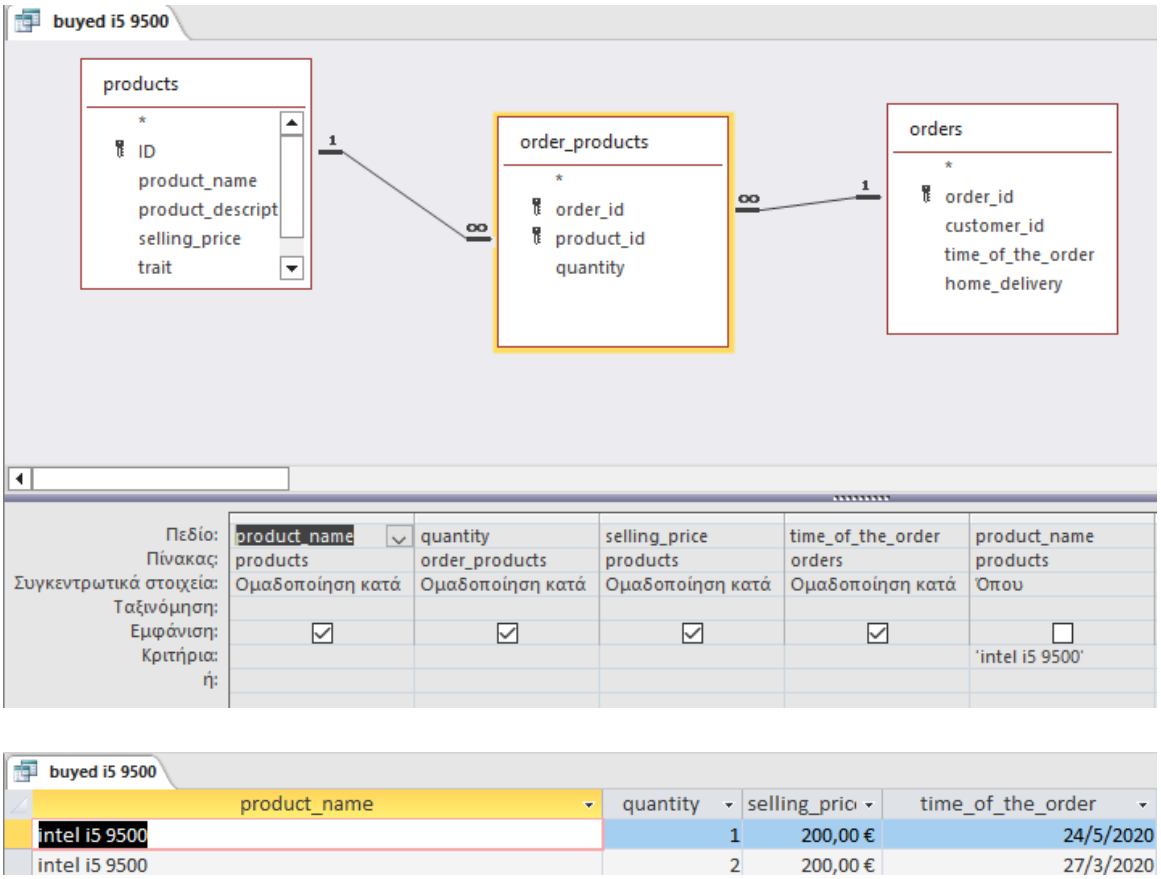
Εικόνα 17 : ερώτημα για την εύρεση αριθμού παραγγελιών ανά πόλη

Αντίστοιχα ερωτήματα μπορούν να γίνουν και για τα προϊόντα και τους προμηθευτές. Για παράδειγμα ένα ερώτημα θα μπορούσε να είναι ποια προϊόντα προμηθεύει στο κατάστημα η εταιρεία intel.



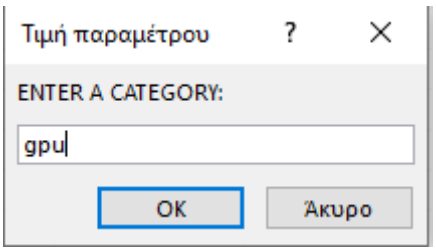
Εικόνα 19: ερώτημα για την εύρεση των προϊόντων της intel

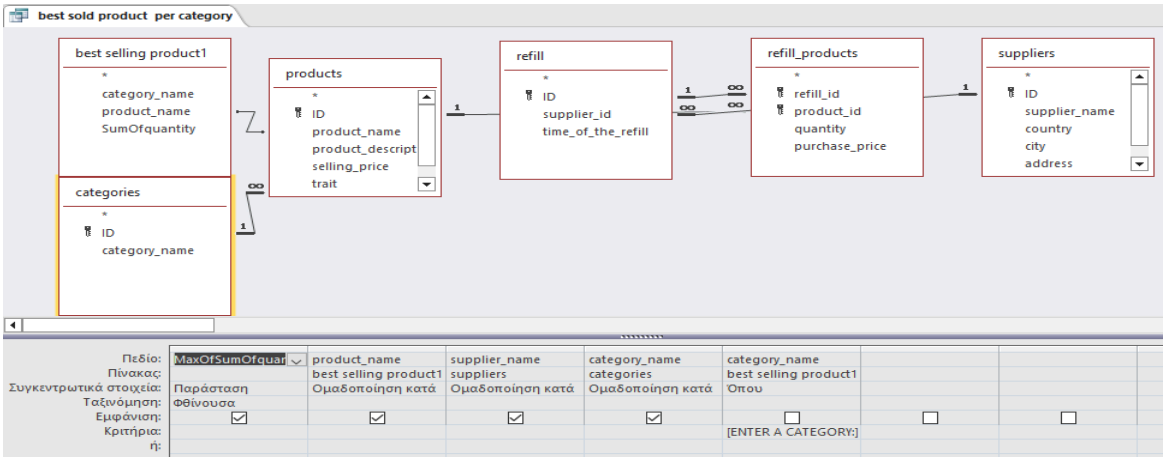
Επιπλέον μπορούν να βρεθούν οι πωλήσεις ανά προϊόν για παράδειγμα θα δημιουργήσουμε ένα ερώτημα για τις πωλήσεις του επεξεργαστή intel I5 9500.



Εικόνα 20: εύρεση πωλήσεων του επεξεργαστή intel i5 9500

Επίσης μπορεί να δημιουργηθεί ερώτημα για την εύρεση του προϊόντος ανά κατηγορία με τις περισσότερες πωλήσεις. Θα δοκιμάσουμε το παραπάνω ερώτημα εισάγοντας ως κατηγορία τις κάρτες γραφικών.



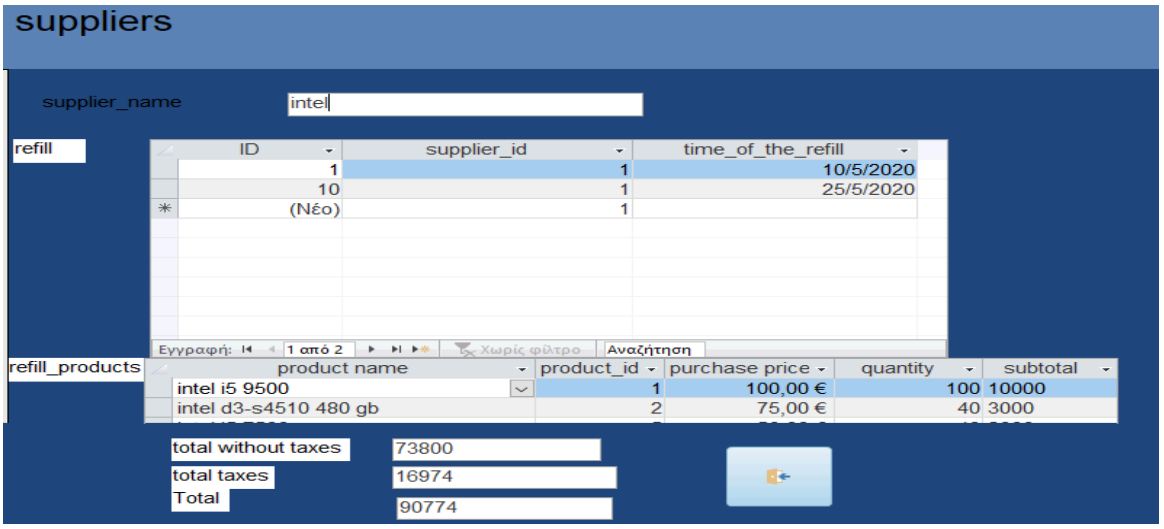


best sold product per category

MaxOfSumO	product_name	supplier_name	category_
9	msi geforce rtx 2080 super	msi	GPU
5	aorus geforce rtx 2070 super 8gb	GIGABYTE	GPU
4	asus radeon rx 5700 xt 8gb rog stix oc	ASUS	GPU
3	msi radeon rx 5700 xt 8gb evoke oc	msi	GPU
3	asus geforce rtx 2080 super 8gb rog strix advanced	ASUS	GPU
3	asus geforce rtx 2060 super 8gb evo advanced	ASUS	GPU
2	msi geforce rtx 2070 super 8gb gaming x	msi	GPU

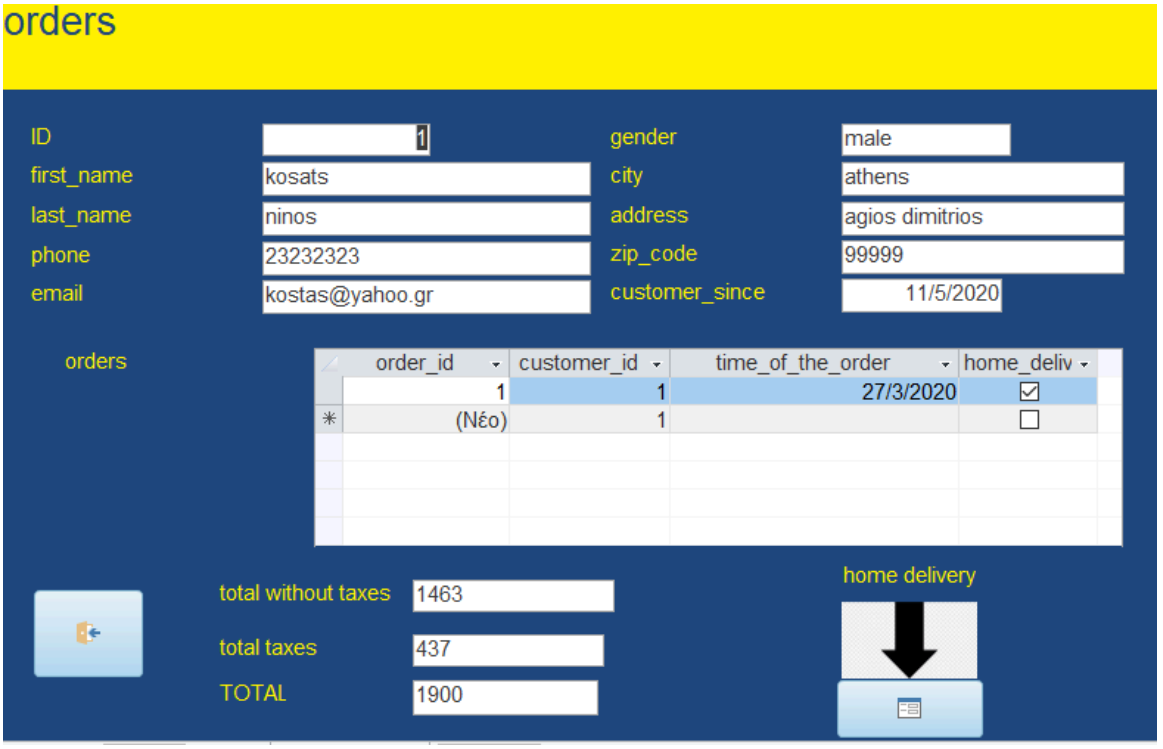
Εικόνα 21: ερώτημα για την εύρεση της κάρτας γραφικών με τις περισσότερες πωλήσεις

Έχοντας υλοποιήσει μερικά ερωτήματα συνεχίζουμε την ανάπτυξη της βάσης δεδομένων με την δημιουργία φορμών. Η πρώτη φόρμα είναι εκείνη η οποία χρησιμοποίει ο υπεύθυνος προμήθειων για να κάνει τις παραγγελίες των προϊόντων από τους προμηθευτές. Σε αυτήν την φόρμα ο υπεύθυνος επιλεγεί την ποσότητα του προϊόντος που θέλει να παραγγείλει και ταυτόχρονα βλέπει την τιμή πριν και μετα τους φόρους.



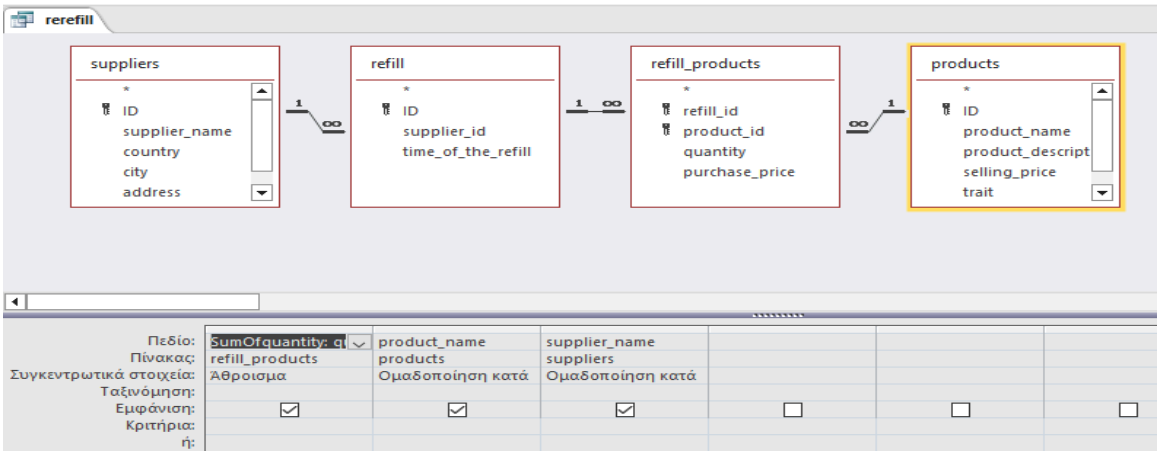
Εικόνα 22: φόρμα για τις παραγγελίες του καταστήματος προς τους προμηθευτές

Με τον ίδιο τρόπο οι πελάτες μπορούν να κάνουν τις παραγγελίες τους μέσα από το κατάστημα δηλαδή επιλέγουν το προϊόν και την ποσότητα στην οποία το θέλουν και στο τέλος υπολογίζεται η συνολική τιμή της παραγγελίες πριν και μετα τους φόρους.



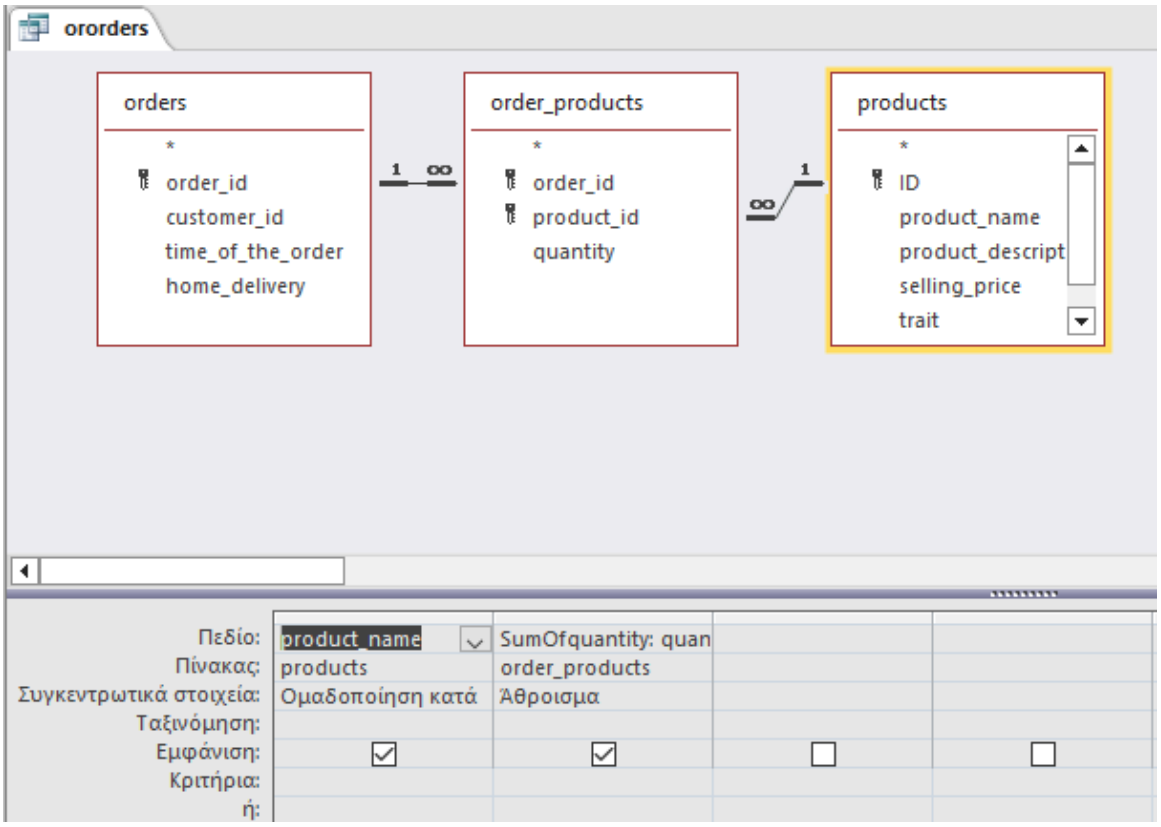
Εικόνα 23: φόρμα για τις παραγγελίες των πελατών προς το κατάστημα

Αφού έχουμε δημιουργήσει τις βασικές φόρμες και έχοντας κάνει τις πρώτες παραγγελίες από τους προμηθευτές και οι πελάτες έχουν κάνει τις πρώτες τους αγορές, πρέπει να υπολογίσουμε το υπολειπόμενο απόθεμα ανά προϊόν. Για να γίνει το παραπάνω πρώτα πρέπει να γίνει έλεγχος των αγορών του καταστήματος από τους προμηθευτές.



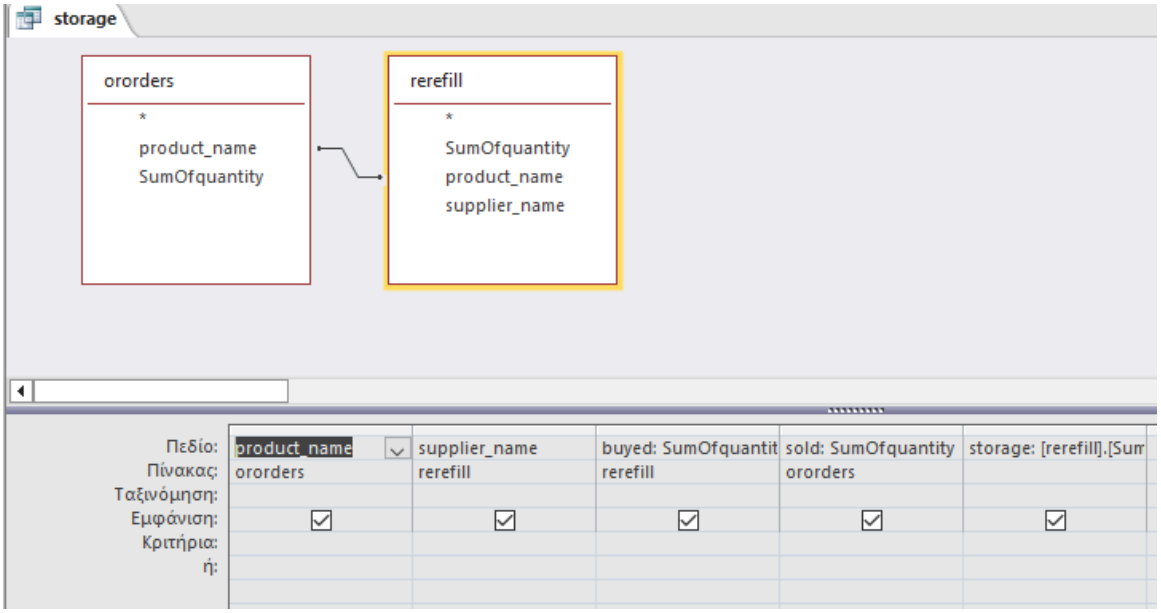
Εικόνα 24: ερώτημα για το απόθεμα μέρος 1

Εν συνέχεια πρέπει να ελέγξουμε ποσά προϊόντα έχει πουλήσει το κατάστημα στους πελάτες του.



Εικόνα 25: ερώτημα για το απόθεμα μέρος 2

Τέλος ενώνουμε αυτά τα δυο ερωτήματα και περνούμε ως αποτέλεσμα το διαθέσιμο απόθεμα του καταστήματος ανά προϊόν.

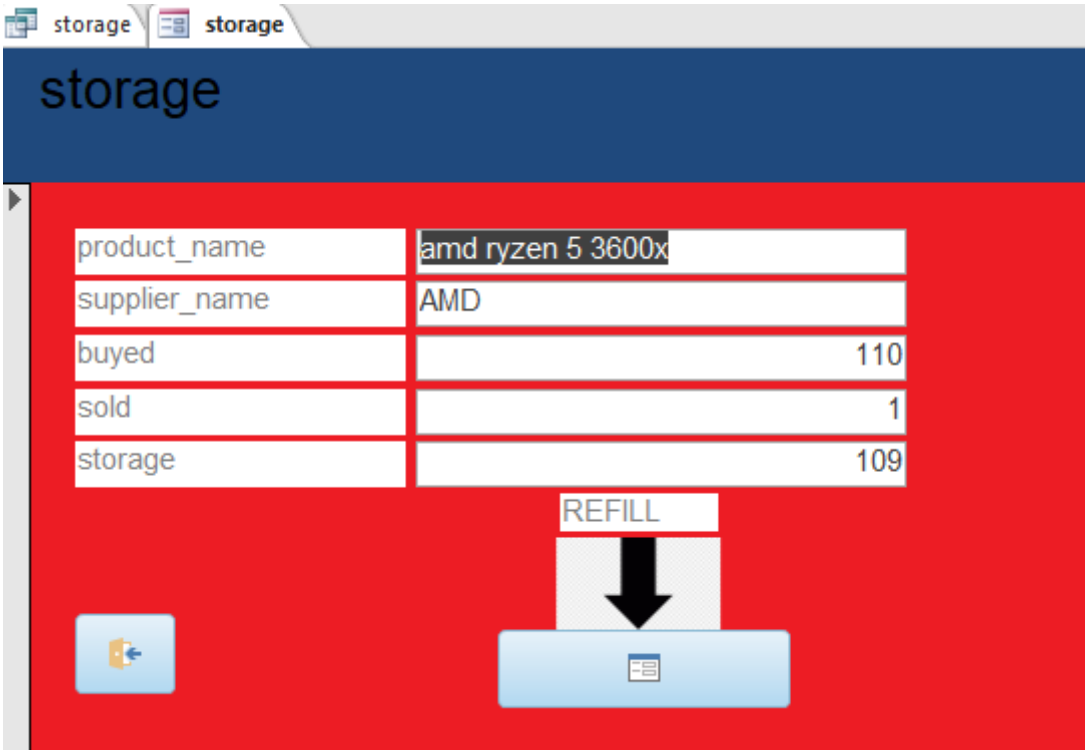


storage

product_name	supplier_na	buyed	sold	storage
amd ryzen 5 3600x	AMD	110	1	109
amd ryzen 7 3700x	AMD	120	2	118
amd ryzen 7 3800x	AMD	200	10	190
aorus geforce rtx 2070 super 8gb	GIGABYTE	12	5	7
aorus p 750w (gold)	GIGABYTE	5	2	3
aorus x570 pro	GIGABYTE	100	1	99
aorus z390 master	GIGABYTE	40	5	35
asus geforce rtx 2060 super 8gb evo advanced	ASUS	40	3	37
asus geforce rtx 2080 super 8gb rog strix advanced	ASUS	30	3	27
asus radeon rx 5700 xt 8gb rog stix oc	ASUS	20	4	16
asus rog stix z390-f	ASUS	100	4	96
ASUS rog strix 750w (gold)	ASUS	20	2	18
asus rog strix x570-f gaming	ASUS	70	9	61
be quiet base 500 black	be quiet	20	1	19

Εικόνα 26: ερώτημα για το απόθεμα μέρος 3

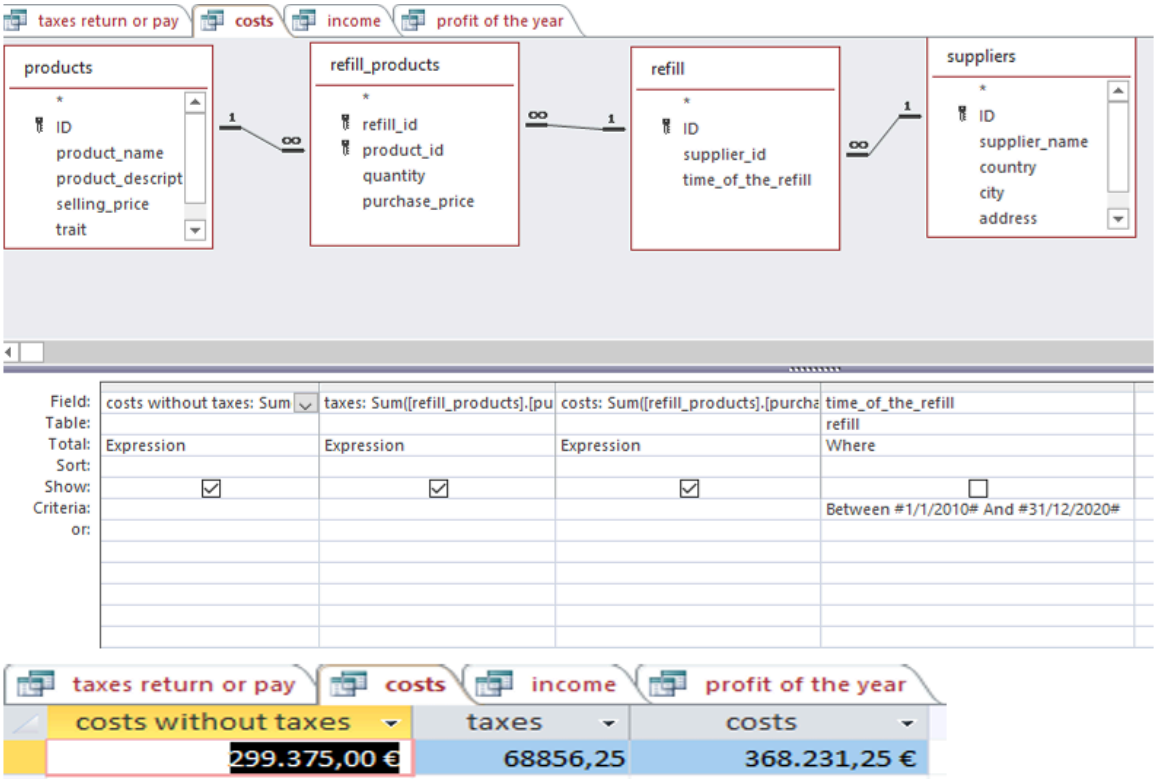
Για τα παραπάνω μπορεί να δημιουργηθεί και η αντίστοιχη φόρμα όπου ο υπεύθυνος προμήθειων μπορεί να δει το διαθέσιμο απόθεμα ανά προϊόν και αν θεωρεί ότι πρέπει να γίνει νέα παραγγελία στον προμηθευτή μπορεί να πατήσει το κουμπί και να μεταφερθεί άμεσα στην φόρμα refill.



Εικόνα 27: φόρμα αποθέματος

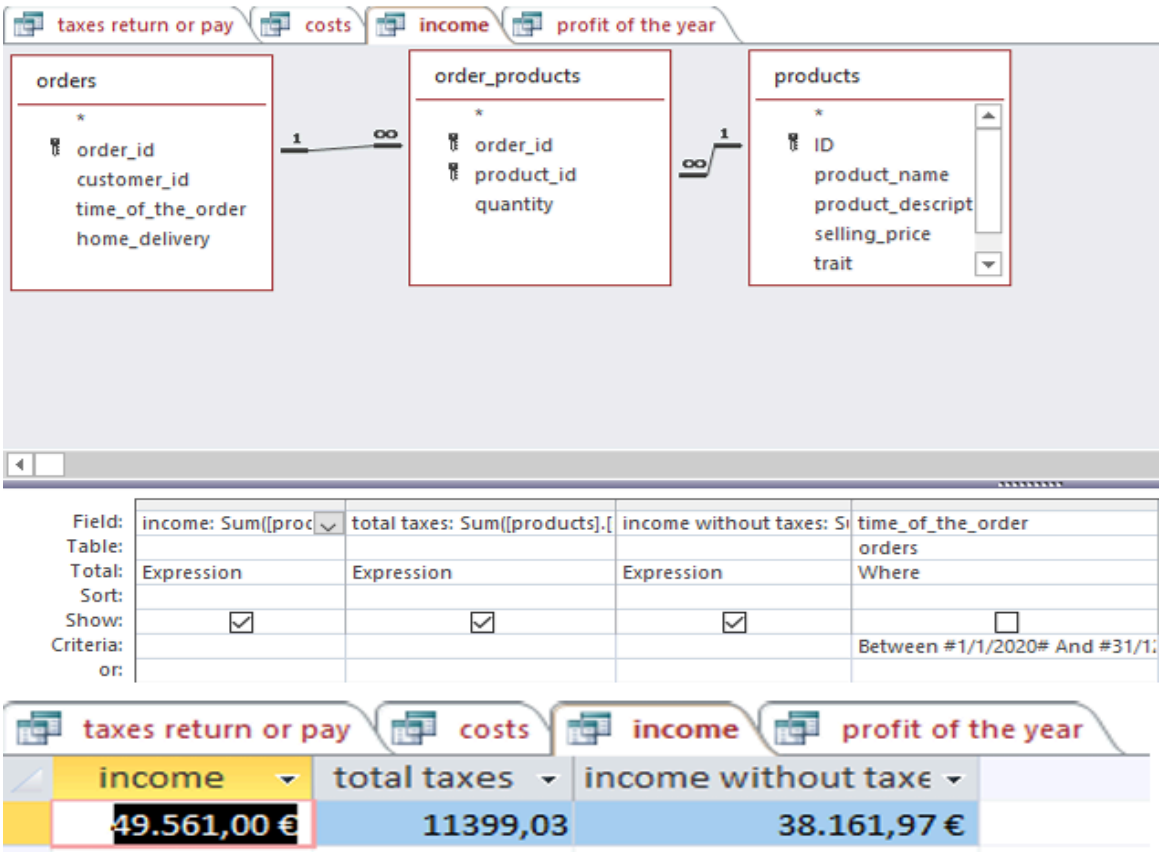
Ο βασικός στόχος μιας εταιρείας είναι η δημιουργία εσοδών και κατά επέκταση κερδών, σε αυτό το σημείο της ανάλυσης θα δημιουργήσουμε εκείνα τα ερωτήματα ώστε να απαντήσουμε στο παραπάνω ερώτημα ποσά ήταν τα κέρδη ή οι ζημίες του καταστήματος.

Αρχικά υπολογίζουμε το κόστος των αγορών από τους προμηθευτές sum (purchase_price* quantity) και προσθέτουμε τους φόρους (ΦΠΑ εισροών).



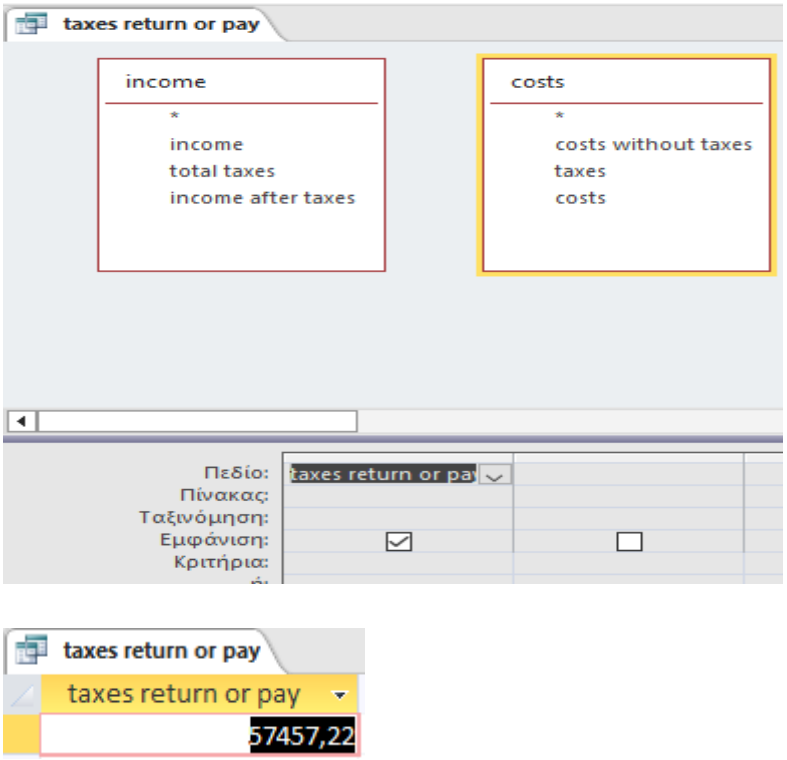
Εικόνα 28: ερώτημα για τα κέρδη ή τις ζημίες μέρος 1

Έχοντας ολοκληρώσει τον υπολογισμό των εξόδων , υπολογίζουμε τα έσοδα από τις πωλήσεις αφαιρώντας τους φόρους εκρών.



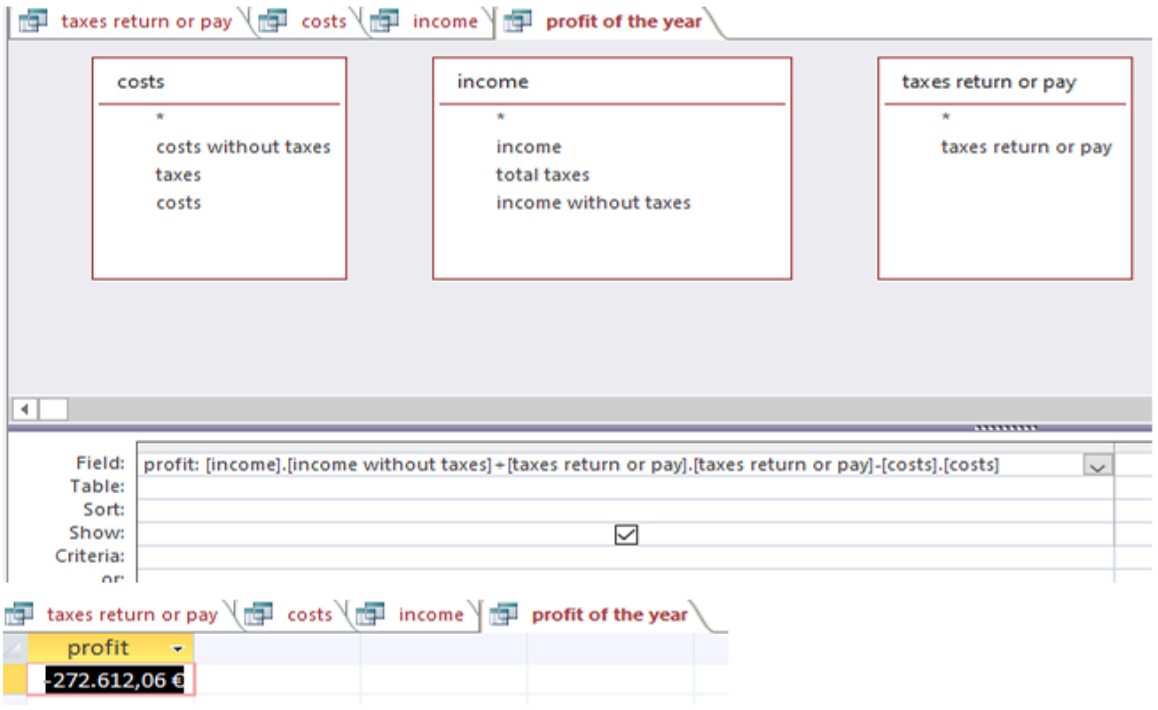
Εικόνα 29: ερώτημα για τα κέρδη ή τις ζημίες μέρος 2

Στην συνέχεια πρέπει να ελέγξουμε αν τελικά η εταιρεία πρέπει να πληρώσει φόρους ή να της γίνει επιστροφή , για τα συγκεκριμένα τυχαία δεδομένα και μέχρι εκείνη την χρονική περίοδο, το ΦΠΑ εισροών είναι μεγαλύτερο από το ΦΠΑ εκροών οπότε η επιχείρηση θα λάβει επιστροφή φόρου.



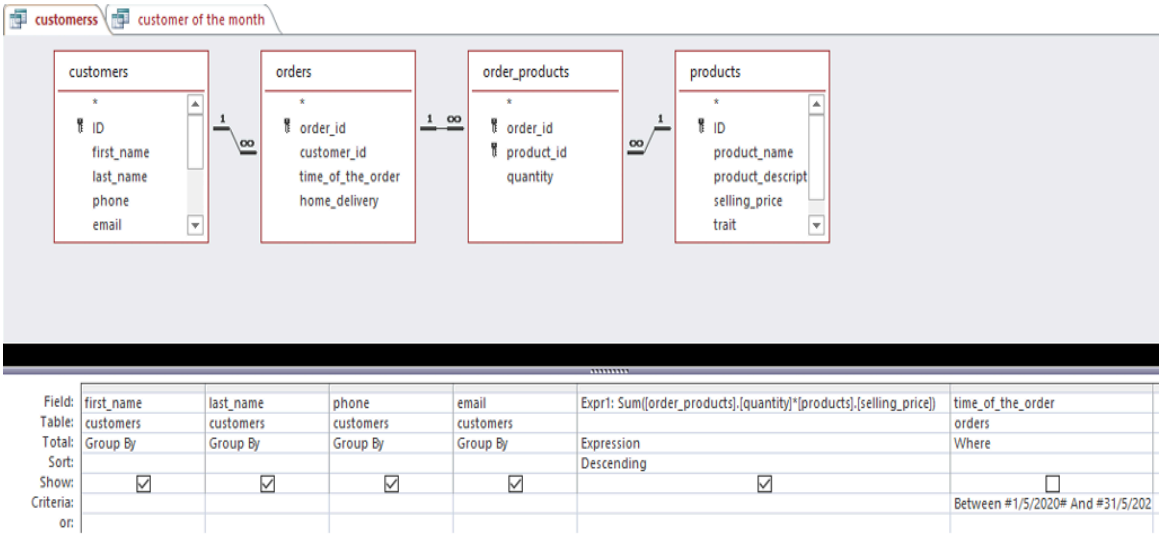
Εικόνα 30: ερώτημα για τα κέρδη ή τις ζημίες μέρος 3

Έχοντας συλλέξει όλα τα παραπάνω καταλήγουμε στο τελικό αποτέλεσμα δηλαδή στον υπολογισμό των ζημίων.



Εικόνα 31: ερώτημα για τα κέρδη ή τις ζημίες μέρος 4

Τέλος όπως είδαμε και στην προηγουμένη ενότητα οι βάσεις δεδομένων και πιο συγκεκριμένα τα crm για την διαχείριση των πελατών και των αναγκών τους βοηθούν στον υπολογισμό διαφόρων στατιστικών. Στην συγκεκριμένη περίπτωση θα βρούμε μέσο της βάσης δεδομένων εκείνο τον πελάτη που έκανε τις περισσότερες σε αξία παραγγελίες (δηλαδή <<ξόδεψε>> τα περισσότερα χρήματα στο κατάστημα) με στόχο την παροχή μεγάλης έκπτωσης στην επόμενη αγορά του. Για να γίνει το παραπάνω δημιουργούμε το ερώτημα



customerss | customer of the month

first_name	last_name	phone	email	Expr1
vaso	kipriou	2107463745	vasokip@hotmail.com	7.652,00 €
sofia	pissa	210887669	sofip@gmail.com	5.051,00 €
giorgos	disketas	6932331223	gdisc@hotmail.com	3.870,00 €
anna	vixou	23121434	annavix@hotmail.com	2.650,00 €
Barb	Comoletti	607-187-8242	bcomoletti3@upenn.e	2.500,00 €
nikos	papanastasiou	6937371212	nikpap@hotmail.com	2.430,00 €
nikoleta	papaueodorou	6923344511	nickpapade8eodorou@	2.355,00 €
mixalis	poupakis	21094394934	mixpoup@gmail.com	2.300,00 €
maria	nikolopoulou	6943824832	marianick@gmail.com	2.269,00 €
ilias	papandreou	2104546567	ilias2000@hotmail.cor	1.595,00 €
stelios	papanikolaou	2104565657	steliospap@yahoo.gr	1.560,00 €
bill	papas	2109318389	pap@yahoo.gr	1.465,00 €
eleni	xristou	21047897664	helenxr@yahoo.gr	1.443,00 €
xara	argirou	213323345	xaraar@hotmail.com	1.425,00 €
kostas	nikolaou	692343432	k.nick@hotmail.com	1.377,00 €
nikoleta	aggelopoulou	6934737423	aggnick@hotmail.com	1.299,00 €
alex	kapas	2109344211	kap@yahoo.gr	660,00 €
Michal	Jereatt	894-697-2618	mjereatt2@utexas.edu	660,00 €

Εικόνα 32: ερώτημα για την εύρεση του πελάτη με την μεγαλύτερη αξία αγορών μέσα στον μηνά

Ολοκληρώνοντας την ενότητα μπορούμε να αντιληφθούμε ποσό χρήσιμες είναι οι βάσεις δεδομένων ακόμα και για πολύ μικρά καταστήματα. Προφανώς ακόμα και για εκείνα εκδικιέται η αγορά μιας βάσης δεδομένων από μια τρίτη επιχείρηση όπως πχ η softone ή η epsilonet και όχι η κατασκευή της από την ιδιά την επιχείρηση καθώς με την πρώτη επιλογή επιτυγχάνεται η συνεχή συντήρηση και εξυπηρέτηση της εταιρίας σε τυχόν προβλήματα κατά την χρήση της. Βέβαια το κόστος της πρώτης επιλογής είναι αρκετές φορές υψηλότερο από την κατασκευή μιας βάσης δεδομένων in house(από την ιδιά την επιχείρηση).

2. ΕΙΣΑΓΩΓΗ ΣΤΗΝ ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ

2.1 AI

Η Τεχνητή Νοημοσύνη αναφέρεται στην ανάπτυξη αλγορίθμων που δίνουν τη δυνατότητα σε ένα υπολογιστικό σύστημα να εκτελεί εργασίες που μέχρι πρόσφατα απαιτούσαν ανθρώπινη νοημοσύνη. Οι εφαρμογές της είναι πλέον ευρέως διαδεδομένες και περιλαμβάνουν λειτουργίες όπως η αναγνώριση φωνής, που χρησιμοποιείται σε αυτόματα τηλεφωνικά κέντρα, αλλά και η ανάλυση εικόνων, όπως γίνεται στα αυτόνομα οχήματα της Tesla, τα οποία αναγνωρίζουν σήματα κυκλοφορίας όπως τα όρια ταχύτητας. Η τεχνητή νοημοσύνη μπορεί να ταξινομηθεί σε τρεις βασικές κατηγορίες.Η πρωτη κατηγορια είναι το narrow ai οπού οι αλγόριθμοι σε αυτά έχουν σχεδιαστεί για να εκτελούν συγκεκριμένες εργασίες με υψηλή ακρίβεια, αλλά δεν διαθέτουν γενική αντίληψη ή κατανόηση. Μερικά παραδείγματα είναι ένας αλγόριθμο που παίζει μόνο σκάκι, μοντέλα που απαντούν σε τυποποιημένες ερωτήσεις πελατών. Συστήματα που ταξινομούν εικόνες με βάση το αν περιέχουν γάτες ή σκύλους. Αν και είναι πολύ χρήσιμοι, οι αλγόριθμοι αυτοί λειτουργούν αποκλειστικά στο πλαίσιο για το οποίο εκπαιδεύτηκαν και δεν μπορούν να εφαρμοστούν σε άλλες εργασίες.

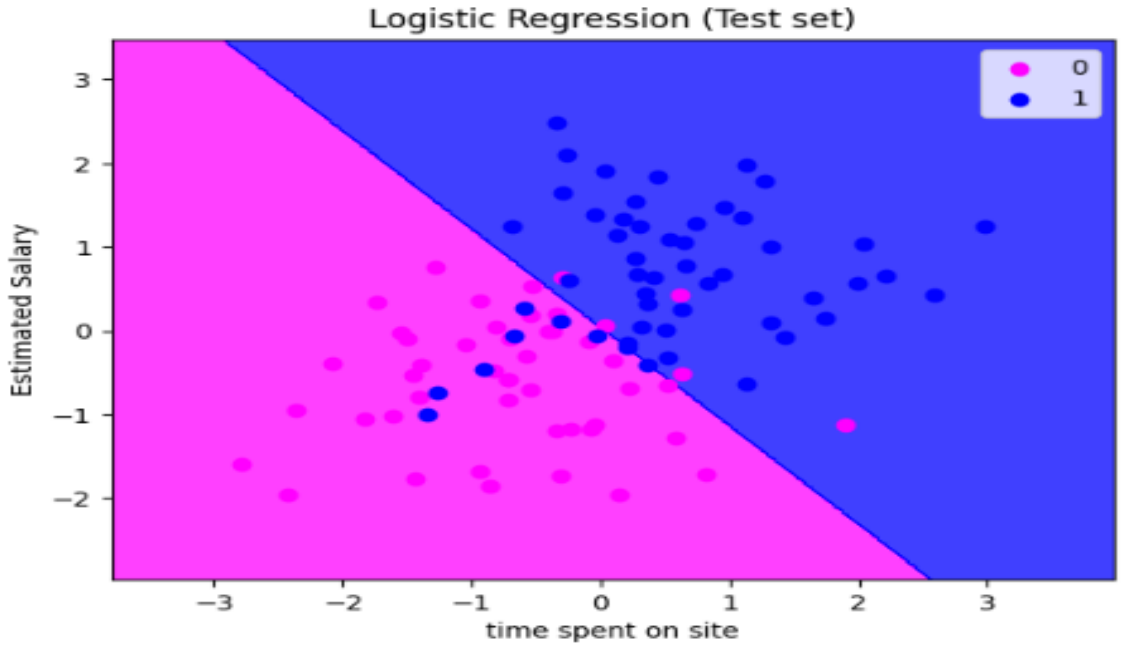
Στην δεύτερη κατηγορία ανήκουν οι αλγόριθμοι general ai, σε αυτό το επίπεδο οι αλγόριθμοι διαθέτουν την ικανότητα να αντιλαμβάνονται, να μαθαίνουν και να επιλύουν προβλήματα σε πολλούς τομείς, όπως ένας άνθρωπος. Θα μπορούσαν δηλαδή να εκτελούν πολλαπλές πνευματικές εργασίες χωρίς την ανάγκη ξεχωριστής εκπαίδευσης για κάθε μία. Αν και πρόκειται για έναν θεωρητικό στόχο της επιστήμης της AI, δεν έχει ακόμα υλοποιηθεί με επιτυχία.

Superintelligent AI

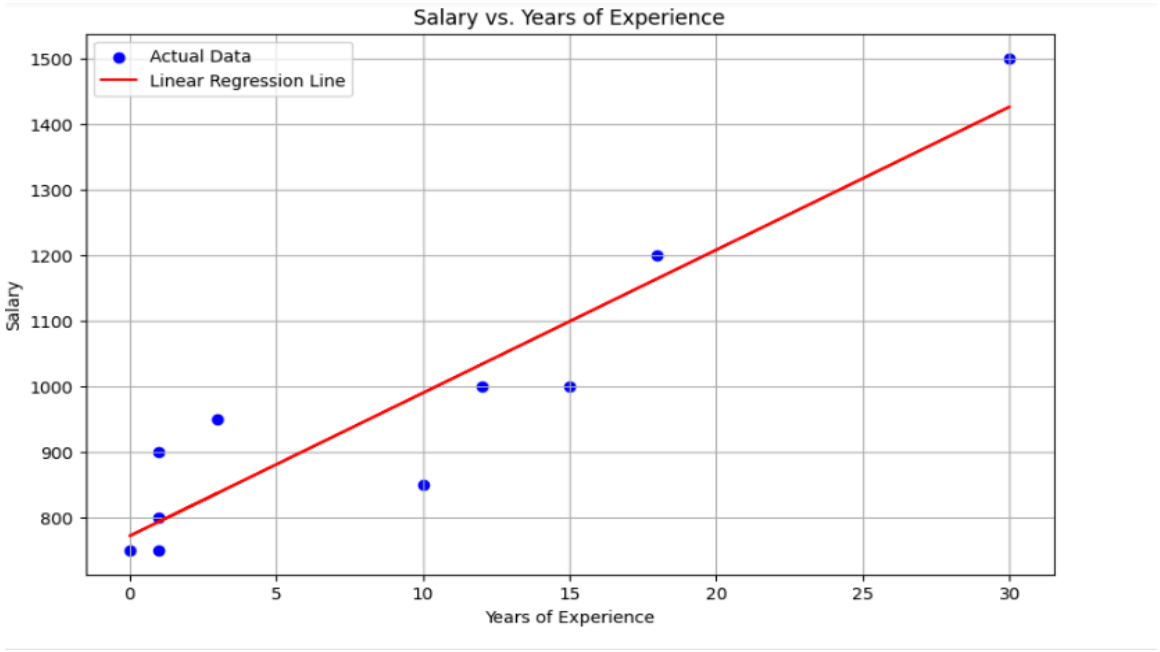
Και τέλος στην τρίτη κατηγορία ανήκουν οι αλγόριθμοι superintelligent ai οπού είναι η πιο προχωρημένη και θεωρητική μορφή τεχνητής νοημοσύνης, στην οποία τα συστήματα ξεπερνούν τις διανοητικές δυνατότητες του ανθρώπου. Τα συστήματα αυτά θα μπορούσαν να αναλύουν, να αποφασίζουν και να καινοτομούν πιο αποτελεσματικά από οποιονδήποτε άνθρωπο. Αυτή η μορφή παραμένει προς το παρόν υποθετική και αποτελεί αντικείμενο εντατικής έρευνας και φιλοσοφικού προβληματισμού.

Τι όμως είναι μηχανική μάθησή; Υπάρχουν διάφοροι ορισμοί για την μηχανική μάθηση όπως για παράδειγμα αυτή του Arthur Samuel οπού αναφέρει ότι μηχανική μάθηση είναι

το πεδίο μελέτης που δίνει στους υπολογιστές την ικανότητα να μαθαίνουνε, χωρίς να έχουν ρητά προγραμματιστεί ή για παράδειγμά ο ορισμούς tom M Mitchell όπου αναφερει ότι μηχανική μάθησή είναι ένα πρόγραμμα υπολογιστή όπου μαθαίνει από εμπειρία E ως προς μια κλάση εργασιών T και ένα μέτρο επίδοσης P, αν η επίδοσή του σε εργασίες της κλάσης T, όπως αποτιμάται από το μέτρο P, βελτιώνεται με την εμπειρία E. Η ένας άλλος πιο διαισθητικός ορισμούς για την μηχανικοί μάθησή είναι ότι μηχανική μάθησή είναι η επίλυση γεωμετρικών προβλημάτων για παράδειγμα η παλινδρόμηση ένα από τα πιο γνωστά μοντέλα μηχανικής μάθησης έχει ως στόχο να βρεθεί εκείνη η γραμμή η οποία περνάει πλησιέστερα και ταυτόχρονος από όλες τις παρατηρήσεις X,Y ή για παράδειγμα στα μοντέλα ταξινόμησης πρέπει να βρεθεί εκείνη η γραμμή οπου διαχωρίζει όσο το δυνατόν καλυτέρα τα δεδομένα. Δηλαδή οι αλγόριθμοι μηχανικής μάθησης έχουν ως στόχο να μοντελοποιησουν όσο το δυνατό καλυτέρα τα δεδομένα(να αποκτήσουν εμπειρία) με στόχο να κάνουν προβλέψεις είτε για το πως θα κινηθούν τα επόμενα δεδομένα αν πρόκειται για ανάλυση χρονοσειρων είτε να ταξινομήσουν διάφορες καταστάσεις ανάλογα με τα χαρακτηριστικά τους.



Εικόνα 33: ταξινόμηση



Εικόνα 34: παλινδρόμηση

Σε αυτό το σημείο θα αναλύσουμε τις κατηγορίες των αλγορίθμων μηχανικής μάθησης οι οποίες είναι τρεις η επιβλεπομένη μάθηση(supervised learning), η μη επιβλεπομένη μάθηση(unsupervised learning) και η ενισχυμένη μάθηση (reinforcement learning). Στην επιβλεπομένη μάθηση τα δεδομένα αποτελούνται από διάφορες ερμηνευτικές μεταβλητές χ και έχουν ήδη έτοιμη την τιμή έξοδου η αλλιώς εξαρτημένη μεταβλητή y δηλαδή για παράδειγμα σε ένα συνολο δεδομένων από σχόλια για προϊόντα χ τα οποία έχουν είδη ταξινομηθεί ως θετικά σχόλια και ως αρνητικά σχόλια δηλαδή το μοντέλο έχει είδη έτοιμη την απάντηση στόχος του είναι να καταλάβει το μοτίβο των δεδομένων και να αποφασίσει στην συνέχεια όταν του δώσει ο αναλυτής σχόλια που δεν έχουν κατηγοριοποιεί ως θετικά η αρνητικά τι όντος είναι χωρίς να τα διαβάσει ο ίδιος ο αναλυτής. Το ίδιο ακριβός γίνεται και με τα μοντέλα γραμμικής παλινδρόμησης δηλαδή έστω ότι έχουμε διαφορά χ έστω χρονιά προϋπηρεσίας , φύλο , μορφωτικό επίπεδο και το μισθό που περνούν αυτά τα άτομα και θέλουμε να κάνουμε μια πρόβλεψη για τον μισθό ενός ατόμου οπού έχουμε τα παραπάνω χ χαρακτηριστικά αλλά δεν μας έχει δώσει τον μισθό του. Τέτοιοι αλγόριθμοι είναι η γραμμική παλινδρόμηση οι κ- κοντινότεροι γείτονες ,η λογιστική παλινδρόμηση ,τα svms ,τα νευρωνικα δίκτια και τα δέντρα αποφάσεων κλπ. οπού θα δούμε αναλυτικά σε αυτή την εργασία. Η δεύτερη κατηγορία είναι η μη επιβλεπομένη μάθηση οπού εκεί το μοντέλο δεν έχει λυμένες τις απαντήσεις δηλαδή δεν υπάρχει το Y έχει μόνο ερμηνευτικές μεταβλητές χ και μέσα από αυτές προσπαθεί να γενικεύσει τα δεδομένα τέτοιοι αλγόριθμοι clustering είναι το κ-means ο dbscan και άλλοι. Τέλος η ενισχυμένη μάθηση

αναφέρεται σε αλγορίθμους οπού το σύστημα αλληλοεπιδρά με το περιβάλλον πιο συγκεκριμένα το μοντέλο ονομάζεται πράκτορας (agent) και μαθαίνει μέσο εκείνης της στρατηγικής policy οπού μεγιστοποιεί τις ανταμοιβές οπού έχει βάλει στο ψηφιακό περιβάλλον ο αναλυτής. Δηλαδή το μοντέλο κάνει κάποιες ενέργειες οι οποίες έχουν ως αποτέλεσμα ή την αμοιβή ή την τιμωρία αν το μοντέλο πάρει τις σωστές αποφάσεις με βάση τις επιδιώξεις του αναλυτή ανταμείβεται αν πάρει τις λάθος τιμωρείται. Τέτοια μοντέλα χρησιμοποιούνται για την εκπαίδευση κύριος ρομπότ που διαγωνίζονται σε διαγωνισμούς αλλά και σε βιντεοπαιχνίδια που με αυτό το τρόπο εκπαιδεύονται οι αντίπαλοι ai χαρακτήρες (αντίπαλοι οδηγοί αυτοκίνητων πχ σε παιχνίδια όπως η formula 1).

Στην συνέχεια θα αναλύσουμε την μεθοδολογία ανάπτυξης αλγορίθμων μηχανικής μάθησης αρχικά για να εκπαιδευτεί και να λειτουργήσει σωστά ένας αλγόριθμος μηχανικής μάθησης είναι απαραίτητο να ακολουθηθούν ορισμένα στάδια προετοιμασίας και επεξεργασίας. Η διαδικασία ξεκινά με τη συλλογή των κατάλληλων δεδομένων, τα οποία αποτελούν τη βάση για τη μοντελοποίηση του προβλήματος. Αυτά τα δεδομένα μπορεί να προέρχονται από επίσημες στατιστικές βάσεις, όπως είναι η ΕΛΣΤΑΤ ή η Eurostat, από επιχειρηματικές πλατφόρμες, το χρηματιστήριο ή ακόμη και από ιστοσελίδες που διαθέτουν δεδομένα για ερευνητικούς σκοπούς, όπως το Kaggle. Επιπλέον, είναι δυνατό να προκύψουν δεδομένα μέσω πρωτογενούς δειγματοληψίας, δηλαδή με άμεση συλλογή από τον ίδιο τον αναλυτή. Όσο μεγαλύτερος είναι ο όγκος των δεδομένων, τόσο μεγαλύτερη η ικανότητα του μοντέλου να γενικεύει σωστά το φαινόμενο που αναλύει. Για παράδειγμα, αν ένας αναλυτής μελετά τις τιμές των συμβατικών οχημάτων και μέσα στα δεδομένα συμπεριλάβει και την τιμή μιας Bugatti, τότε η συγκεκριμένη τιμή θεωρείται "εκτός πλαισίου" ή αλλιώς ακραία (outlier). Αυτές οι τιμές μπορούν να διαταράξουν την εκπαίδευση του μοντέλου, καθώς ενδέχεται να προσπαθήσει να τις ενσωματώσει, αντί να ακολουθήσει τη γενική τάση των δεδομένων. Το φαινόμενο αυτό αυξάνει το "θόρυβο" στη διαδικασία μάθησης.

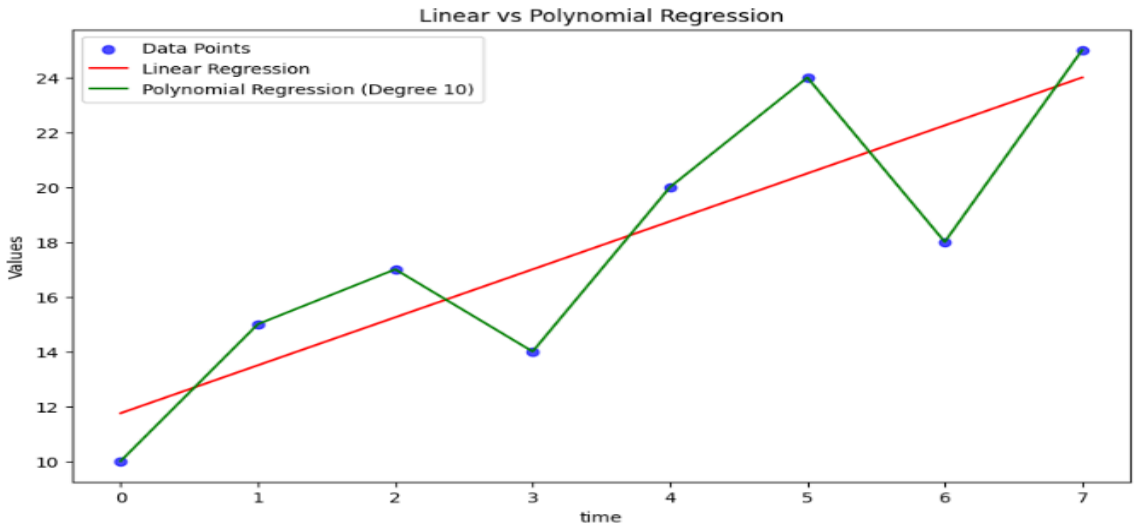
Μετά τη συλλογή, ακολουθεί το στάδιο του καθαρισμού των δεδομένων. Σε αυτό, γίνεται έλεγχος για λανθασμένες ή ελλιπείς εγγραφές. Για παράδειγμα, αν στα δεδομένα υπάρχει άτομο 10 ετών που έχει κάνει αίτηση για σύναψη δάνειου, αυτό αποτελεί λανθασμένη

καταχώριση και πρέπει είτε να διορθωθεί είτε να αφαιρεθεί εντελώς από το σύνολο. Στις περιπτώσεις ελλιπών τιμών, ο αναλυτής μπορεί να επιλέξει μεθόδους όπως η συμπλήρωση με τον μέσο όρο ή με την προηγούμενη διαθέσιμη τιμή φαινόμενο που είναι πολύ συνηθισμένο σε χρονοσειρές, όπως τιμές μετοχών. Για παράδειγμα, αν δεν υπάρχουν τιμές για Σάββατο και Κυριακή, θα μπορούσαν να συμπληρωθούν με την τιμή της Παρασκευής ή με την τιμή της Δευτέρας, εφόσον είναι διαθέσιμη.Στη συνέχεια τα δεδομένα χωρίζονται σε δύο υποσύνολα εκείνα που θα χρησιμοποιηθούν για την εκπαίδευση του μοντέλου και εκείνα που θα αξιοποιηθούν για τον έλεγχο της απόδοσής του. Ένα κλασικό παράδειγμα είναι η διάσπαση του συνόλου σε 80% για εκπαίδευση και 20% για έλεγχο. Υπάρχει επίσης και η τεχνική του cross validation δηλαδή ο τυχαίος διαχωρισμός των δεδομένων από διαφορά σημεία του dataset, το μειονέκτημα αυτής της μεθόδου είναι ότι δεν μπορεί να χρησιμοποιηθεί στις χρονοσειρές, καθώς εκεί η χρονική σειρά των τιμών(δεδομένων) έχει σημασία. Στην συνέχεια ακολουθεί το στάδιο της μετατροπής ή προ επεξεργασίας των χαρακτηριστικών. Σε αυτό το στάδιο μπορεί να γίνει η κανονικοποίηση των δεδομένων (π.χ., να προσαρμοστούν οι τιμές ώστε να έχουν μέση τιμή μηδέν και τυπική απόκλιση ίση με ένα), ή να εφαρμοστούν μετασχηματισμοί όπως οι διαφορές στις χρονοσειρές, για την αντιμετώπιση της επίδρασης ακραίων τιμών. Κατόπιν, γίνεται η επιλογή και ρύθμιση του μοντέλου. Ανάλογα με την περίπτωση, μπορεί να οριστεί ο αριθμός των χρονικών καθυστερήσεων (lags) σε ένα μοντέλο πρόβλεψης, να επιλεγεί η συνάρτηση ενεργοποίησης (όπως ReLU, tanh, sigmoid) στα νευρωνικά δίκτυα, καθώς και να καθοριστεί ο optimizer (όπως ο Adam), το μέγεθος των batch και ο αριθμός των επαναλήψεων (epochs). Όλα αυτά συμβάλλουν στον καθορισμό του τρόπου εκπαίδευσης του μοντέλου.Αφού ολοκληρωθεί η εκπαίδευση, πρέπει να εκτιμηθεί η αποτελεσματικότητά του. Για να γίνει αυτό, συγκρίνονται οι προβλέψεις του με τις πραγματικές τιμές του test set, δηλαδή δεδομένα που δεν είχε δει το μοντέλο κατά την εκπαίδευση. Υπάρχουν διαφορετικοί τρόποι αξιολόγησης, ανάλογα με το είδος του μοντέλου. Για παράδειγμα, για προβλήματα παλινδρόμησης χρησιμοποιούνται δείκτες όπως το MSE ή ο συντελεστής R^2, ενώ σε περιπτώσεις ταξινόμησης μπορεί να χρησιμοποιηθεί το F1-score.Τέλος, αφού διαπιστωθεί ότι το μοντέλο έχει ικανοποιητική απόδοση, έρχεται η φάση της υλοποίησης στον πραγματικό κόσμο. Από εκεί και πέρα, απαιτείται τακτική συντήρηση και αναβάθμιση, ώστε να διατηρείται επίκαιρο και να ανταποκρίνεται στις ενδεχόμενες αλλαγές του περιβάλλοντος ή των δεδομένων.

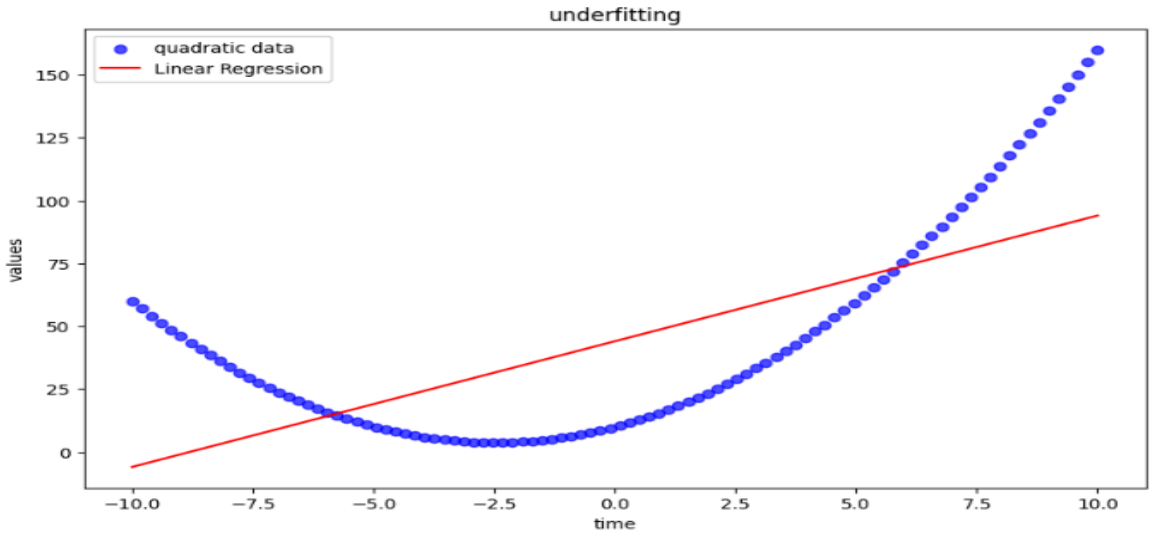
Κατά την ανάπτυξη μοντέλων μηχανικής μάθησης, είναι σύνηθες να προκύπτουν προβλήματα που σχετίζονται με τον βαθμό προσαρμογής του μοντέλου στα δεδομένα. Τα δύο πιο χαρακτηριστικά φαινόμενα είναι το overfitting (υπερπροσαρμογή) και το underfitting (υποπροσαρμογή).

Το φαινόμενο της υπερπροσαρμογής παρατηρείται όταν το μοντέλο μαθαίνει υπερβολικά καλά τα δεδομένα του συνόλου εκπαίδευσης (training set), ακόμα και τον θόρυβο ή τις τυχαίες διακυμάνσεις τους, με αποτέλεσμα να αποδίδει εξαιρετικά στην εκπαίδευση αλλά να αποτυγχάνει στις προβλέψεις σε νέα, άγνωστα δεδομένα (test set).

Αυτό συνήθως προκύπτει όταν χρησιμοποιείται ένα πολύπλοκο μοντέλο σε σχέση με τον όγκο ή την ποιότητα των διαθέσιμων δεδομένων. Το μοντέλο δεν καταφέρνει να γενικεύσει τη βασική τάση των δεδομένων και αντίθετα "θυμάται" κάθε λεπτομέρεια, χάνοντας τη γενική εικόνα. Για παράδειγμα ενα βαθύ νευρωνικό δίκτυο που έχει πάρα πολλά επίπεδα μπορεί να μάθει ακόμα και τις μικρές, μη σημαντικές αποκλίσεις στα δεδομένα εκπαίδευσης, αλλά να αποδώσει πολύ άσχημα σε νέα δεδομένα. Αντίθετα, όταν ένα μοντέλο είναι πολύ απλό για να αποτυπώσει την πολυπλοκότητα των δεδομένων, τότε έχουμε underfitting. Το μοντέλο δεν καταφέρνει να "μάθει" ούτε τα βασικά μοτίβα από το training set, με αποτέλεσμα να παρουσιάζει χαμηλή ακρίβεια και στις προβλέψεις εκπαίδευσης και δοκιμής. Ένα παράδειγμα είναι ένα μοντέλο γραμμικής παλινδρόμησης που προσπαθεί να προβλέψει δεδομένα με μη γραμμικές σχέσεις, δεν θα καταφέρει να αποδώσει σωστά, καθώς είναι "πολύ απλό" για το πρόβλημα. Τα δύο αυτά φαινόμενα συνδέονται στενά με την ισορροπία μεταξύ προκατάληψης (bias) και διασποράς (variance). Το underfitting σχετίζεται με υψηλό bias και χαμηλό variance, ενώ το overfitting σχετίζεται με χαμηλό bias και υψηλό variance. Αυτή η ισορροπία είναι καθοριστική για τη δημιουργία ενός μοντέλου που μπορεί να γενικεύσει σωστά.



Εικονα 35: overfitting



Εικονα 36: underfitting

Στις επόμενες ενότητες ακολουθεί η θεωρητική ανάλυση των πιο γνωστών μεθόδων μηχανικής μάθησης

2.2 ΓΡΑΜΜΙΚΗ ΠΑΛΙΝΔΡΟΜΗΣΗ

Μια από τις πιο ευρέος χρησιμοποιημένες μέθοδους ανάλυσης και πρόβλεψης μελλοντικών τιμών σε ένα σύνολο δεδομένων είναι μέσο της γραμμικής και πολυγαμικής παλινδρόμησης. Στόχος της γραμμικής παλινδρόμησης είναι να βρεθούν εκείνοι οι συντελεστές β0 και β1 μιας ευθείας που ελαχιστοποιούν την διαφορά μεταξύ των πραγματικών τιμών και των τιμών όπου πρόβλεψε το μοντέλο , αυτό επιτυγχάνεται μέσο της ελαχιστοποίησης του μέσου τετραγωνικού σφάλματος για αυτό και η μέθοδος

ονομάζεται και ols(ordinary least squares), για να γίνει ακόμα πιο κατανοητό ουσιαστικά το μοντέλο προσπαθεί να βρει εκείνη την ευθεία γραμμή όπου περνάει ταυτόχρονα και όσο πιο κοντά γίνεται από όλες τις παρατηρήσεις. Υπάρχουν πολλές εφαρμογές της γραμμικής παλινδρόμησης στον πραγματικό κόσμο βέβαια για να μπορέσει το μοντέλο να αποδώσει καλές προβλέψεις θα πρέπει να υπάρχει γραμμική σχέση μεταξύ των ερμηνευτικών μεταβλητών(X) και της εξαρτημένης μεταβλητής (Y), παραδείγματα αυτών είναι η πρόβλεψη του μισθού ενός εργαζομένου (εξαρτημένη μεταβλητή) με βάση τα χρονιά προϋπηρεσία του (ερμηνευτική μεταβλητή χ1) τα χρονιά εκπαίδευσης του (ερμηνευτική μεταβλητή χ2) κλπ. Η εξίσωση της γραμμικής παλινδρόμησης είναι η παρακάτω.

$$Y = b0 + b1 * x1 + b2 * x2 + \dots bn * xn$$

Οπού Υ είναι η εξαρτημένη μεταβλητή πχ ο μισθός των εργαζομένων δηλαδή η μεταβλητή για την οποία θέλουμε να προβλέψουμε τις μελλοντικές τιμές της, χ1,χ2 κλπ. είναι οι ερμηνευτικές μεταβλητές δηλαδή οι μεταβλητές που μέσο αυτών θέλουμε να μοντελοποιησουμε την διακύμανση της εξαρτημένης μεταβλητής πχ τα χρονιά προϋπηρεσίας, τα χρονιά εκπαίδευσης , ο αριθμός των παιδειών του εργαζομένου κλπ. β0 είναι η σταθερά και β1, β2 είναι οι συντελεστές κλήσης της παλινδρόμησης. Οι συντελεστές β1,β2 κλπ. μοιάζουν σε κάποιο βαθμό με τον συντελεστή συσχέτισης που είδαμε στο κεφάλαιο 2 αν οι συντελεστές βν είναι θετικοί τότε υπάρχει θετική συσχέτιση μεταξύ της ερμηνευτικής μεταβλητής και της εξαρτημένης μεταβλητής παραδείγματος χάρη όσο περισσότερα χρονιά προϋπηρεσίας έχει ένας υπάλληλος τόσο παραπάνω αυξάνεται και ο μισθός του αντίθετα αν το βν είναι αρνητικό τότε υπάρχει αρνητική συσχέτιση με την εξαρτημένη μεταβλητή για παράδειγμα αν Υ είναι οι πωλήσεις παγωτών και έχουμε ως ερμηνευτική μεταβλητή χ1 μια ψευδομεταβλητη όπου παίρνει την τιμή 1 όταν είναι χειμώνας και την τιμή μηδέν όταν είναι καλοκαίρι τότε υπάρχει αρνητική συσχέτιση μεταξύ των δυο μεταβλητών. Όπως αναφέραμε και παραπάνω το μοντέλο της παλινδρόμησης είναι τόσο χρήσιμο που υπάρχει προ εγκατεστημένο σε πολλά προγράμματα όπως το excel , παρακάτω ακολουθεί ένα παράδειγμα υπολογισμού με χειροκίνητο τρόπο δηλαδή μέσο πολλαπλασιασμού πινάκων στο excel πιο συγκεκριμένα έχει επιλεγεί ως ερμηνευτική μεταβλητή τα έτη αυξανόμενα από το 1 έως το 10 και ως εξαρτημένη μεταβλητή τα αντίστοιχα κέρδη εκείνου του έτους της εταιρείας ‘ΧΛ’. Ο τύπος για την εύρεση των συντελεστών β είναι ο παρακάτω.

$$\hat{b} = inverse(xtranspose * x) * xtranspose * y$$

Τα δεδομένα φαίνονται στην παρακάτω εικόνα

YEAR	Y(YEARLY SALES)	X0	X1
2014	9050151	1	1
2015	7060215	1	2
2016	6356855	1	3
2017	7843482	1	4
2018	9491501	1	5
2019	9372543	1	6
2020	6120439	1	7
2021	10266591	1	8
2022	16630862	1	9
2023	13316742	1	10

Βάζουμε ως χ0 την τιμή 1 σε ολόκληρη την στήλη ώστε να μπορέσει το μοντέλο να υπολογίσει την σταθερά β0. Στην συνέχεια δημιουργούμε τον ανάστροφο πίνακα των ερμηνευτικών μεταβλητών.

X TRANSPOSE

1	1	1	1	1	1	1	1	1	1
1	2	3	4	5	6	7	8	9	10

Εν συνέχεια πολλαπλασιάζουμε τον ανάστροφο πίνακα χ με τον αρχικό πίνακα χ

XTR*X	10	55
	55	385

και στην συνέχεια το πίνακα που προκύπτει τον αναστρέφουμε

INV(XTR*X)	0,466666667	-0,06667
	-0,06666667	0,012121

Εχοντας υπολογισει το πρωτο σκλελος του πολλαπλασιασμου συνεχιζουμε πολλαπλασιαζοντας τον πινακα με τα δεδεομενα της εξαρτημενης μεταβλητης Y με τον αναστρτοφο πινακα των ερμηνευτικων μεταβλητων χ.

XTR*Y	95509381
	585128816

Τελος για τον υπολογισμο των συντελεστων β πολλαπλασιαζουμε αυτους τους δυο πινακες το αποτελεσμα είναι το παρακατω. Το οποιο επιβεβαιονεται και από το αντιστοιχο στατιστικο πακετο με την ετοιμη συναρτηση παλινδρομισης του excel.

INV(XTR*X)*XTRY	β0=	5562456,733
	β1=	725178,4303

Στατιστικά παλινδρόμησης	
Πολλαπλό R	0,671578
R Τετράγωνο	0,451018
Τυπικό σφάλμα	2569268
Μέγεθος δείγματος	10

ΑΝΑΛΥΣΗ ΔΙΑΚΥΜΑΝΣΗΣ

	βαθμοί ελευθερίας	*SS*	*MS*	*F*	*Σημαντικότητα F*
Παλινδρόμηση	1	4,34E+13	4,34E+13	6,572415	0,033455
Υπόλοιπο	8	5,28E+13	6,6E+12		
Σύνολο	9	9,62E+13			

	Συντελεστές	*Τυπικό σφάλμα*	*t*	*τιμή-P*
Τεταγμένη επί την αρχή	5562457	1755144	3,169231	0,01321
Μεταβλητή X 1	725178,4	282867,1	2,563672	0,033455

Εικονα 37: αποτελεσματα γραμμικης παλινδρομισης στο excel

Το στατιστικο πακετο εξαγει ταυτοχρονα πολλα επιπροσθετα στατιστικα όπως το p-value που εξηγισαμε στο κεφαλαιο 2 , με το οποιο διαπιστονουμε στο συγκερημενο παραδειγμα ότι και το β0 και το β1 είναι στατιστικα σηματικοι ακομα και με α=5% καθως το p-value και των δυο είναι μικροτερο από 0,05.

2.3 ΛΟΓΙΣΤΙΚΗ ΠΑΛΙΝΔΡΟΜΙΣΗ

Η λογιστική παλινδρόμηση (logistic regression) αποτελεί μία από τις πιο θεμελιώδεις μεθόδους στη μηχανική μάθηση και στη στατιστική για την αντιμετώπιση προβλημάτων δυαδικής ταξινόμησης. Χρησιμοποιείται όταν η εξαρτημένη μεταβλητή Y είναι κατηγορική με δύο δυνατές τιμές, συνήθως 0 και 1. Στόχος του μοντέλου είναι να προσδιορίσει την πιθανότητα ένα παρατηρούμενο σύνολο χαρακτηριστικών (ερμηνευτικές μεταβλητές X) να ανήκει σε μία από τις δύο κατηγορίες. Για παράδειγμα, έστω ότι διαθέτουμε ένα σύνολο χαρακτηριστικών όπως η ηλικία ενός ατόμου, η θερμοκρασία του σώματος, η παρουσία βήχα κ.ά., και θέλουμε να προβλέψουμε εάν το άτομο είναι ασθενές (κατηγορία 1) ή υγιές (κατηγορία 0). Η λογιστική παλινδρόμηση προσπαθεί να κατασκευάσει ένα μοντέλο που, βάσει αυτών των μεταβλητών, να μπορεί με ακρίβεια να ταξινομεί νέα παραδείγματα στην κατάλληλη κατηγορία. Σε αντίθεση με τη γραμμική παλινδρόμηση, η οποία επιδιώκει να ελαχιστοποιήσει το άθροισμα τετραγώνων σφαλμάτων (SSE), η λογιστική παλινδρόμηση βασίζεται στη μέθοδο της μέγιστης πιθανοφάνειας (Maximum Likelihood Estimation MLE) που παρουσιάζεται παρακάτω. Μέσω αυτής, υπολογίζονται οι συντελεστές των ερμηνευτικών μεταβλητών έτσι ώστε να μεγιστοποιηθεί η πιθανότητα οι προβλέψεις του μοντέλου να ανταποκρίνονται στα πραγματικά δεδομένα.

$$J(\mathbf{w}, b) = \frac{1}{m} \sum_{i=1}^{m} \left[-y^{(i)} \log(\hat{y}^{(i)}) - (1 - y^{(i)}) \log(1 - \hat{y}^{(i)}) \right]$$

Για την εφαρμογή της λογιστικής παλινδρόμησης ο αναλυτής τοποθετεί τις ερμηνευτικές μεταβλητές X σε ένα διάνυσμα ταυτόχρονα το μοντέλο υπολογίζει το διάνυσμα των

συντελεστών w οι οποίοι είναι όσοι και ο αριθμός των ερμηνευτικών μεταβλητών. Εν συνέχεια το μοντέλο υπολογίζει την γραμμική πρόβλεψη με βάση τον παρακατω τύπο

$$z = \mathbf{w}^T\mathbf{x} + b = w_1x_1 + w_2x_2 + \cdots + w_nx_n + b$$

Επακόλουθο της παραπάνω ενέργειας είναι η εφαρμογή της sigmoid συνάντησης στο αποτέλεσμα έτσι ώστε να έχει την μορφή πιθανότητας.

$$\hat{y} = \sigma(z) = \frac{1}{1 + e^{-(\mathbf{w}^T\mathbf{x}+b)}}$$

Και τέλος εάν το αποτέλεσμα της παραπάνω μετατροπής είναι μεγαλύτερο από το 0,5 η εξαρτημένη μεταβλητή τοποθετείται στην κατηγορία 1 ενώ εάν είναι μικρότερη από το 0,5 τότε ταξινομείται στην κατηγορία 0

$$\hat{y} = \{0\ OTAN\ \sigma(\zeta) <\ 0.5\ 1\ OTAN\ \sigma(\zeta) >\ 0.5$$

2.4 Κ-ΚΟΝΤΕΙΝΟΤΕΡΟΙ ΓΕΙΤΟΝΕΣ

Ο αλγόριθμος κ κοντινότερων γειτόνων (K-NN) αποτελεί μια από τις πιο απλές και διαισθητικές μεθόδους ταξινόμησης στην μηχανική μάθησή. Η βασική του ιδέα στηρίζεται στην αναλογία και την εγγύτητα μεταξύ των παρατηρήσεων δηλαδή ότι ένα άγνωστο δεδομένο ταξινομείται με βάση την κατηγορία (ετικέτα) των κοντινότερων του γειτόνων. Για να καταλάβουμε τον τρόπο λειτουργίας του αλγορίθμου θα δούμε ένα απλό παράδειγμα έστω ότι μια εταιρία ρούχων θέλει να προβλέψει το κατάλληλο μέγεθος ενδύματος (π.χ. μικρό ή μεγάλο) για έναν νέο πελάτη, βασιζόμενη σε δύο χαρακτηριστικά το ύψος και το βάρος του. Τα δεδομένα που έχουμε στη διάθεσή μας περιλαμβάνουν τις μετρήσεις προηγούμενων πελατών και το μέγεθος που τελικά αγόρασαν. Η διαδικασία ξεκινάει ως εξής αρχικά ο αναλυτής ορίζει των αριθμό των κοντινότερων γειτόνων που θα ληφθούν υπόψη για την ταξινόμηση έστω στο παράδειγμα μας ο αριθμός των γειτόνων κ θα είναι ίσος με 5, στην συνέχεια για τον νέο πελάτη ο αλγόριθμος υπολογίζει την ευκλείδεια απόσταση (υπάρχουν και άλλοι τρόποι υπολογισμού της απόστασης όπως ο Manhattan) μετάξι των τιμών που μας έδωσε ο πελάτης και των ειδή υπαρχόντων παρατηρήσεων, εντοπίζουμε τους κ κοντινότερους γείτονες από το σύνολο δεδομένων και

με βάση την πλειοψηφία αυτών των γειτόνων ταξινομείται ο πελάτης στην αντίστοιχη κατηγορία, στο παράδειγμα μας στο excel οι κοντινότεροι γείτονες είχαν ετικέτες μικρο,μικρο,μικρο,μικρο,μεγαλο αρά 4 γείτονες σηματοδοτούν ότι με βάση αυτά τα χαρακτηριστικά ο πελάτης θα πρέπει να αγοράσει μικρό μέγεθος μπλούζας. Παρότι ο αλγόριθμος είναι εύκολος και κατανοητός το δύσκολο για τον αναλυτή είναι η επιλογή του βέλτιστου αριθμού κ καθώς ένα μικρό κ οδηγεί το μοντέλο στο να παίρνει βιαστικές και γρήγορες αποφάσεις που μπορεί να επηρεάζονται από τον θόρυβο ενώ ένα μεγάλο κ κάνει την εκτέλεση του αλγορίθμου χρονοβόρα και ίσως πάλι ο αλγόριθμος να μην μπορέσει να οδηγηθεί στο βέλτιστο αποτέλεσμα.

SUM =SQRT((A21-A10)^2+(B21-B10)^2)

	A	B	C	D	E
1	**υψος σε εκατοστα**	**βαρος σε κιλα**	**μεγεθος μπλουζας**	**ευκλιδια αποσταση**	**αριθμιση αποστασης**
2	158	58	small	4,242640687	
3	158	59	small	3,605551275	
4	158	63	small	3,605551275	
5	160	59	small	2,236067977	3
6	160	60	small	1,414213562	1
7	163	60	small	2,236067977	3
8	163	61	small	2	2
9	160	64	large	3,16227766	5
10	163	64	large	^2+(B21-B10)^2)	
11	165	61	large	4	
12	165	62	large	4,123105626	
13	165	65	large	5,656854249	
14	168	62	large	7,071067812	
15	168	63	large	7,280109889	
16	168	66	large	8,602325267	
17	170	63	large	9,219544457	
18	170	64	large	9,486832981	
19	170	68	large	11,40175425	
20	**δεδομενα νεου πελατη**				
21	**161**	**61**			
22	**εστω k = 5**				

Εικόνα 38: εφαρμογή κ-κοντινότερων γειτόνων στο excel

2.5 SUPPORT VECTOR MACHINE (SVM)

Οι Μηχανές Διανυσμάτων Υποστήριξης (Support Vector Machines - SVM) αποτελούν έναν από τους πιο γνωστούς αλγορίθμους επιβλεπόμενης μάθησης και εφαρμόζονται κυρίως σε προβλήματα ταξινόμησης, αν και υπάρχουν επεκτάσεις που επιτρέπουν τη

χρήση τους και σε προβλήματα παλινδρόμησης που θα δούμε στο κεφάλαιο 7. Ο πυρήνας της μεθοδολογίας τους βασίζεται στην εύρεση ενός υπερεπιπέδου (hyperplane) που διαχωρίζει τις παρατηρήσεις διαφορετικών κατηγοριών στον χώρο με το μέγιστο δυνατό περιθώριο (margin).Στο πλαίσιο ενός γραμμικά διαχωρίσιμου προβλήματος δύο κατηγοριών, το SVM επιχειρεί να βρει το υπερεπίπεδο εκείνο που μεγιστοποιεί την απόσταση μεταξύ των δύο κλάσεων αυτή η απόσταση καθορίζεται από τις πλησιέστερες παρατηρήσεις κάθε κατηγορίας, γνωστές ως διανύσματα υποστήριξης (support vectors). Το υπερεπίπεδο μπορεί να περιγράφει μαθηματικά ως:

$$w^T * X + B = 0$$

Οπού χ είναι οι ερμηνευτικές μεταβλητές β το σφάλμα και w το διάνυσμα των βαρών.

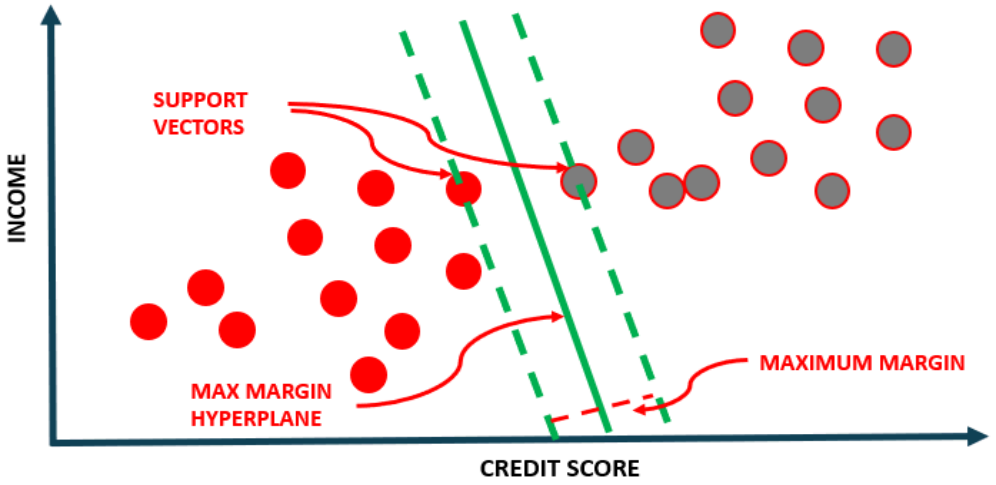
Στόχος του μοντέλου είναι να επιλυθεί η παρακατω συνάρτηση με την εύρεση εκείνων των βαρών που την ελαχιστοποιούν, το Υι παίρνει τιμές από -1 έως 1.

$$\min_{w,b} \frac{1}{2}\|w\|^2 \quad \text{υπό τους περιορισμούς: } y_i(w^T x_i + b) \geq 1 \quad \forall i$$

Σε περιπτώσεις όπου τα δεδομένα δεν είναι γραμμικά διαχωρίσιμα, εισάγονται μεταβλητές χαλάρωσης (slack variables) και ένας όρος κανονικοποίησης ελέγχει το βαθμό ανοχής στα σφάλματα ταξινόμησης.

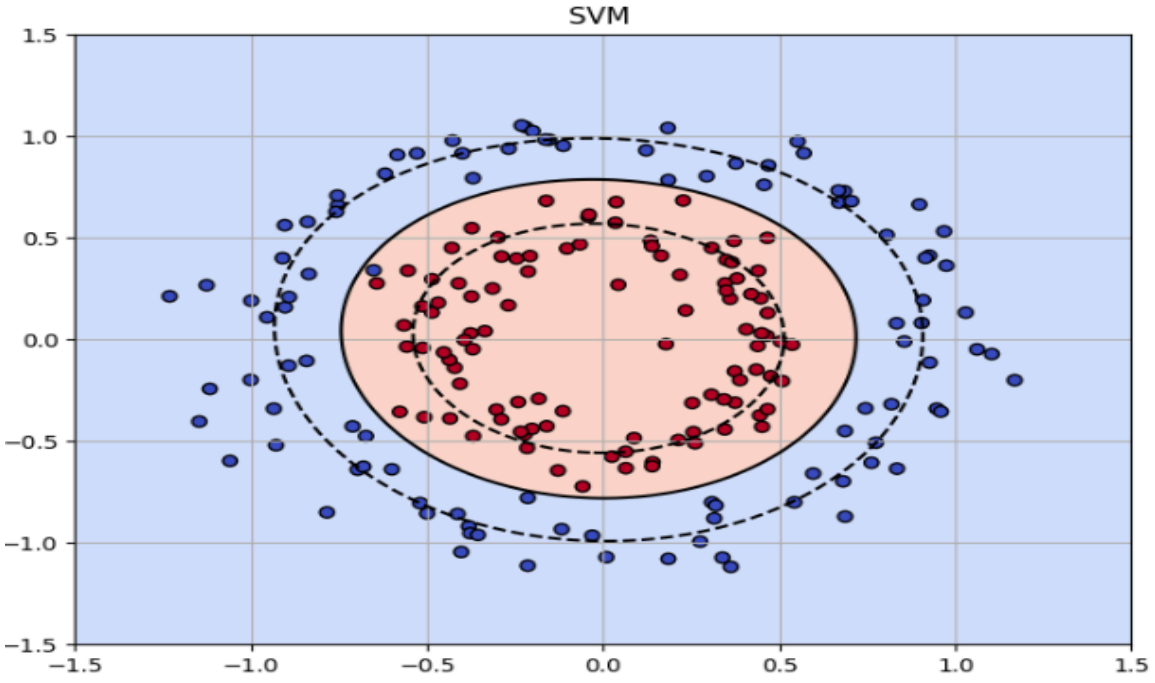
$$\min_{w,b,\xi} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n} \xi_i \quad \text{υπό τους περιορισμούς: } y_i(w^T x_i + b) \geq 1 - \xi_i,\ \xi_i \geq 0$$

Η παράμετρος C ελέγχει το trade-off μεταξύ του μεγάλου περιθωρίου και των λαθών ταξινόμησης στο σύνολο εκπαίδευσης. Ένα μεγάλο C τιμωρεί περισσότερο τις λανθασμένες ταξινομήσεις, ενώ ένα μικρό C ενισχύει τη γενίκευση του μοντέλου επιτρέποντας μεγαλύτερη ανοχή στα σφάλματα.



Εικόνα 39: svm

Για δεδομένα που δεν είναι γραμμικά διαχωρίσιμα ακόμη και με slack variables, το SVM αξιοποιεί τη μέθοδο των πυρήνων (kernel trick), ώστε να χαρτογραφήσει τα δεδομένα σε έναν χώρο υψηλότερων διαστάσεων όπου είναι πιο πιθανό να είναι διαχωρίσιμα γραμμικά. Ο kernel είναι ουσιαστικά μια συνάρτηση ομοιότητας που υπολογίζει το εσωτερικό γινόμενο δύο παρατηρήσεων σε αυτόν τον νέο χώρο χωρίς να απαιτείται ο ρητός υπολογισμός των μετασχηματισμένων διανυσμάτων. Μερικοί από τους πιο γνωστούς πυρήνες είναι ο γραμμικός ,ο πολυονυμικος ,ο sigmoid και ο rbf.



Εικόνα 40: svm με χρήση kernal

Με παρόμοιο τρόπο λειτουργεί ο αλγόριθμος και για την μοντελοποιηση χρονοσειρων μόνο που τώρα συμβολίζεται με svr από το regression. Η βασική ιδέα πίσω από το SVR είναι να προσδιοριστεί μία συνάρτηση f(x) η οποία αποκλίνει το πολύ κατά ϵ από τις πραγματικές τιμές των δεδομένων εκπαίδευσης και ταυτόχρονα είναι όσο το δυνατόν πιο επίπεδη. Στόχος του μοντέλου είναι να ελαχιστοποιεί η απόσταση μεταξύ των πραγματικών τιμών σε σχέση με εκείνες τις τιμές όπου πρόβλεψε το μοντέλο με βάση την παρακάτω συνάρτηση.

$$\min_{w,b,\xi,\xi^*} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}(\xi_i + \xi_i^*)$$

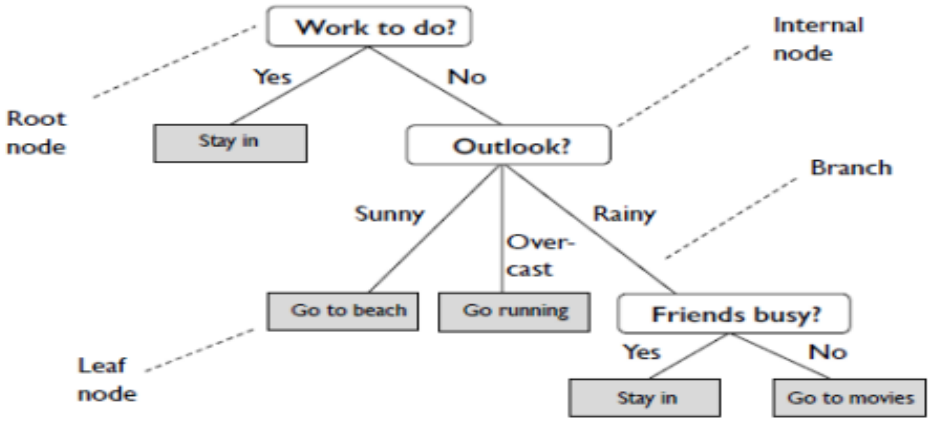
Υπό τους περιορισμούς

$$\begin{cases} y_i - \langle w, x_i\rangle - b \leq \epsilon + \xi_i \\ \langle w, x_i\rangle + b - y_i \leq \epsilon + \xi_i^* \\ \xi_i, \xi_i^* \geq 0 \end{cases}$$

Οπού χ είναι οι ερμηνευτικές μεταβλητές Υ είναι η εξαρτημένη μεταβλητή(δηλαδή η μεταβλητή για την οποία ο αναλυτής θέλει να δημιουργήσει προβλέψεις), β είναι το σφάλμα, c το hyperplane και ε το πλάτος ανοχής. Η τελική συνάρτηση παλινδρόμησης που προκύπτει είναι η παρακάτω όπου αι και α*ι είναι οι συντελεστές lagrance και κ είναι η συνάρτηση του πυρήνα.

$$f(x) = \sum_{i=1}^{n}(\alpha_i - \alpha_i^*)K(x_i, x) + b$$

2.6 ΔΕΝΤΡΑ ΑΠΟΦΑΣΕΩΝ ΚΑΙ ΤΥΧΑΙΑ ΔΑΣΗ



εικόνα 41: αναπαράσταση δέντρου αποφάσεων

Τα δέντρα απόφασης αποτελούν έναν από τους πιο γνωστούς και χρησιμοποιούμενους αλγορίθμους ταξινόμησης στη μηχανική μάθηση. Η βασική τους λειτουργία είναι η λήψη αποφάσεων με τη χρήση ενός δέντρου, όπου κάθε κόμβος αντιπροσωπεύει μία απόφαση ή μία ερώτηση πάνω σε ένα χαρακτηριστικό των δεδομένων και κάθε φύλλο του δέντρου αντιπροσωπεύει την τελική ταξινόμηση ή απόφαση. Ο αλγόριθμος των δέντρων απόφασης χρησιμοποιείται ευρέως για την ταξινόμηση δεδομένων, αλλά και για την παλινδρόμηση.

Ο αλγόριθμος που ακολουθεί η κατασκευή των δέντρων απόφασης, μπορεί να συνοψιστεί σε δύο βασικά στάδια την κατασκευή του δέντρου απόφασης και την εφαρμογή του δέντρου για ταξινόμηση νέων παρατηρήσεων.

Η διαδικασία κατασκευής του δέντρου απόφασης περιλαμβάνει την επιλογή του καλύτερου χαρακτηριστικού για κάθε κόμβο, το οποίο θα "διαιρέσει" τα δεδομένα με τον καλύτερο τρόπο. Η επιλογή του κατάλληλου χαρακτηριστικού σε κάθε κόμβο του δέντρου γίνεται με βάση διάφορα κριτήρια, τα πιο συνηθισμένα από τα οποία είναι η εντροπία και το information gain. Η εντροπία είναι ένα μέτρο αβεβαιότητας. Όταν ένα σύνολο δεδομένων είναι εντελώς καθαρό (δηλαδή όλα τα δείγματα ανήκουν στην ίδια κατηγορία), η εντροπία είναι μηδέν. Όταν το σύνολο των δεδομένων είναι ομοιόμορφα κατανεμημένο μεταξύ των κατηγοριών, η εντροπία είναι μέγιστη. Ο στόχος είναι να επιλέξουμε το χαρακτηριστικό που θα μειώσει την αβεβαιότητα (δηλαδή την εντροπία) όσο το δυνατόν περισσότερο. Ο τύπος για την εντροπία E(S) ενός συνόλου S είναι:

$$E(S) = -\sum_{i=1}^{m} p_i \log_2 p_i$$

όπου pi είναι η πιθανότητα εμφάνισης της i κατηγορίας στο σύνολο S. Η διαχωριστική ικανότητα ενός χαρακτηριστικού A είναι η μείωση της εντροπίας που προκαλεί η διάσπαση του συνόλου δεδομένων S με βάση την τιμή του χαρακτηριστικού A. Υπολογίζεται ως η διαφορά της εντροπίας του αρχικού συνόλου από την αναμενόμενη εντροπία των υποσυνόλων μετά τον διαχωρισμό. Ο τύπος για την πληροφορία κέρδους είναι:

$$IG(S, A) = E(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} E(S_v)$$

όπου Sv είναι το υποσύνολο του S που περιέχει παραδείγματα με την τιμή v για το χαρακτηριστικό A.

Η διαδικασία κατασκευής του δέντρου απόφασης ακολουθεί τα εξής βήματα: πρώτα ξεκινάμε με το σύνολο δεδομένων στην κορυφή του δέντρου (ρίζα). Στη συνέχεια, επιλέγουμε το χαρακτηριστικό που θα παρέχει την καλύτερη διαίρεση του συνόλου δεδομένων. Η "καλύτερη" διάκριση είναι αυτή που ελαχιστοποιεί την εντροπία ή μεγιστοποιεί την πληροφορία κέρδους. Στη συνέχεια, χωρίζουμε το σύνολο δεδομένων σε υποσύνολα, με βάση τις τιμές του επιλεγμένου χαρακτηριστικού και επαναλαμβάνουμε τη διαδικασία για κάθε υποσύνολο, μέχρι να ικανοποιηθούν κάποιοι όροι τερματισμού (π.χ., το υποσύνολο είναι καθαρό ή το μέγεθος του υποσυνόλου είναι μικρότερο από ένα όριο).

Μετά την κατασκευή του δέντρου απόφασης, μπορούμε να το χρησιμοποιήσουμε για να ταξινομήσουμε νέα παραδείγματα δεδομένων. Η διαδικασία αυτή είναι απλή: αρχίζουμε από τη ρίζα του δέντρου και εφαρμόζουμε τις ερωτήσεις που υπάρχουν σε κάθε κόμβο σχετικά με τα χαρακτηριστικά των δεδομένων. Αν το χαρακτηριστικό ικανοποιεί την ερώτηση, ακολουθούμε τον κόμβο προς τα αριστερά, αλλιώς προς τα δεξιά. Αυτή η διαδικασία συνεχίζεται μέχρι να φτάσουμε σε ένα φύλλο του δέντρου, το οποίο παρέχει την τελική απόφαση (π.χ., ταξινόμηση σε μία κατηγορία).

Οι αλγόριθμοι ID3, C4.5 και CART είναι οι πιο γνωστοί αλγόριθμοι για την κατασκευή δέντρων απόφασης. Ο αλγόριθμος ID3 είναι ο πρώτος αλγόριθμος που εισήγαγε την κατασκευή δέντρων απόφασης χρησιμοποιώντας την έννοια της πληροφορίας κέρδους. Για κάθε κόμβο, επιλέγεται το χαρακτηριστικό που ελαχιστοποιεί την εντροπία και μεγιστοποιεί την πληροφορία κέρδους. Ο ID3 τερματίζει όταν δεν υπάρχουν άλλα χαρακτηριστικά για διάσπαση ή όταν όλα τα δεδομένα σε έναν κόμβο ανήκουν στην ίδια κατηγορία. Ο C4.5 είναι η βελτίωση του αλγορίθμου ID3 και εισάγει δύο βασικές βελτιώσεις: τη χρήση του λόγου πληροφοριακού κέρδους προς την εντροπία για τη λήψη αποφάσεων σχετικά με την επιλογή των χαρακτηριστικών και την ανάπτυξη υποδέντρων με χρήση αποκοπής (pruning) για την αποφυγή της υπερπροσαρμογής (overfitting). Ο C4.5 δημιουργεί δέντρα αποφάσεων με συνεχιζόμενα χαρακτηριστικά, χρησιμοποιώντας την έννοια της αναλογίας της πληροφορίας κέρδους. Ο αλγόριθμος CART (Δέντρα Ταξινόμησης και Παλινδρόμησης) χρησιμοποιεί την έννοια του Gini Index ή του Μέσου Τετραγωνικού Σφάλματος (MSE) για να επιλέξει τα καλύτερα χαρακτηριστικά, αντί για την πληροφορία κέρδους.

Έχοντας εξηγήσει τα δέντρα αποφάσεων συνεχίζουμε με την ανάλυση των τυχαίων δασών. Η κεντρική ιδέα των Τυχαίων Δασών είναι να κατασκευάσουν έναν δάσος από πολλά, ανεξάρτητα μεταξύ τους, δέντρα απόφασης. Κάθε δέντρο εκπαιδεύεται σε ένα διαφορετικό υποσύνολο των δεδομένων και με ένα υποσύνολο χαρακτηριστικών, ώστε να μειωθεί η συσχέτιση (correlation) μεταξύ των δέντρων και να ενισχυθεί η γενικευτική ικανότητα του συνόλου.Ο αλγόριθμος περιλαμβάνει τα εξής βήματα:

Το πρωτο βημα είναι το Bootstrap Aggregation οπου για κάθε δέντρο, δημιουργείται τυχαία ένα υποσύνολο των δεδομένων εκπαίδευσης με επαναληπτική δειγματοληψία (δηλαδή, το ίδιο δείγμα μπορεί να επιλεγεί περισσότερες από μία φορές). Αυτό το υποσύνολο λέγεται bootstrapped sample. Το επομενο βημα είναι η τυχαία επιλογή χαρακτηριστικών κατά την κατασκευή κάθε κόμβου του δέντρου, δεν εξετάζονται όλα τα χαρακτηριστικά για διαχωρισμό, αλλά μόνο ένα τυχαίο υποσύνολο. Αυτό οδηγεί σε μεγαλύτερη διαφοροποίηση μεταξύ των δέντρων και μειώνει την πιθανότητα overfitting. Το τριτο βημα είναι η κατασκευή πολλών δέντρων, ο αριθμός των δέντρων είναι μια υπερπαράμετρος του μοντέλουΣτην συνεχεια γινεται ο συνδυασμός των προβλέψεων οπου, κάθε δέντρο ψηφίζει για μία κατηγορία και η τελική πρόβλεψη προκύπτει από τη πλειοψηφία των ψήφων (majority voting). Ενώ στην παλινδρόμηση, υπολογίζεται ο μέσος όρος των προβλέψεων όλων των δέντρων.

2.7 ΜΕΘΟΔΟΙ ΑΞΙΟΛΟΓΗΣΗΣ ΑΛΓΟΡΙΘΜΩΝ ΜΗΧΑΝΙΚΗΣ ΜΑΘΗΣΗΣ ΓΙΑ ΧΡΟΝΟΣΕΙΡΕΣ

Ολοκληρώνοντας την θεωρία των πίσω από ανάπτυξη των αλγορίθμων μηχανικής μάθησης θα εστιάσουμε σε αυτήν την ενότητα στις μεθόδους αξιολόγησης των προβλέψεων που προκύπτουν από αυτούς. Η αξιολόγηση των αλγορίθμων κατά κύριο λόγο εφαρμόζεται στις προβλέψεις των μοντέλων στα δεδομένα του test set.

Η πιο γνωστή μέθοδος και αυτή που χρησιμοποιήθηκε κατά κύριο λόγο στην εργασία είναι το μέσο τετραγωνικό σφάλμα (mean square error) το οποίο είναι το άθροισμα των διαφορών των πραγματικών τιμών σε σχέση με τις τιμές όπου πρόβλεψε το μοντέλο υψωμένες στο τετράγωνο και διαιρεμένο με το σύνολο των παρατηρήσεων. Ο λόγος που υψώνουμε τις διαφορές στο τετράγωνο είναι έτσι ώστε να μην επηρεάζεται το άθροισμα από τις παρατηρήσεις με αρνητικό πρόσημο καθώς αυτή η μέθοδος μετράει την απόσταση των πραγματικών τιμών από αυτές που πρόβλεψε το μοντέλο. Το θετικό αυτού του μέτρου είναι η ευκολία χρήσης του αλλά και το γεγονός ότι μπορεί να χρησιμοποιηθεί ως μετρό αξιολόγησης για απλές μεθόδους ανάλυσης χρονοσειρων όπως ο κινητός μέσος ορός μέχρι και σε πιο σύνθετα μοντέλα όπως τα νευρωνικα δίκτυα. Βέβαια το μεγάλο μειονέκτημα του είναι ότι επηρεάζεται ευκολά από ακραίες τιμές και ότι το αποτέλεσμα (σφάλμα) που εξάγεται είναι σε τετραγωνικές μονάδες μέτρησης.

$$mse = \frac{sum(\left(yi - \hat{yi}\right)^2))}{N}$$

Οπού Yi είναι η εκάστοτε πρόβλεψη του μοντέλου , και $\hat{yi}$ είναι η αντιστοιχη πραγματικη τιμη και N είναι ο αριθμος των παρατιρισεων του δειγματος.

Η λύση στο πρόβλημα του σφάλματος το οποίο είναι σε τετραγωνικές μονάδες στο mse λύνεται μέσο του rmse δηλαδή της ρίζας του μέσου τετραγωνικού σφάλματος καθώς το σφάλμα που προκύπτει είναι στην ιδιά μονάδα μέτρησης με τα δεδομένα του dataset.

$$rootmse = \sqrt{\frac{sum(\left(yi - \hat{yi}\right)^2))}{N}}$$

Οπού πάλι το Yi είναι η εκάστοτε πρόβλεψη του μοντέλου , το $\hat{yi}$ είναι η αντιστοιχη πραγματικη τιμη και N είναι ο αριθμος των παρατιρισεων του δειγματος.

Το επόμενο μετρό αξιολόγησης είναι το μέσο απολυτό σφάλμα (mae) το οποίο υπολογίζει την μέση απολυτή διαφορά μεταξύ των πραγματικών τιμών και των αντίστοιχων προβλέψεων. Η διαφορά του σε σχέση με το μέσο τετραγωνικό σφάλμα είναι ότι οι τιμές του σφάλματος είναι εκφρασμένες στην αρχική τους μορφή και ότι δεν επηρεάζεται τόσο ευκολά όπως στο mse από ακραίες τιμές.

$$mae = \frac{sum(|yi - \hat{yi}|)}{N}$$

Οπού για άλλη μια φορά το Yi είναι η εκάστοτε πρόβλεψη του μοντέλου , και $\hat{yi}$ είναι η αντιστοιχη πραγματικη τιμη και τελος το N είναι ο αριθμος των παρατιρισεων του δειγματος.

Ένα ακόμα μετρό αξιολόγησης των προβλέψεων είναι το μέσο απολυτό ποσοστιαίο σφάλμα(mape) το οποίο μετράει το μέσο ποσοστό σφάλματος μεταξύ των πραγματικών και των προβλεπόμενων τιμών. Το αποτέλεσμα που προκύπτει είναι εκφρασμένο σε ποσοστό γεγονός που το κάνει πιο εύκολο στην κατανόηση του για παράδειγμα αν το αποτέλεσμα είναι 9% αυτό σημαίνει ότι το μοντέλο έχει κατά μέσο ορό 9% σφάλμα στις προβλέψεις του. Το αρνητικό της χρήσης του είναι ότι δεν μπορεί να χρησιμοποιηθεί εάν οι πραγματικές τιμές είναι κοντά ή ίσες με το μηδέν και ότι όπως το Mae δεν επηρεάζεται και κατά επέκταση δεν τιμωρεί τις ακραίες τιμές στις προβλέψεις.

$$mape = \frac{100}{N} * sum|\frac{yi - \hat{yi}}{yi}|$$

Τέλος ένα μετρό αξιολόγησης που χρησιμοποιείται κατά κύριο λόγο στις παλινδρομήσεις είναι το R^2 το οποίο είναι αποτέλεσμα της παρακατω διαίρεσης.

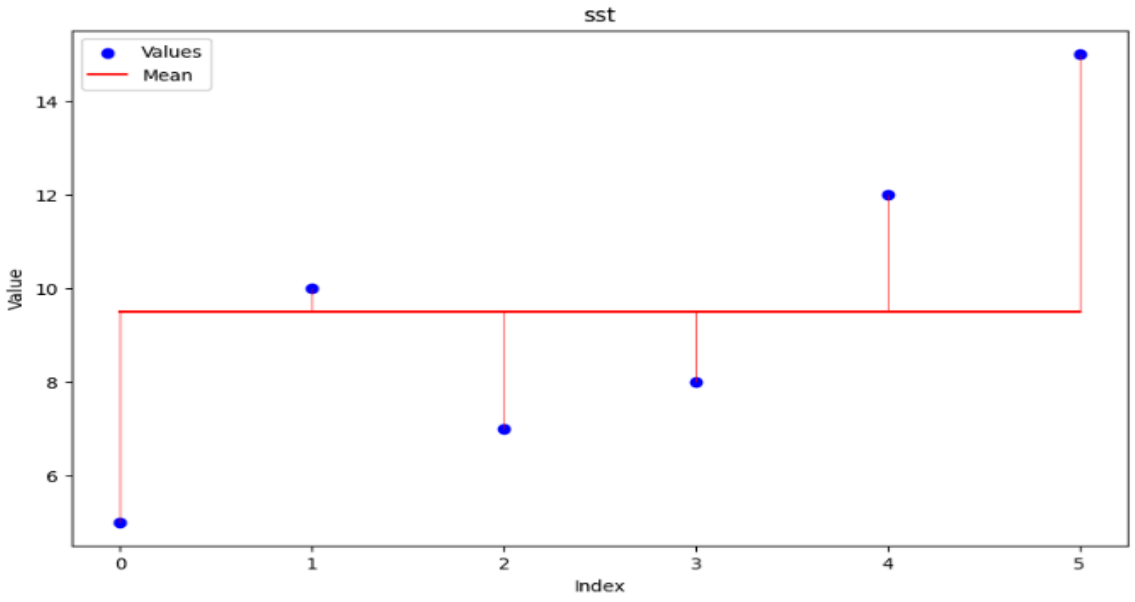
$$R^2 = 1 - \frac{SSE}{SST}$$

$$sse = sum(\left(yi - \hat{yi}\right)^2)$$

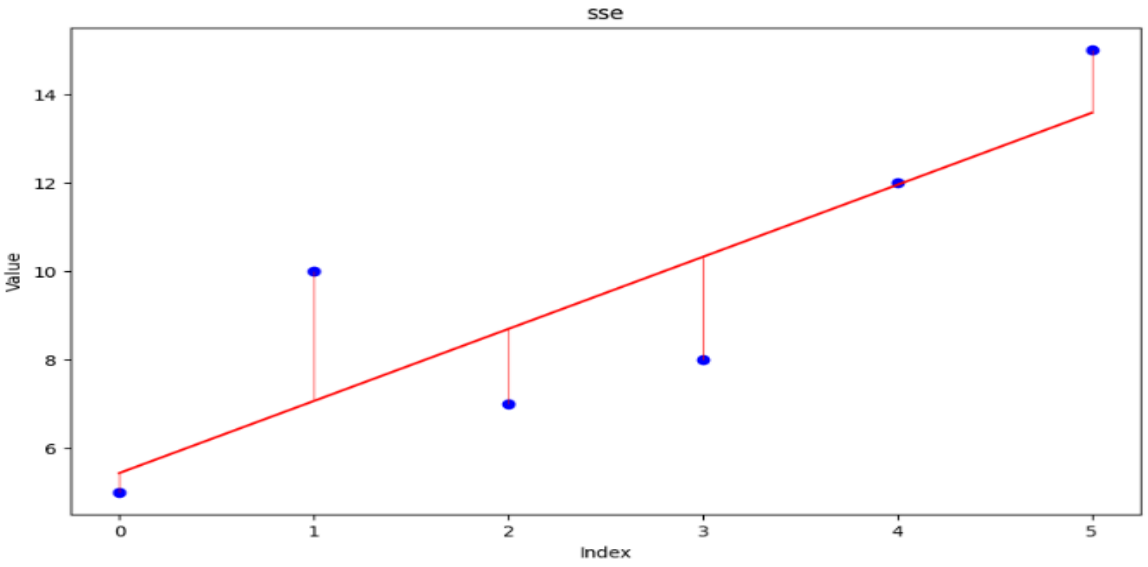
$$sst = sum((yi - \bar{yi})^2)$$

Το sst είναι το άθροισμα της διαφοράς εκάστοτε Y παρατήρησης της εξαρτημένης μεταβλητής σε σχέση με την μέση τιμή της δηλαδή μας δείχνει την διακύμανση της εξαρτημένης μεταβλητής. Ενώ το sse είναι το άθροισμα της τετραγωνικής απόστασης μεταξύ των πραγματικών τιμών του δειγμάτων και τις προβλέψεων του μοντέλου , προφανώς όσο πιο μικρό είναι αυτό το άθροισμα τόσο καλύτερα έχει μοντελοποιησει τα δεδομένα ο αλγόριθμος.

Αρά με απλά λογία ο συντελεστής r^2 δείχνει ποσό καλύτερες είναι οι προβλέψεις του εκάστοτε μοντέλου σε σχέση με το απλούστερο μοντέλο όπου θα μπορούσε να είχε χρησιμοποιηθεί για να τα μοντελοποιησει τον απλό μέσο ορό του δείγματος , δηλαδή μια ευθεία γραμμή στο το ύψος του μέσου ορού της εξαρτημένης μεταβλητής. Το r^2 παίρνει τιμές μεταξύ του μηδέν και του 1 αν το r^2 είναι μεγάλο έστω 90% αυτό σημαίνει ότι οι ερμηνευτικές μεταβλητές του μοντέλου ερμηνεύουν το 90% τις διακύμανσης της εξαρτημένης μεταβλητής.



εικόνα 42: sst



Εικονα 42:sse

Τέλος υπάρχουν και δυο μετρά τα οποία δεν αξιολογούν τις προβλέψεις των μοντέλων έναντι των πραγματικών τιμών αλλά αξιολογούν την πολυπλοκότητα του εκάστοτε μοντέλου σε σχέση με το κατά ποσό αυτό εμπίπτει στα δεδομένα. πρώτο είναι το aic(Akaike information criterion)

$$AIC = 2K - 2LN(l)$$

όπου κ είναι ο αριθμός των παραμέτρων που χρησιμοποιήθηκαν στο μοντέλο δηλαδή στα μοντέλα Arma που θα δομές στην συνέχεια παράμετροι είναι οι συντελεστές του κινητού μέσου ορού και του αυτοπαλινδρομου. Ενώ το l είναι η μέγιστη τιμή της συνάρτησης πιθανοφανείς. Στόχος είναι να βρεθεί εκείνος ο αριθμός κ που μεγιστοποιεί την συνάρτηση πιθανοφανειας και κατά επέκταση κάνει το aic να παίρνει μικρές τιμές αποφεύγοντας με αυτό τον τρόπο το overfitting.

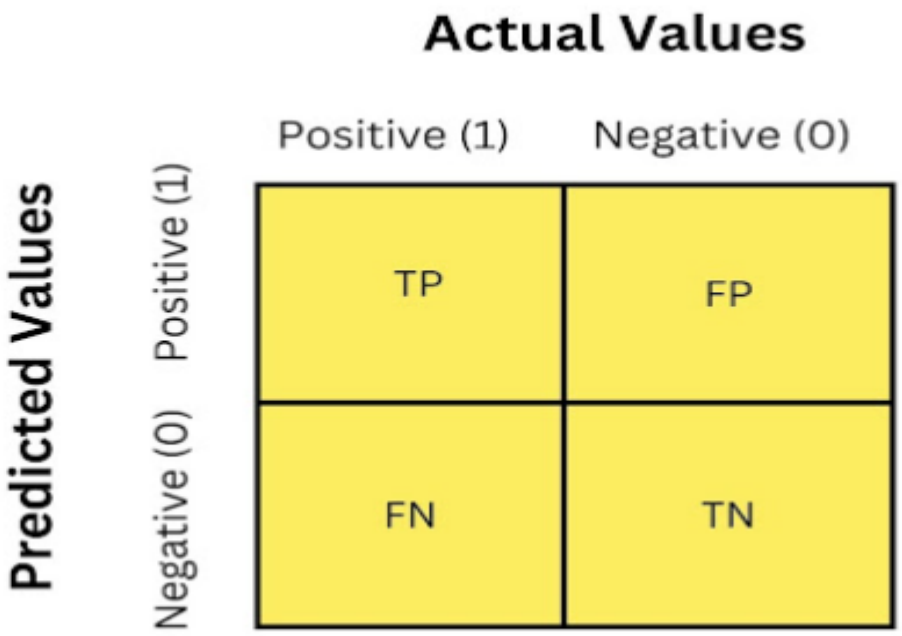
Το δεύτερο μετρό είναι το Bic(Bayesian information criterion) το οποίο διαφοροποιείται σε σχέση με το aic καθώς αντί για να πολλαπλασιάζεται ο αριθμός των παραμέτρων με 2 πολλαπλασιάζεται με τον λογαριθμιζμενο αριθμό παρατηρήσεων του δείγματος με αποτέλεσμα το νούμερο να είναι μεγάλο όταν το μέγεθος του δείγματος είναι εξίσου μεγάλο και κατά συνέπεια να τιμωρούνται περισσότερο τα μοντέλα τα οποία έχουν μεγάλο αριθμό παραμέτρων κ. Το συμπέρασμα από τα παραπάνω είναι ότι το Bic είναι πιο αυστηρό μετρό από το aic όταν το σύνολο των δεδομένων είναι μεγάλο.

$$bic = \ln ln\,(n) \ * \ k \ - \ 2ln(L)$$

Οπού όπως και στο aic το k είναι ο αριθμός των παραμέτρων του μοντέλου το n είναι ο αριθμός των παρατηρήσεων και l είναι η μέγιστη τιμή της συνάρτησης πιθανοφανειας.

2.8 ΜΕΘΟΔΟΙ ΑΞΙΟΛΟΓΙΣΗΣ ΑΛΓΟΡΙΘΜΩΝ ΜΗΧΑΝΙΚΗΣ ΜΑΘΗΣΗΣ ΓΙΑ ΤΑΞΙΝΟΜΙΣΗ

Έχοντας αναλύσει τις μεθόδους αξιολόγησης των αλγορίθμων μηχανικής μάθησης για τις χρονοσειρες συνεχίζουμε σε αυτήν την ενότητα αναλύοντας τον τρόπο αξιολόγησης αυτή την φορά των αλγορίθμων ταξινόμησης. Ένας από τους πιο γνωστούς τρόπους για να επιτευχθεί το παραπάνω είναι μέσο του confusion πίνακα στον οποίο αντιπαρατίθενται οι πραγματικές τιμές του dataset(εξαρτημένες μεταβλητές) τις οποίες έχει διακρατισει ο αναλυτής μέσο του διαχωρισμού μεταξύ train και test set σε σύγκριση με τις τιμές όπου πρόβλεψε το μοντέλο. Πιο αναλυτικά στην πάνω αριστερή πλευρά του πίνακα εμφανίζεται ο αριθμός των παρατηρήσεων όπου το μοντέλο πρόβλεψε σωστά ότι ανήκουν στην κατηγορία 1 (αληθής true positive tp) δηλαδή έστω ένα dataset με διαφορά ερμηνευτικά χαρακτηριστικά κυττάρων όπως το μέγεθος τους το πλάτος το μήκος κτλ. και η εξαρτημένη μεταβλητή είναι το αν αυτό το κύτταρο είναι καρκινικό όπου το 1 συμβολίζει ότι είναι όντος καρκινικό από τον γιατρό και το 0 ότι δεν είναι καρκινικό , ο αριθμός των tp παρατηρήσεων συμβολίζει ότι το μοντέλο πρόβλεψε σωστά ότι ένας χ αριθμός ατόμων έχει καρκίνο και στην πραγματικότητα όντως έχει καρκίνο με βάση την γνωμάτευση του γιατρού. Στην πάνω δεξιά πλευρά φαίνεται ο αριθμός των παρατηρήσεων όπου το μοντέλο πρόβλεψε λάθος ότι ανήκουν στην κατηγορία 1 ενώ στην πραγματικότητα ανήκουν στην κατηγορία 0 δηλαδή στο παράδειγμα μας το μοντέλο πρόβλεψε ότι τα άτομα έχουν καρκίνο (1) ενώ στην πραγματικότητα δεν έχουν καρκίνο(0) (fp false positive). Στην κάτω αριστερή πλευρά ο αναλυτής βλέπει τον αριθμό των παρατηρήσεων όπου το μοντέλο πρόβλεψε ότι ανήκουν στην κατηγορία 0 ενώ στην πραγματικότητα ανήκουν στην κατηγορία 1(fn false negative) δηλαδή στο παράδειγμα μας είναι ο αριθμός των ατόμων που ο αλγόριθμος θεώρησε ότι δεν έχουν καρκίνο ενώ στην πραγματικότητα έχουν. Και τέλος στην κάτω δεξιά θέση βρίσκονται οι παρατηρήσεις όπου ο αλγόριθμος ταξινόμησε σωστά ότι ανήκουν στην κατηγορία 0 και όντως ανήκουν σε αυτή (true negative TN) δηλαδή στο παράδειγμα είναι τα άτομα που το μοντέλο τοποθέτησε στην κατηγορία ότι δεν έχουν καρκίνο και στην πραγματικότητα δεν έχουν καρκίνο.



Εικονα 43 : confusion matrix

Προβλεπομενες/ πραγματικες	Καρκινικά κύτταρα	Μη καρκινικά κύτταρα
Καρκινικά κύτταρα	40	20
Μη καρκινικά κύτταρα	10	30

Στο παραπάνω confusion πίνακα παρουσιάζεται ένα αντίστοιχο παράδειγμα όπου το μοντέλο ταξινόμησε σωστά ότι 40 άτομα έχουν καρκίνο και όντως έχουν καρκίνο ,επίσης σωστά ταοϊσμέ ότι 30 άτομα δεν έχουν καρκίνο και όντως δεν είχαν ενώ έκανε λάθους σε 20 παρατηρήσεις όπου θεώρησε ότι τα άτομα είχαν καρκίνο ενώ δεν είχαν στην πραγματικότητα και 10 παρατηρήσεις όπου τα άτομα είχαν καρκίνο και ο αλγόριθμος θεώρησε / τα ταξινόμησε στην κατηγορία των μη καρκινικών.

Το πρώτο μετρό αξιολόγησης αλγορίθμων ταξινόμησης που θα αναλύσουμε με βάση τον confusion matrix είναι το accuracy το οποίο δείχνει ποσό καλά προβλέπει το μοντέλο τις παρατηρήσεις συνολικά καθώς είναι μια διαίρεση των σωστών προβλέψεων σε σύγκριση με όλες τις παρατηρήσεις σωστά ή λάθος ταξινομημένες.

$$accuracy = \frac{tp+tn}{tp+tn+fp+fn}$$

Στην συνέχεια το precision είναι ένα μετρό που δείχνει ποσό αξιόπιστες είναι οι θετικές προβλέψεις ενός αλγορίθμου δηλαδή συγκρίνει τις σωστά ταξινομημένες θετικές τιμές σε σχέση με όλες τις θετικές τιμές που πρόβλεψε το μοντέλο είτε ήταν σωστά ταξινομημένες είτε λανθασμένες , αυτό το μετρό είναι σημαντικό αν θέλει ο αναλυτής να εστιάσει στην μείωση των λανθασμένων θετικών τιμών για παράδειγμα σε ταξινομήσεις ασθενειών δεν υπάρχει τόσο μεγάλο πρόβλημα όταν ο ασθενής είναι καλά στην υγειά του αλλά το μοντέλο τον ταξινόμησε ως άρρωστο (fn) γιατί αυτός κάθε αυτός ο ασθενής είναι καλά στην υγειά του, αλλά το κακό είναι όταν ο ασθενής είναι όντως άρρωστος και το μοντέλο τον ταξινομεί ως υγιή(fp).

$$precision = \frac{tp}{tp+fp}$$

Το ακριβός αντίθετο ελέγχει το recall δηλαδή εστιάζει στα λάθη (fn) δηλαδή στα λάθη όπου κάποιος ασθενής είναι υγιής και το μοντέλο τον ταξινομεί ως άρρωστο, αυτό είναι πιο σημαντικό μετρό για τράπεζες που δίνουν δάνεια γιατί οι δανειολήπτες που κανονικά θα έπρεπε να πάρουν δάνειο το μοντέλο τους απορρίπτει και αρά η τράπεζα χάνει πελάτες. Ή για παράδειγμα σε ένα ηλεκτρονικό κατάστημα λιανικών πωλήσεων το μοντέλο προβλέπει ότι οι πελάτες δεν θα αγοράσουν ένα προϊόν και αρά το κατάστημα δεν κάνει νέες παραγγελίες για αυτό στους προμηθευτές του ενώ στην πραγματικότητα η ζήτηση για το προϊόν είναι μεγάλη με αποτέλεσμα η επιχείρηση να χάνει πελάτες , ή να υπάρχει μεγάλη αναμονή για την ολοκλήρωση της παραγγελίας του πελάτη λόγο έλλειψης αποθέματος.

$$recall = \frac{tp}{tp+fn}$$

Και τέλος ο συνδυασμός αυτών των δυο είναι το f1 όπου δίνει ένα μέσο ορό των δυο προηγουμένων μέτρων δηλαδή ελέγχει την συνολική αξιοπιστία του μοντέλου.

$$f1 = 2 * \frac{precision*recall}{precision+recall}$$

3. ΒΑΣΙΚΕΣ ΒΙΒΛΙΟΘΗΚΕΣ ΣΤΗΝ PYTHON

Σε αυτό το κεφάλαιο θα αναλύσουμε εν συντομία μερικές από τις ποιο διαδεδομένες και χρήσιμες στην ανάλυση δεδομένων βιβλιοθήκες της python.

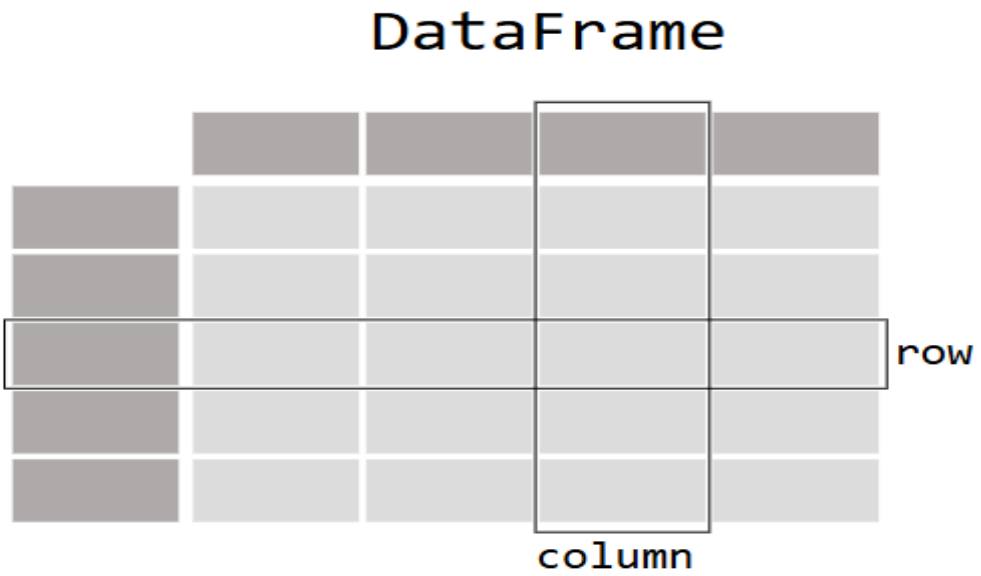
3.1 NUMPY

Η NumPy είναι μια βιβλιοθήκη ανοικτού κώδικα της γλώσσας προγραμματισμού python της οποίας η ανάπτυξη ξεκίνησε το 2005 από τον ερευνητή Travis oliphant ως επέκταση των βιβλιοθηκών numeric και numarray, από τότε η NumPy εξελίχθηκε σε βασικό εργαλείο όλων των στατιστικών ερευνητών ανεξάρτητος τομέα και αποτέλεσε η βάση για την ανάπτυξη βιβλιοθηκών όπως η pandas και η sklearn. Πιο συγκεκριμένα η NumPy χρησιμοποιείται στην μηχανική και τεχνητή μάθησή για την διαχείριση δεδομένων και την διεξαγωγή μαθηματικών πράξεων.Αυτό την καταστεί χρήσιμη σε πολλούς τομείς της οικονομίας όπως για παράδειγμα τα χρηματοοικονομικά ,το μάρκετινγκ ,την ιατρική κλπ. Μια από τις βασικές λειτουργείες της NumPy είναι η διαχείριση και αποθήκευση των δεδομένων που εισάγει ο αναλυτής σε arrays αντί για τις απλές λίστες. Πιο συγκεκριμένα μέσο της NumPy μπορούν να δημιουργηθούν και να αποθηκευτούν τα δεδομένα σε πίνακες για παράδειγμα μέσο της εντολής np.zeros() μπορεί να δημιουργηθεί ένας μηδενικός πίνακας διάφορων διαστάσεων με την εντολή np.eye() δημιουργείται ένας διαγώνιος πίνακας αντίστοιχα. Προφανώς ο αναλυτής στην συνέχεια μπορεί να αλλάξει τα δεδομένα μέσα στο πίνακα αλλά και να διαγράψει και να προσθέσει στήλες και γραμμές , όπως επίσης και να συνενώσει πίνακες μεταξύ τους και να διαχειριστεί τυχόν κενές (null) τιμές. Επίσης εφόσον ο αναλυτής έχει δημιουργήσει τους πίνακες μπορεί να κάνει διάφορες πράξεις με αυτούς όπως για παράδειγμα να τους προσθέσει ή να τους αφαιρέσει , να τους πολλαπλασιάσει (A*3) με μια τιμή ή μεταξύ τους (A*B) , να υπολογίσει το ανάστροφο και τον αντίστροφο πίνακα και πολλές άλλες. Επιπλέον μέσο της NumPy ο χριστής μπορεί να εκτελέσει και στατιστικές πράξεις όπως την εύρεση της ελάχιστης και μέγιστης τιμής μέσα σε ένα array ,τον υπολογισμό της μέσης τιμής (np.mean()) ,της διάμεσου (np.median()) ,του mode της διακύμανσης (np.var()) ,της δειγματικής και του πληθυσμού της τυπικής απόκλισης (np.std()) τα τεταρτημόρια (np.percentile()) την συν διακύμανση(np.cov()) , το συντελεστή συσχέτισης (np.corrcoef()) των δεδομένων και λογική συνέχεια τον παραπάνω είναι ο συνδυασμός τους για την πραγματοποίηση ποιο συνθέτων πράξεων όπως για παράδειγμα η κανονοικοποιηση και η τυποποίηση των

δεδομένων ,και μέσο αυτής δημιουργούνται πιο συνθέτες συναρτήσεις όπως ο minmaxscaler ο οποίος τυποποιεί τα δεδομένα πριν την εισαγωγή του στα νευρωνικα δίκτυα. Τέλος η NumPy είναι μια από τις πιο σημαντικές βιβλιοθήκες για την ανάπτυξη μοντέλων όπως το monte Carlo simulation καθώς μέσο αυτής μπορούν να δημιουργηθούν arrays με τυχαίους αριθμούς από διάφορες κατανομές (δηλαδή λειτουργεί σαν γεννήτρια τυχαίων αριθμών) όπως για παράδειγμα από την κανονική κατανομή random.normal() ,την ομοιόμορφη (random.uniform()) την δειονυμηκη ,την poisson και άλλες όπως την shuffle και choice όπου η βιβλιοθήκη επιλεγεί στην τύχη αριθμούς από ένα array μπορούν να χρησιμοποιηθούν για την ανάπτυξη τυχερών παιχνιδιών και simulations.

3.2 PANDAS / MATPLOTLIB / SEABORN

Η pandas είναι μια από τις πιο δημοφιλείς βιβλιοθήκες στην python καθώς χρησιμοποιείται για την διαχείριση και ανάλυση των δεδομένων. Η βιβλιοθήκη αναπτύχθηκε το 2008 από τον προγραμματιστή wes McKinney ο οποίος εργαζόταν στον χρηματοοικονομικό κλάδο εκείνη την περίοδο. Η pandas μπορεί να διαχειριστεί κύριος δυο μορφές δεδομένων τις σειρές οι οποίες είναι μονοδιάστατοι πίνακες με μια σειρά από τιμές όπου σε κάθε μια αντιστοιχεί ένας δείκτης (index) , η δεύτερη μορφή δεδομένων είναι τα datadrames τα οποία μοιάζουν με τα υπολογιστικά φυλά του excel δηλαδή τα δεδομένα αποτελούνται από γραμμές και στήλες με το αντίστοιχα χαρακτηριστικά τους. Θα μπορούσε να πει κάνεις ότι κάθε μια στειλη ενός data frame είναι μια σειρά (series).



Εικονα 44: dataframe

Εικονα 45 :series

Μερικές από τις βασικές εντολές που μπορεί να εκτελέσει ο αναλυτής σε μια σειρά είναι η ταξινόμηση τους ,να βρεθεί ο μεγαλύτερος , ο μικρότερος αριθμός να αλλαχθεί ένα στοιχείο της σειράς με ένα άλλο κλπ. Αντίστοιχα σε ένα dataframe υπάρχουν εντολές που επιστέφουν το αριθμό ή το όνομα τον στειλεών , τις πρώτες(df.head()) ή τις τελευταίες γραμμές (df.tail()), το μέγεθος του dataframe (df.shape()) , ή ακόμα και στατιστικά όπως την τυπική απόκλιση της κάθε στήλης με αριθμητικές τιμές μέσο της εντολής describe. Ένας βασικός λόγος που η pandas είναι πλέον πολύ διαδεδομένη βιβλιοθήκη για την δίχειρης δεδομένων στην python είναι διότι μέσο αυτής μπορούν να εισαχθούν και να εξαχθούν τα δεδομένα από διάφορες μορφές αρχείων από excel μέσο της εντολής pd.read_excel() μέχρι βάσεων δεδομένων με την εντολή pd.read_sql() .

Εικονα 46: εισαγωγή και εξαγωγή data frame με χρήση pandas

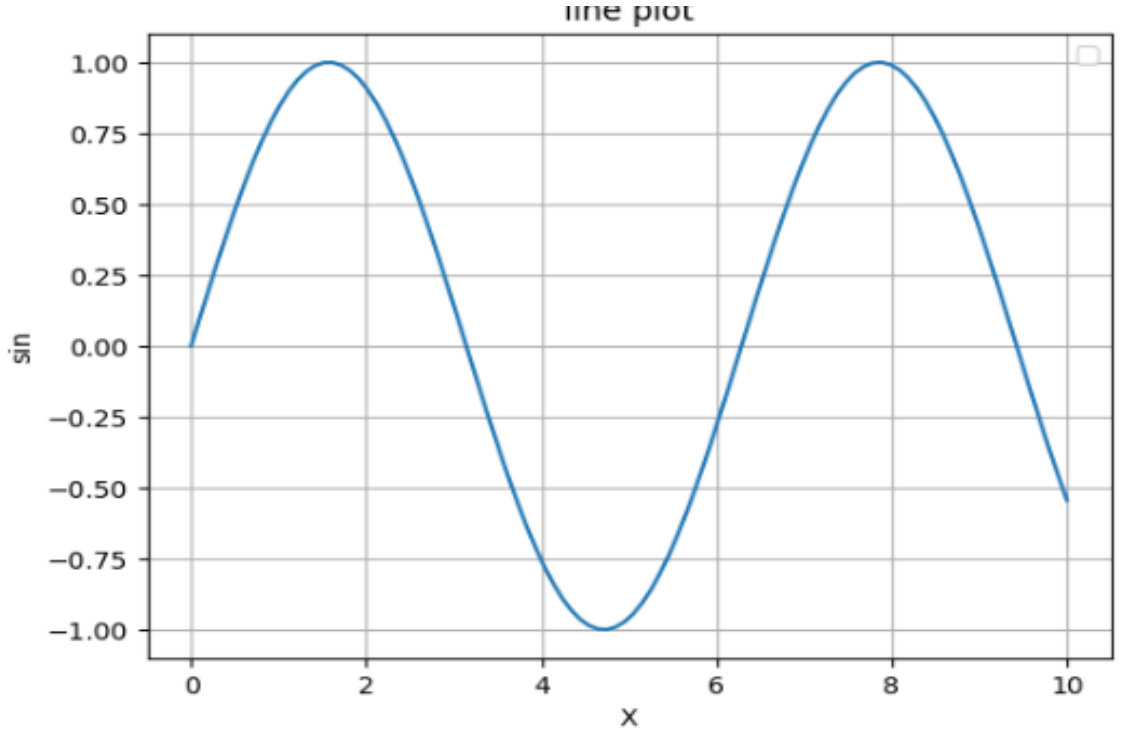
Στην συνέχεια εφόσον ο αναλυτής έχει εισάγει τα δεδομένα του σε ένα dataframe η pandas μέσα από απλές εντολές μπορεί να τον βοηθήσει ώστε να ανιχνεύσει και να διαχειριστεί κενές τιμές μέσο πχ της εντολή df.isnull().sum() όπου επιστρέφει το πλήθος των κενών

τιμών ανά στήλη ή μέσο της εντολής df.info() όπου εμφανίζει πια κελιά έχουν κενές τιμές , εν συνέχεια ο αναλυτής μπορεί να διαγράψει τελείως τις γραμμές ή τις στήλες που περιέχουν κενές τιμές μέσο της εντολής drop ή μπορεί να τις συμπληρώσει με την εντολή fiillna είτε με τον μέσο ορό των τιμών της στήλης είτε με την προηγουμένη ή την επόμενη τιμή της στήλης. Με τον ίδιο τρόπο μπορεί ο αναλυτής να βρει διπλότυπες τιμές df.duplicated().sum() και να τις αφαιρέσει από το dataframe. Προφανώς υπάρχουν και άλλες εντολές που βοηθούν στην επεξεργασία και ανάλυση των δεδομένων όπως για παράδειγμα η μετατροπή categorical δεδομένων σε ψευδομεταβλητες (0 και 1) pd.ged_dummies() , επίσης όπως και με την NumPy έτσι και η pandas μπορεί να χρησιμοποιηθεί για την κανονικοποιηση και τυποποίηση των δεδομένων. Επιπλέον όπως και το excel έχει τα pivot tables και η sql έχει τα queries με εντολές όπως το where και το group by έτσι και η pandas έχει τις αντίστοιχες μεθόδους για την αναπαράσταση των δεδομένων δηλαδή για την δημιουργία υποσυνόλων των δεδομένων όπου στην συνέχεια μπορούν να γίνουν πράξεις με αυτά , πολλά παραδείγματα αυτών γίνονται στο κεφάλαιο 6 και 8 της παρούσης εργασίας επίσης η pandas χρησιμοποιείται για την αρχική προεργασία των δεδομένων σε μορφή χρονοσειρων καθώς μπορεί να μετατρέψει στο κατάλληλο format τις ημερομηνίες και γενικότερα τους δείκτες του κάθε timestamp με εντολές όπως pd.to_datetime().

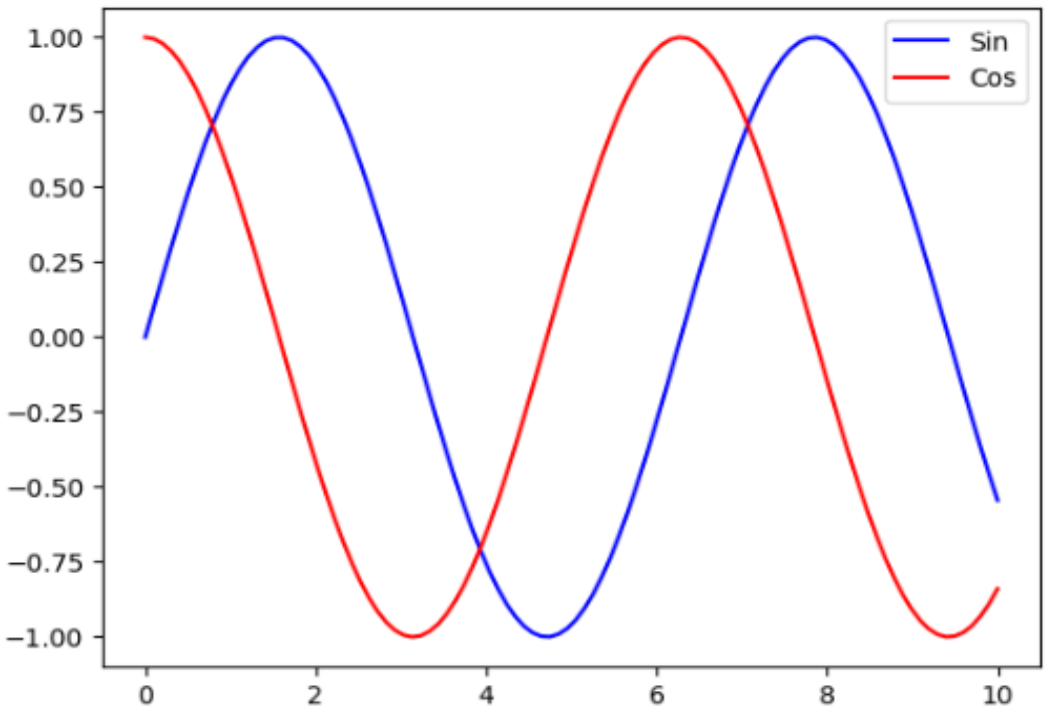
Εικονα 47 : μετατροπή datafrme

Τέλος σε συνδυασμό με την matplotlib και την seaborn μέσο της pandas μπορούν τα δεδομένα να οπτικοποιηθουν μέσο διαγραμμάτων όπως απλές γραμμικές παραστάσεις με την εντολή plot(kind=’line’) , ραβδογραμματα συχνοτήτων με συνδυασμό της εντολής value_counts και της plot(kind=’bar’) , ιστογράμματα plot(kind=’hist’) ,πίτες plot(kind=’pie’) κλπ.

Αναφορικά με την matplotlib είναι η πιο γνωστή βιβλιοθήκη της python για την δημιουργία γραφικών παραστάσεων , για όσους γνωρίζουν την γλώσσα προγραμματίσου R η matplotlib είναι η αντίστοιχη ggplot2 βιβλιοθήκη της python βέβαια δεν είναι τόσο εύκολη στην χρήση όπως η δεύτερη παρόλα αυτά μέσο αυτής αναπτυχθήκαν βιβλιοθήκες όπως η seaborn όπου θα αναλύσουμε στην συνέχεια που κάνουν την δημιουργία διαγραμμάτων πολύ πιο εύκολη για τον χρήστη. Αναφορικά με την matplotlib αναπτύχθηκε από τον john D. Hunter το 2003 με στόχο την δημιουργία ενός περιβάλλοντος θα έχει τις ιδίες δυνατότητες με την γλώσσα προγραμματίσου MATLAB, η βιβλιοθήκη είναι ανοιχτού κώδικα και καθημερινά πόλοι χριστές την παραμετροποιουν προσθέτοντας νέες λειτουργίες. Μέσο της matplotlib ο χρήστης μπορεί να δημιουργήσει απλά και πιο σύνθετα γραμμικά διαγράμματα με την εντολή plt.plot() όπως τα παρακατω.

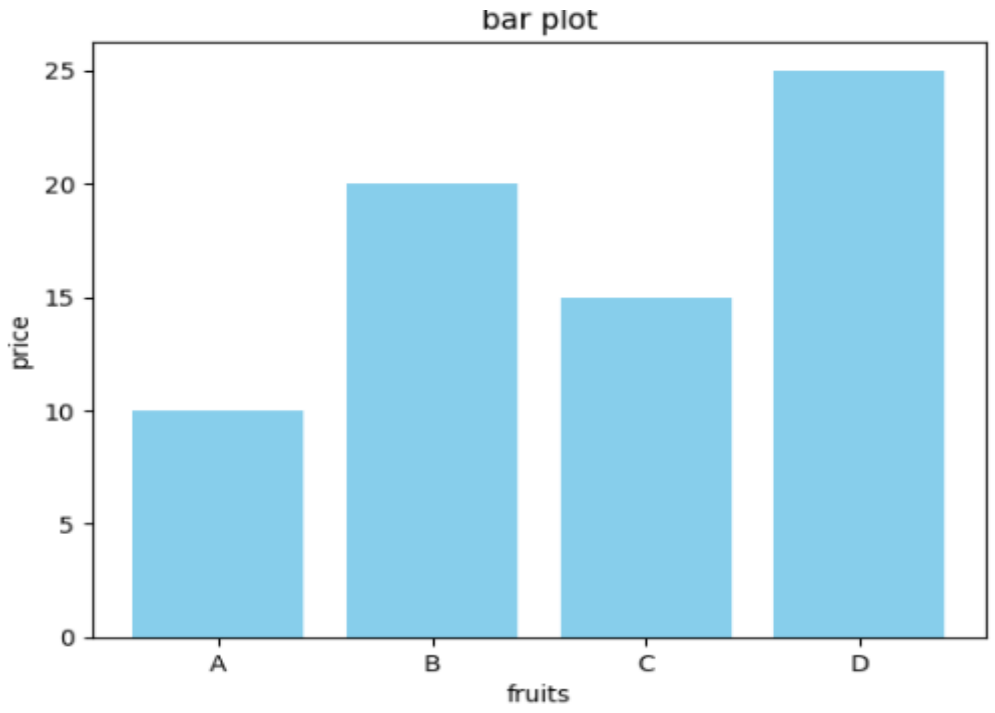


Εικονα 48: line plot

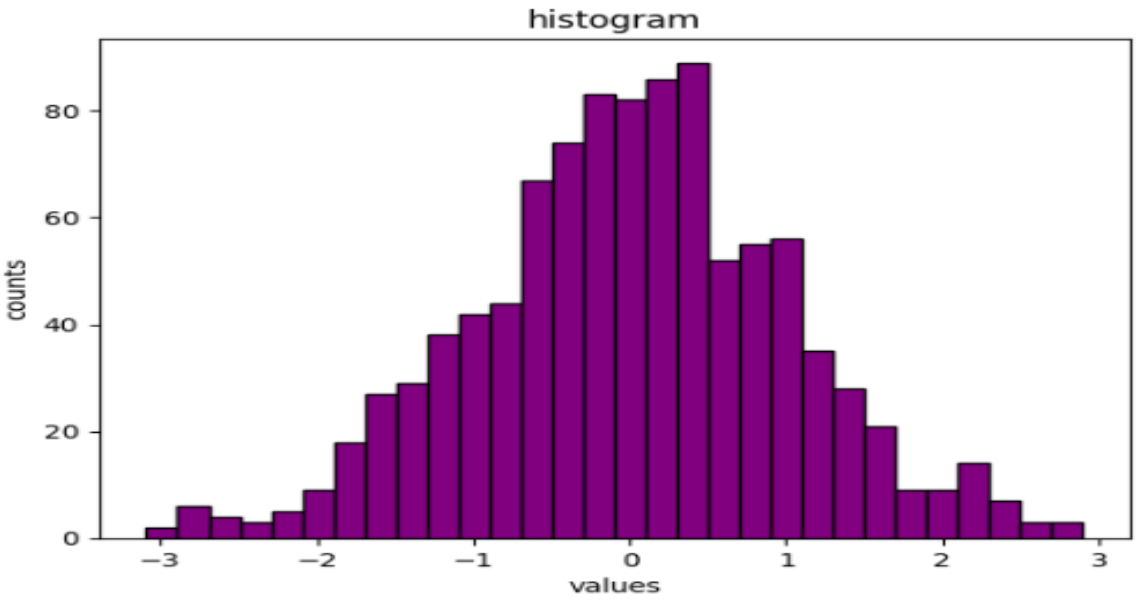


Εικονα 49: line plot 2

Μπορει επίσης να δημιουργήσει ραβδογραμματα με την εντολή plt.bar() και ιστογράμματα με την εντολή plt.hist().

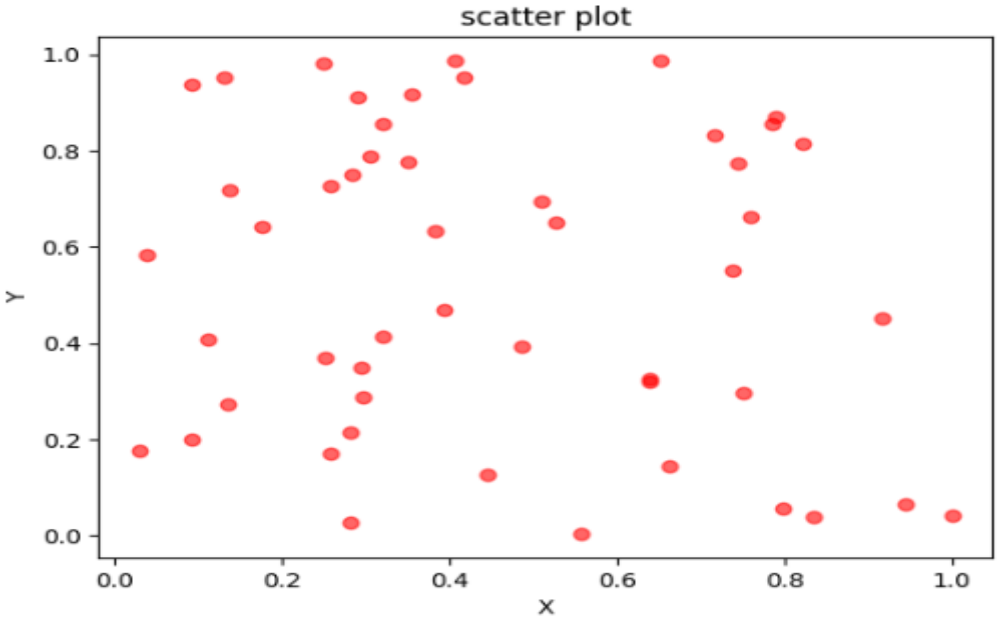


Εικονα 50: bar plot

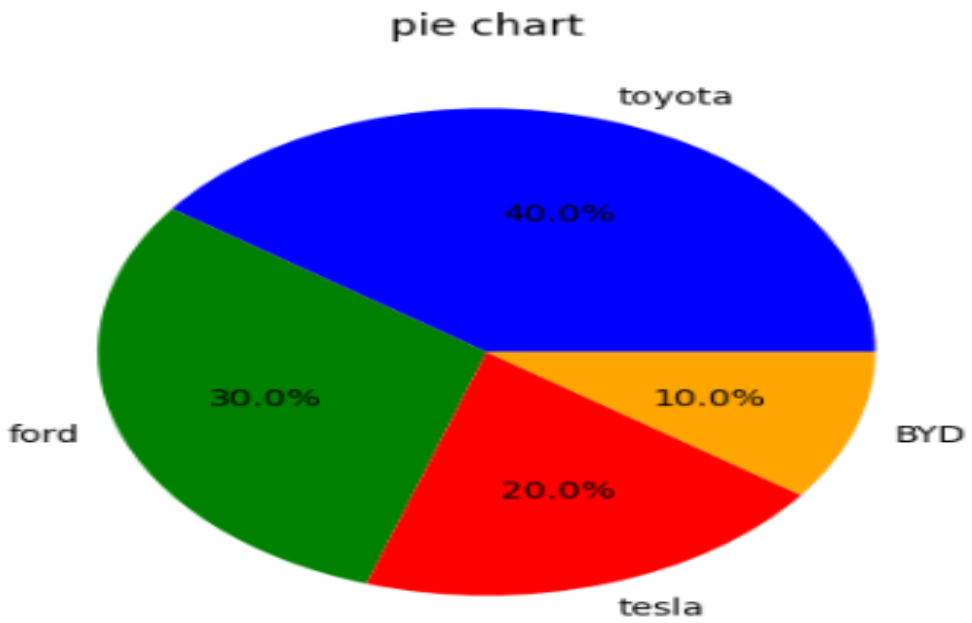


Εικονα 51: ιστόγραμμα

όπως και στο excel έτσι και μέσο της matplotlib μπορούν να δημιουργηθούν διαγράμματα διασποράς μεταξύ δυο τιμών με την εντολή plt.scatter() και διαγράμματα τύπου πίτας με την εντολή plt.pie().

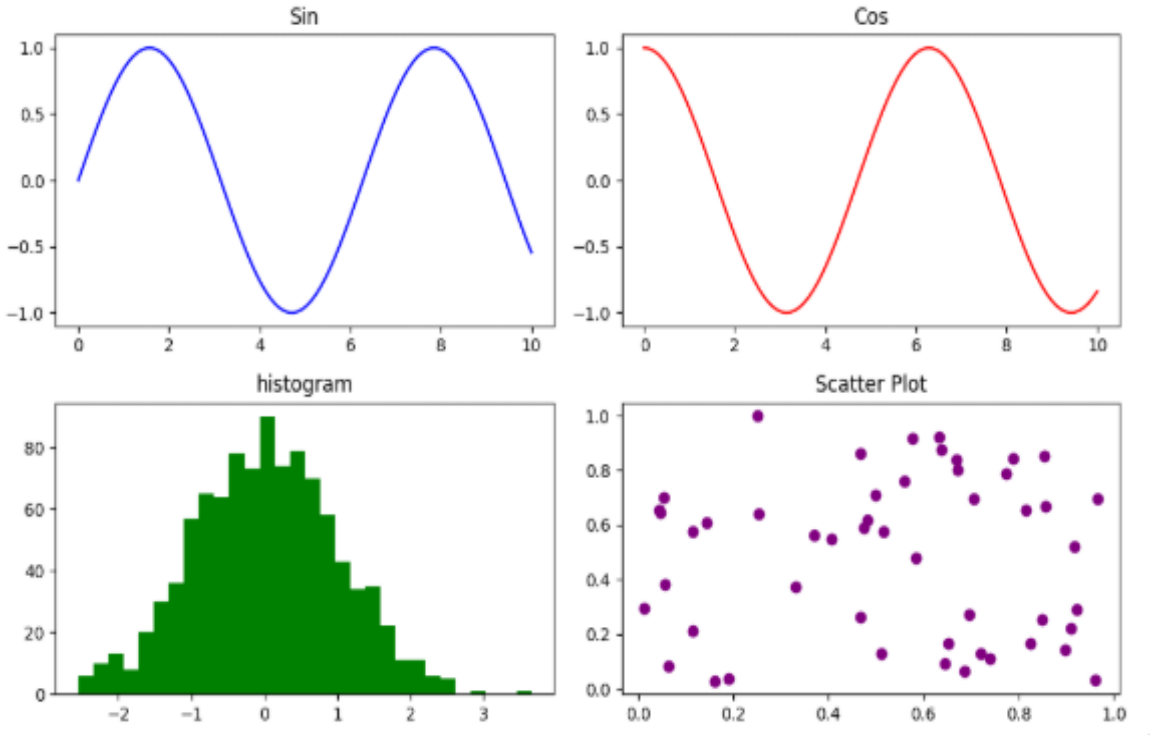


Εικονα 52: scatter plot



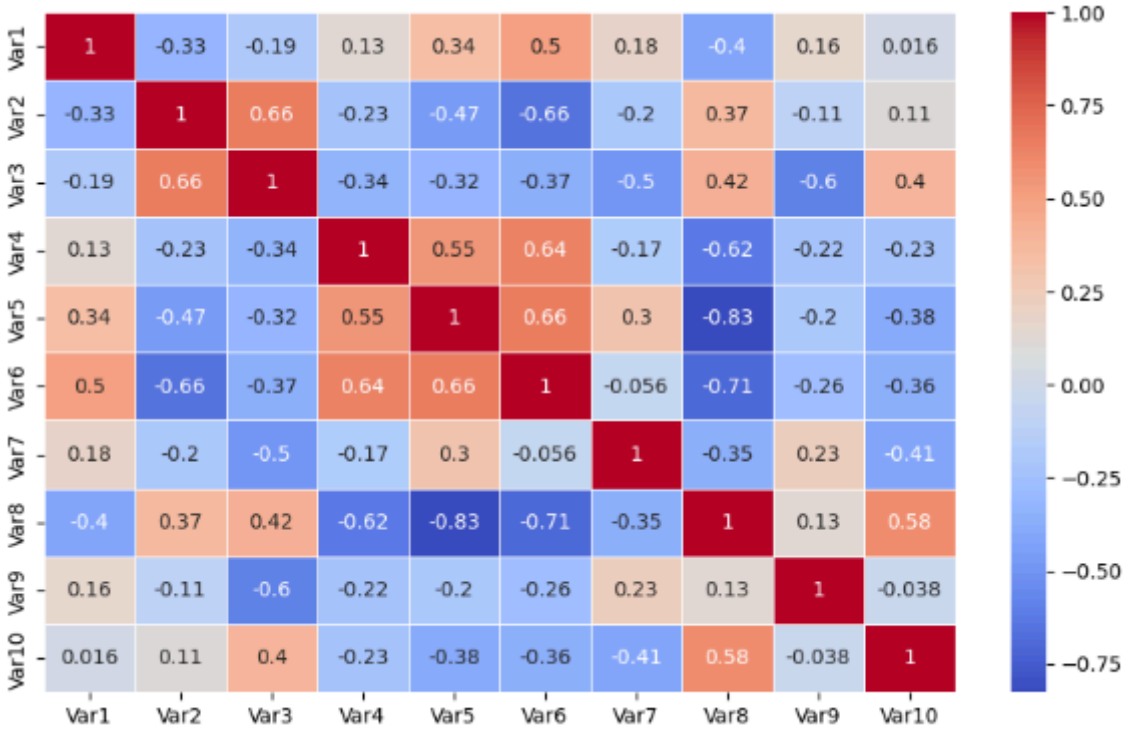
Εικονα 53: pie plot

Προφανώς σε όλα τα παραπάνω διαγράμματα η matplotlib δίνει την δυνατότητα στον χρήστη να αλλάξει τα χρώματα το μέγεθος του διαγράμματος τον τίτλο του την ονομασία των αξόνων , των δεδομένων κλπ. επίσης δίνει την δυνατότητα δημιουργίας και προβολής πολλών διαγραμμάτων ταυτόχρονα μέσο της εντολής plt.subplots()

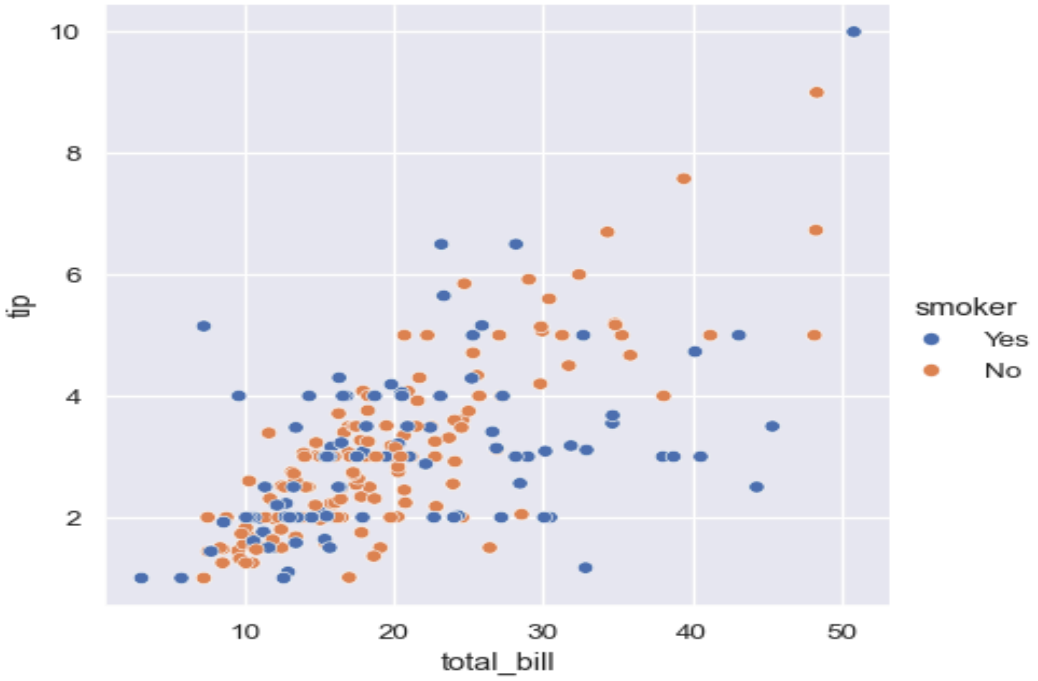


Εικονα 54: subplot

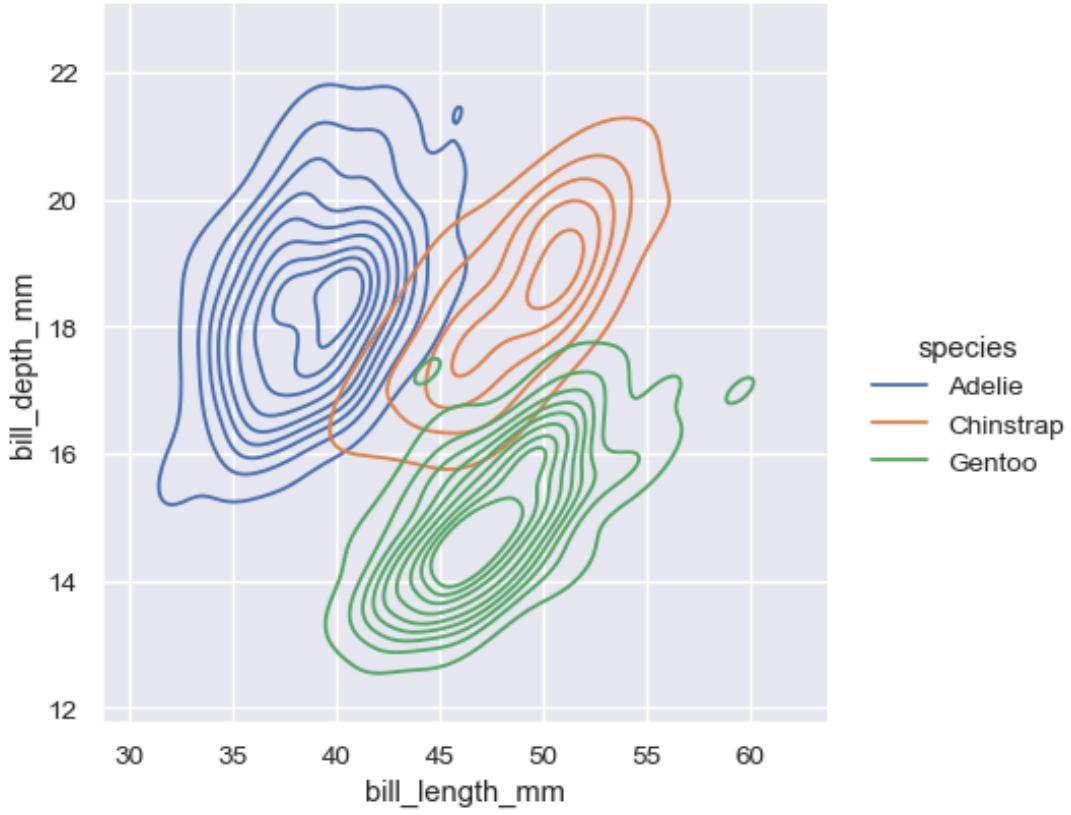
Η επόμενη βιβλιοθήκη που θα αναλύσουμε είναι η seaborn η οποία αναπτύχθηκε από τον Michael Waskom το 2012 με σκοπό να διευκολύνει τους χρήστες της python με μια πιο εύχρηστη και εστιασμένη στα στατιστικά γραφήματα και στην ανάλυση δεδομένων εκδοχή της matplotlib. Η seaborn χρησιμοποιείται σε διάφορους κλάδους της οικονομίας για την ανάλυση και παρουσίαση δεδομένων καθώς μέσο εκείνης μπορούν να δημιουργηθούν εκτός από τα απλά διαγράμματα που είδαμε παραπάνω πιο σύνθετα και πολύπλοκα όπως για παράδειγμα heatmaps που δείχνουν την συσχέτιση μεταξύ των μεταβλητών που έχει εισάγει ο χρήστης με την εντολή sns.heatmap(), η συσχέτιση μπορεί επίσης να αναδειχθεί μέσο διαγραμμάτων διασποράς όπου έχει τεθεί hue δηλαδή γίνεται συσχέτιση μεταξύ δυο αριθμητικών μεταβλητών και ταυτόχρονα μιας categorical μεταβλητής όπως στο παρακατω διάγραμμα από την ιστοσελίδα του seaborn όπου συσχετίζονται οι μεταβλητές total bill (δηλαδή το σύνολο του λογαριασμού σε ένα εστιατόριο) με την μεταβλητή tip(φιλοδώρημα) που άφησε ο πελάτης σε συνδυασμό με την categorical μεταβλητή του αν ο πελάτης είναι καπνιστής ή όχι. Η εντολή δημιουργίας του διαγράμματος είναι η sns.relplot() , επιτρέπει μέσο της εντολής sns.displot() την δημιουργία δίγραμμων από δεδομένα που ανήκουν σε διαφορετικές κατανομές για την οπτικοποιηση της κατανομής πιθανότητας τους, επιπλέον μέσο της εντολής sns.regplot() μπορεί ο αναλυτής να δημιουργήσει scatter plot με την αντίστοιχη γραμμή παλινδρόμησης των δεδομένων.



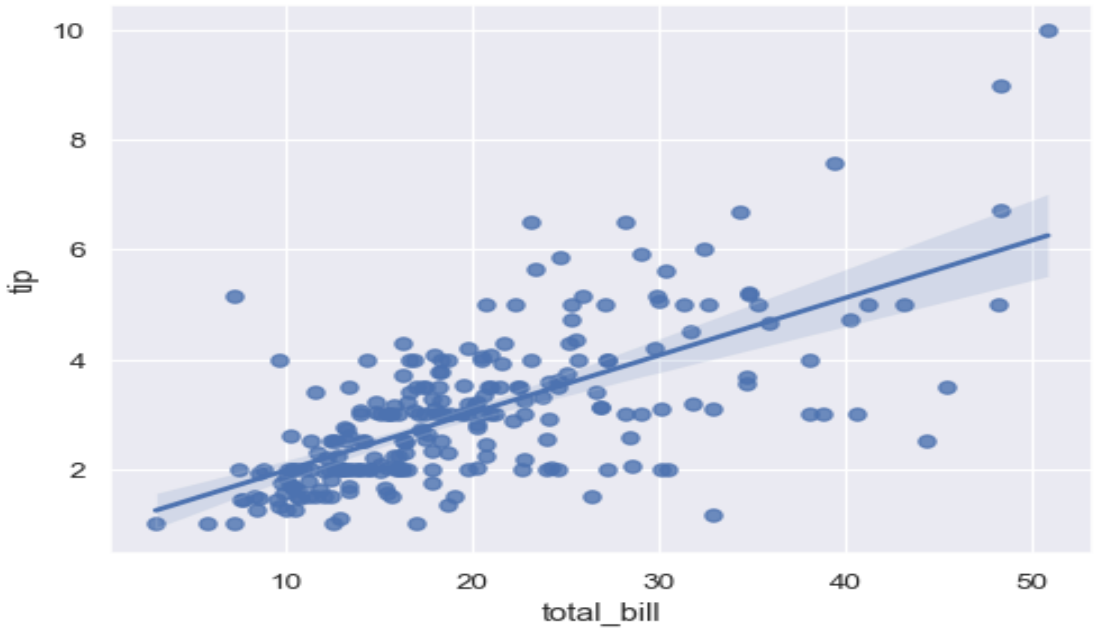
Εικονα 55: heatmap



Εικονα 56: scatter plot με hue

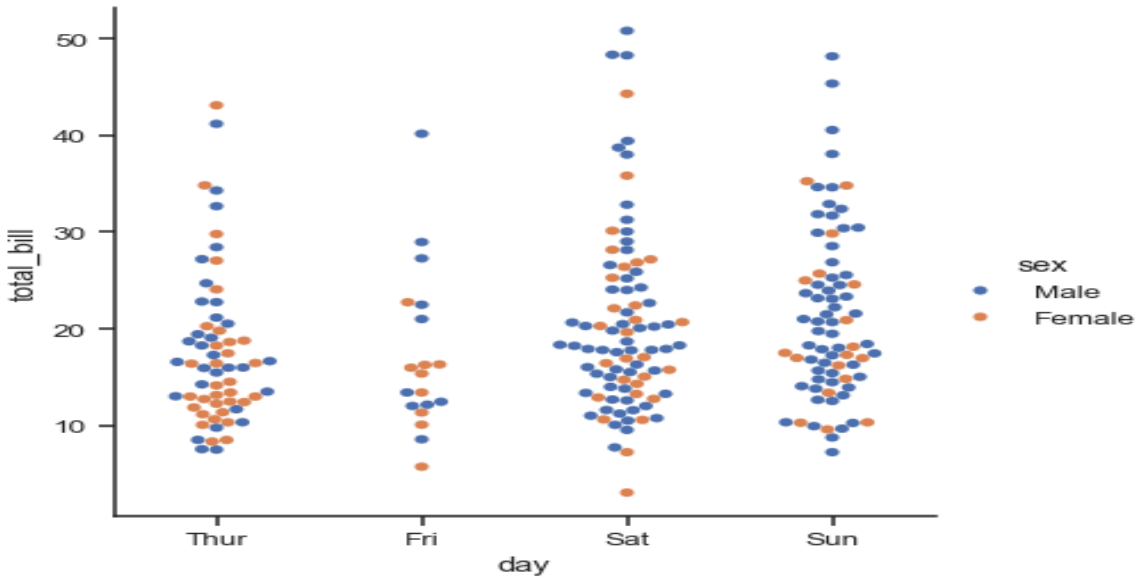


Εικονα 57: rel plot

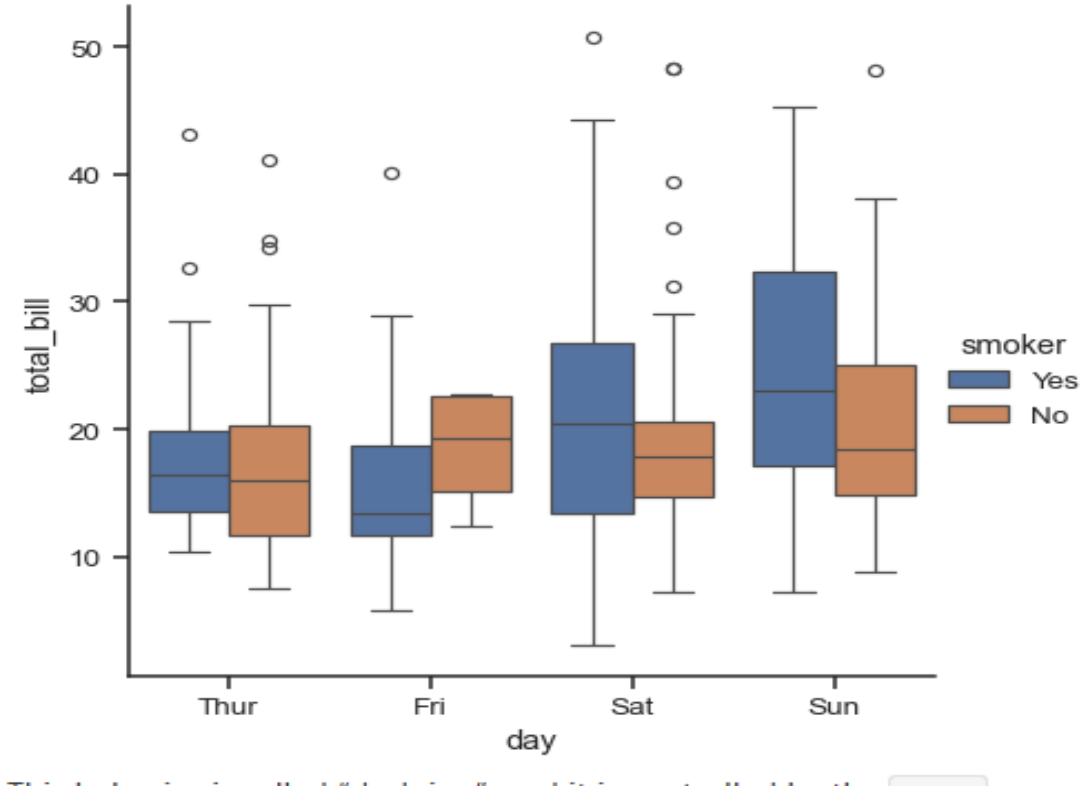


Εικονα 58: regression plot

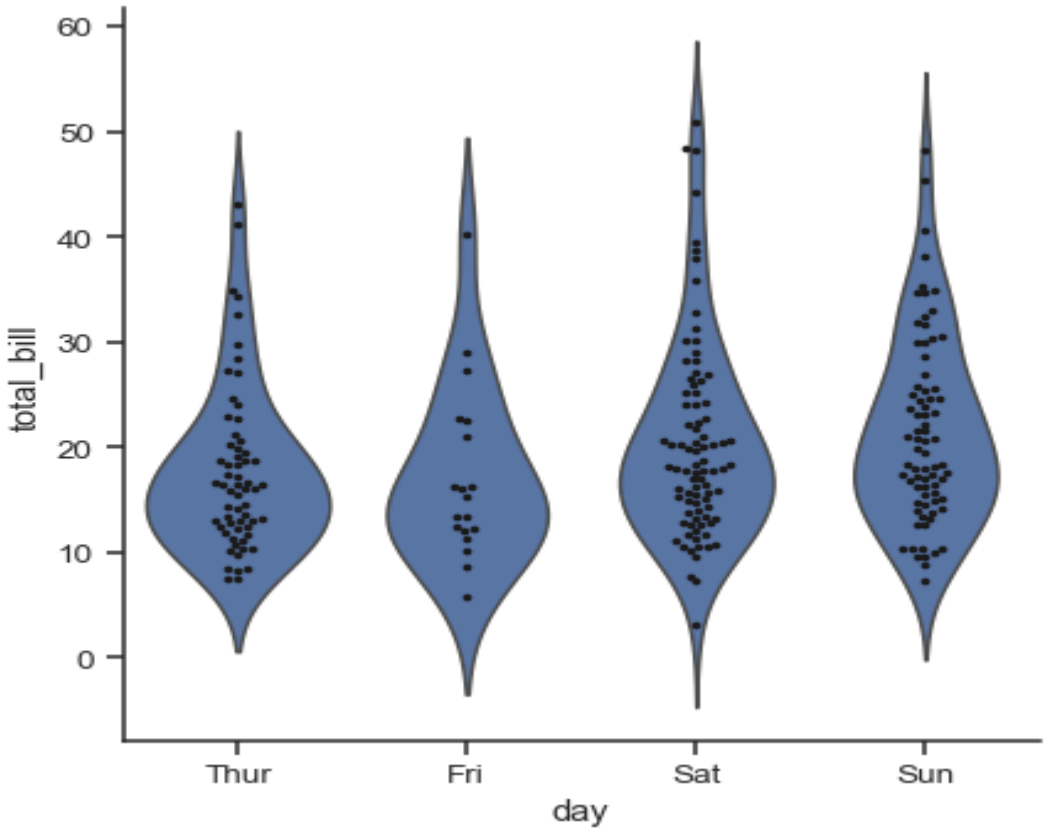
Αναφορικά με τα categorical δεδομένα μπορούν να δημιουργηθούν για αυτά μέσο της βιβλιοθήκης διαγράμματα όπως categorical scatter plots με την εντολή sns.catplot() όπου μέσο μιας μικρής παραλλαγής της εντολής μπορούν επίσης να δημιουργηθούν box plots τοποθετώντας μέσα στην εντολή την παράμετρο kind=’box’ όπου η κεντρική τιμή δείχνει την διάμεσο , οι γραμμές που εξέχουν από τα κουτάκια δείχνουν την κατανομή των δεδομένων ενώ οι κουκίδες τις ακραίες τιμές αυτών. Μια παραλλαγή των box plot είναι τα violin plots που δημιουργούνται με την εντολή sns.violinplot() όπου σε εκείνα εκεί όπου το φύλο είναι αρκετά πλατύ σημαίνει ότι υπάρχουν πολλά δεδομένα με την ιδιά τιμή ενώ τα σημεία του φύλου που είναι πιο λεπτά σηματοδοτούν ότι δεν υπάρχουν πολλά δεδομένα με την ιδιά τιμή.



Εικονα 58: catplot



Εικονα 59: box plot



Εικονα 60: violin plot

3.3 STATSMODELS / SKLEARN / TENSORFLOW

Σε αυτήν την ενότητα θα εστιάσουμε στις πιο γνωστές βιβλιοθήκες της python για την ανάπτυξη στατιστική μοντέλων, αλγορίθμων μηχανικής μάθησης και νευρωνικων δικτυών. Πιο συγκεκριμένα η βιβλιοθήκη η οποία εστιάζει στην στατιστική μοντελοποιηση ενός dataset είναι η statsmodels με δημιουργό τον Jonathan Taylor το 2009 εν συνέχεια την συντήρηση και επέκταση της ανέλαβαν οι skipper seabold και josef perktold. Μέσο αυτής της βιβλιοθήκης ο χρήστης μπορεί να αναπτύξει μοντέλα όπως την γραμμική παλινδρόμηση μέσο της εντολής statsmodels.OLS() και εν συνέχεια ο αναλυτής μπορεί να δει τα αποτελέσματα του μοντέλου όπως το p-value το R^2 το aic Bic τους συντελεστές συσχέτισης μέσο της εντολής print(summary()) , προφανώς η βιβλιοθήκη μπορεί να χρησιμοποιηθεί για την ανάπτυξη πολυωνυμιών παλινδρομήσεων όπου οι ερμηνευτικές μεταβλητές του μοντέλου είναι υψωμένες στο τετράγωνο, στο κύβο κλπ. Επίσης όπως θα δούμε και στο κεφάλαιο 7 η βιβλιοθήκη μπορεί να χρησιμοποιηθεί και για την ανάπτυξη multi linear , λογαριθμικών και εκθετικών παλινδρομήσεων.

Πέραν από τα μοντέλα παλινδρομήσεων η βιβλιοθήκη όπως είδαμε και στο κεφάλαιο 2 μπορεί να χρησιμοποιηθεί για διάφορους ελέγχους υποθέσεων όπως για παράδειγμα το t-test για τον έλεγχο της διαφοράς των μέσων ορών δυο δειγμάτων μέσο της εντολής stats.ttest(),η για τον έλεγχο Shapiro wilk που έλεγχοι για το αν τα δεδομένα ακολουθούν την κανονική κατανομή και εκτελείται μέσο της εντολής statsmodels.shapiro(), η για τον έλεγχο Levene όπου ελέγχει για το αν οι διακυμάνσεις δυο δειγμάτων είναι ίσες μέσο της εντολής statsmodels.levene(), ή ακόμα και για τον έλεγχο συσχέτισης μεταξύ δυο categorical μεταβλητών μέσο του ελέγχου chi-square. Επιπλέον η βιβλιοθήκη μπορεί να χρησιμοποιηθεί και για ελέγχους διακυμάνσεων μεταξύ δυο ή περισσοτέρων ομάδων δεδομένων (μέθοδος Anova).Επιπρόσθετα η statmodels ενδείκνυται για την ανάπτυξη glm τα οποία χρησιμοποιούνται στα μοντέλα μηχανικής μάθησης καθώς σε αντίθεση με τα απλά γραμμικά μοντέλα όπου η εξαρτημένη μεταβλητή ακολουθεί την κανονική κατανομή σε αυτά τα δεδομένα της εξαρτημένη μεταβλητής μπορεί να προέρχονται από την διονιμηκη την πουασον και την γαμμα κατανομή. Τέλος όπως θα δούμε και στο κεφάλαιο 7 με την ανάλυση των χρονολογικών σειρών η συγκεκριμένη βιβλιοθήκη χρησιμοποιείται για την ανάπτυξη μοντέλων όπως τα ar(αυτοπαλινδρομα) ma (κινητος μεσος ορος) Arma (συνδυασμός των δυο παραπάνω) , Arima (είναι το μοντέλο Arma μόνο που τα δεδομένα της ερμηνευτικής μεταβλητής έχουν διαφοροποιηθεί χ φορές), auto Arima(επιλεγεί τους συντελεστές q,d,p αυτόματα χωρίς ο χρηστής να πρέπει να ελέγξει τα διαγράμματα acf και pacf) και sarimax (είναι το μοντέλο Arima στο οποίο προσθετονται οι εποχικοί συντελεστές για την μοντελοποιηση της εποχικότητας και εξωγενείς μεταβλητές).

Η πιο γνωστή βιβλιοθήκη για την ανάπτυξη μοντέλων μηχανικής μάθησης στην python είναι η scicit learn η οποία αναπτύχθηκε αρχικά από τον David Cournapeau το 2007 ως μέρος ενός google summer of code project και πλέον έχει μετατραπεί σε ένα βασικό εργαλείο ανάπτυξης αλγορίθμων μηχανικής μάθησης σε πολλές επιχείρησης ανά τον πλανήτη καθώς μέσα από αυτήν οι χρήστες μπορούν να προ επεξεργαστούν τα δεδομένα δηλαδή να τα κανονικοποιησουν με διάφορους τρόπους όπως τον min max scaler(τα δεδομένα περνούν τιμές από το 0 έως το 1 ή το -1 έως το 1) , τον standardscaler (κάνει τα δεδομένα να ακολουθούν την κανονική κατανομή με μέσο ορό μηδέν και τυπική απόκλιση 1) , τον robust scaler(όπου βασίζεται στην διάμεσο τον δεδομένων και στο εύρος μετάξι του πρώτού και του τρίτοι τεταρτημόριού. Επίσης μπορεί να χρησιμοποιηθεί για να μετατρέψει τις κατηγορικές μεταβλητές σε ψευδομεταβλητες που περνούν τις τιμές μηδέν

και ένα μέσο του onehotrncoder(). Στην συνέχεια μπορεί επίσης να χρησιμοποιηθεί για να διαχωρίσει τα δεδομένα σε δεδομένα όπου θα εκπαιδευτεί το μοντέλο και σε εκείνα οπου θα ελεγχθεί η απόδοση του μέσο της εντολής train_test_split(). Και εφόσον έχει γίνει αυτή η προετοιμασία των δεδομένων μέσο αυτής της βιβλιοθήκης μπορούν να αναπτυχθούν μοντέλα ταξινόμησης όπως αυτά της logistic regression ,τα support vector machine ,τα random forests ,οι κ-κοντινότεροι γείτονες ,ο naïve bayes και αλλά. Επιπλέον μπορεί να χρησιμοποιηθεί και για την ανάπτυξη αλγορίθμων μηχανικής μάθησης για την ανάλυση χρονολογικών σειρών όπως την γραμμική παλινδρόμηση ,την random forest regression ,και την support vector regression. Πέρα από τα παραπάνω η βιβλιοθήκη περιλαμβάνει και αλγορίθμους για ανάπτυξη μοντέλων ομαδοποίησης (clustering) όπως τον k- means τον dbscan και άλλους που δεν θα αναλύσουμε σε αυτή την εργασία. Και τέλος εφόσον έχουν εκπαιδευτεί τα μοντέλα μέσο της αυτής μπορούν να αξιολογηθούν όπως είδαμε και στο κεφάλαιο 4 παραδείγματος χάρη τα μοντέλα ταξινόμησης αξιολογούνται μέσο του confusion matrix του f1-score, precision και του recall.

Τέλος η TensorFlow είναι μια από τις πιο γνωστές βιβλιοθήκες της python για ανάπτυξη αλγορίθμων βαθιάς μάθησης , η οποία αναπτύχθηκε από την ομάδα google brain και κυκλοφόρησε για πρώτη φορά το 2015 με αρχικό σκοπό την βελτιστοποίηση συστημάτων τεχνητής νοημοσύνης της google όπως για παράδειγμα την μηχανή αναζήτησης ,την μετάφραση και τις διαφημίσεις που θα εμφανίζονται στον εκάστοτε χρήστη του YouTube. Πλέον όμως η συγκεκριμένη βιβλιοθήκη έχει εφαρμογές σε πάρα πολλούς τομείς της οικονομίας, για παράδειγμα η google και το Facebook την χρησιμοποιούν για την αναγνώριση εικόνων , προσώπων και βίντεο μέσο νευρωνικων δικτυών όπως τα cnn ,πιο συγκεκριμένα η tesla χρησιμοποιεί την TensorFlow για την ανάπτυξη των αυτονόμων αυτοκίνητων της ,επιπλέον τέτοιοι αλγόριθμοι χρησιμοποιούνται κατά κύριο λόγο και στην ιατρική για την ανάλυση ακτινογραφιών και την διάγνωση καρκινικών κυττάρων στους ανθρώπους. επίσης αλγόριθμοι ann(artificial neural network) και rnn (recurrent neural networks) χρησιμοποιούνται από την Siri και άλλους ψηφιακούς βοηθούς για την αναγνώριση και επεξεργασία της φυσικής γλώσσας , την μετάφραση της ακόμα και για την ανάλυση συναισθημάτων και της πιθανότητας αγοράς ή και επαναγοράς προϊόντων από ένα κατάστημα. Άλλες επιχρίσεις όπως η PayPal την χρησιμοποιεί για την ανίχνευση απάτητης , ενώ επιχειρήσεις όπως η uber και η Airbnb για την ανάλυση της μελλοντικής ζητητής. Αναφορικά με τον τρόπο λειτουργίας της βιβλιοθήκης ο χρήστης εισάγει τα

δεδομένα σε μορφή πολυδιάστατων πινάκων (tensors) στην συνέχεια επιλεγεί και δομεί το μοντέλο και στο τελευταίο στάδιο γίνεται η εκπαίδευση και εξάγονται οι προβλέψεις του. Αναφορικά με την εκπαίδευση του νευρώνα αυτή γίνεται είτε μέσο του επεξεργαστή του υπολογιστή του χρήστη είτε μέσο της κάρτας γραφικών του ή τέλος μέσο cloud συστημάτων (tpu).

4. ΕΦΑΡΜΟΓΗ ΜΕΘΟΔΩΝ ΜΗΧΑΝΙΚΗΣ ΜΑΘΗΣΗΣ ΣΤΑ ΔΕΔΟΜΕΝΑ ΑΓΟΡΑΣ Ή ΜΗ ΑΓΟΡΑΣ ΤΩΝ ΠΕΛΑΤΩΝ ΕΝΟΣ ΗΛΕΚΤΡΟΝΙΚΟΥ ΚΑΤΑΣΤΗΜΑΤΟΣ(CUSTOMER PURCHASE BEHAVIOR)

Στην συνέχεια θα δοκιμάσουμε τις τεχνικές μηχανικής μάθησης που αναλύσαμε στο κεφάλαιο 4 με στόχο να ταξινομήσουμε το εάν οι πελάτες ενός ηλεκτρονικού καταστήματος με βάση διαφορά χαρακτηριστικά όπως την ηλικία τους ,το φύλο τους ,το ύψος του ετήσιου εισοδήματος τους ,την κατηγορία προϊόντος που σκοπεύουν να αγοράσουν ,το χρόνο που αφιέρωσαν στον ισότοπο της εταιρείας αλλά και το αν έχουν κάποιο πρόγραμμα μέλους στην εταιρία , το ποσοστό έκπτωσης που έχουν στην διάθεση τους για αγορά των προϊόντων που έχουν τοποθετήσει στο καλάθι ,και τέλος το αριθμό των προηγουμένων αγορών που έχουν πραγματοποιήσει στο κατάστημα αν εν τέλη αγοράσαν η όχι το προϊόν. Δηλαδή πρόκειται για ένα σύνολο δεδομένων που έχει ως στόχο την ανάλυση της ζήτησης με βάση χαρακτηριστικά των πελατών το οποίο συνδυάζει το μάρκετινγκ και την διαχείριση εφοδιαστικής αλυσίδας καθώς τελικός σκοπός της είναι ο πελάτης να αγοράσει ένα προϊόν ή μια υπηρεσία. Τα δεδομένα του dataset αποκτήθηκαν μέσο της πλατφόρμας Kaggle και οποιοσδήποτε θέλει να πειραματιστεί παραπάνω με αυτά δεν έχει πάρα να τα κατεβάσει πατώντας στον παρακάτω σύνδεσμο.

https://www.kaggle.com/datasets/rabieelkharoua/predict-customer-purchase-behavior-dataset

Αρχικά εισάγουμε τα δεδομένα στην python και τα μετατρέπουμε με την χρήση της βιβλιοθήκης pandas σε dataframe. Μπορούμε να δούμε ότι τα δεδομένα αποτελούνται από εννέα στήλες δηλαδή τα οκτώ διαφορετικά χαρακτηριστικά (ερμηνευτικές μεταβλητές) των πελατών και την εξαρτημένη μεταβλητή purchasestatus που παίρνει δυο τιμές 1 εάν ο πελάτης αγόρασε το προϊόν και 0 αν δεν αγόρασε το προϊόν εν τέλη. Επίσης τα δεδομένα αποτελούνται από 1500 γραμμές δηλαδή έχουμε δεδομένα 1500 πελατών. Στόχος του αναλυτή είναι να βρει εκείνες τις ερμηνευτικές μεταβλητές που έχουν το σημαντικότερο αντίκτυπο στην τελική απόφαση της αγοράς ή μη των προϊόντων. Στην συνέχεια ελέγχουμε το dataset για το εάν έχει κενές τιμές σε κάποια μεταβλητή που στην περίπτωση μας δεν ισχύει κάτι τέτοιο όλα τα κελιά είναι συμπληρωμένα. Πριν αναλύσουμε κάθε μια στήλη ξεχωριστά θα πρέπει να περιγράψουμε το τι είναι η εκάστοτε μεταβλητή. Αρχικά

υπάρχει η μεταβλητή age δηλαδή η ηλικία του κάθε πελάτη ,το gender δηλαδή το φύλο του πελάτη το οποίο παίρνει τιμές ίσο με 1 αν είναι γυναίκαι και 0 αν είναι άνδρας στην συνέχεια έχουμε την στήλη annual income το οποίο είναι το ετήσιο εισόδημα του εκάστοτε πελάτη , τον αριθμό των προηγουμένων αγορών από το ηλεκτρονικό κατάστημα(number of purchases) , την στήλη που περιέχει την κατηγορία προϊόντος που σκοπεύει να αγοράσει ο πελάτης (product category) η οποία παίρνει τιμές από 0 έως 4 το μηδέν αντιστοιχεί σε ηλεκτρονικά προϊόντα όπως κινητά τηλεφώνα , η τιμή 1 αντιστοιχεί σε προϊόντα ρουχισμού όπως μπλούζες μπουφάν κλπ. ο αριθμός 2 αντιστοιχεί σε προϊόντα σπιτιού ,ο 3 σε προϊόντα ομορφιάς όπως καλλυντικά και το 4 σε προϊόντα αθλητισμού. Η επόμενη μεταβλητή είναι ο χρόνος σε λεπτά που ο πελάτης αφιέρωσε για να ψάξει τα προϊόντα μέσα στο ηλεκτρονικό κατάστημα (στην ιστοσελίδα). Επίσης υπάρχει η μεταβλητή loyalty program που παίρνει τις τιμές 1 εάν ο πελάτης έχει κάρτα μέλους του καταστήματος και 0 αν δεν έχει κάρτα μέλους, και τέλος η μεταβλητή discountavailed η οποία δείχνει το ποσοστό της έκπτωσης που είναι διαθέσιμο στον πελάτη για να αγοράσει το συγκεκριμένο προϊόν ,οι τιμές που παίρνει η συγκεκριμένη μεταβλητή είναι από 0 έως 5%. Στην συνέχεια θα αναλύσουμε κάθε μεταβλητή ξεχωριστά για να είναι πιο ωραία η παρουσίαση των μεταβλητών στα διαγράμματα αντικαθιστούμε την πραγματική τιμή σαν λέξη στις μεταβλητές αντί για την αριθμητική τιμή όπως για παράδειγμα στην μεταβλητή φύλο αντικαθιστούμε την τιμή 0 με male και την τιμή 1 με female, κάνουμε την αντίστοιχη μετατροπή και στις υπόλοιπες μεταβλητές όπως το πρόγραμμα μέλους, την κατηγορία των προϊόντων το ποσοστό της έκπτωσης κλπ. Αρχικά θα αναλύσουμε μια προς μια τις μεταβλητές μετα θα αναλύσουμε τις μεταβλητές σε σχέση με την εξαρτημένη μεταβλητή (εάν αγόρασε η όχι το προϊόν ο πελάτης) στην συνέχεια θα κάνουμε μερικές παλινδρομήσεις μετάξι των μεταβλητών , μετέπειτα θα κάνουμε γραφική σύγκριση των μεταβλητών και θα ολοκληρώσουμε την ανάλυση δημιουργώντας και αναλύοντας τα αποτελέσματα των αλγορίθμων μηχανικής μάθησης για ταξινόμηση.

```
[2] import pandas as pd
    import numpy as np
    import matplotlib.pyplot as plt
    import seaborn as sns
```

```
[3] beh=pd.read_csv('customer_purchase_data.csv')
```

```
beh.isnull().sum()
```

	0
Age	0
Gender	0
AnnualIncome	0
NumberOfPurchases	0
ProductCategory	0
TimeSpentOnWebsite	0
LoyaltyProgram	0
DiscountsAvailed	0
PurchaseStatus	0

dtype: int64

	Age	Gender	AnnualIncome	NumberOfPurchases	ProductCategory	TimeSpentOnWebsite	LoyaltyProgram	DiscountsAvailed	PurchaseStatus
0	40	1	66120.267939	8	0	30.568601	0	5	1
1	20	1	23579.773583	4	2	38.240097	0	5	0
2	27	1	127821.306432	11	2	31.633212	1	0	1
3	24	1	137798.623120	19	3	46.167059	0	4	1
4	31	1	99300.964220	19	1	19.823592	0	0	1
...	...	...	...	...	...	...	...	...	...
1495	39	1	65048.141834	13	0	34.590743	0	5	1
1496	67	1	28775.331069	18	2	17.625707	0	1	1
1497	40	1	57363.247541	7	4	12.206033	0	0	0
1498	63	0	134021.775532	16	2	37.311634	1	0	1
1499	50	0	52625.665974	13	0	25.348017	1	4	1

1500 rows × 9 columns

εικόνα 61 : εισαγωγή δεδομένων

```
beh['ProductCategory'] = beh['ProductCategory'].replace({0: 'Electronics', 1: 'Clothing', 2: 'Home Goods', 3: 'Beauty', 4: 'Sports'})
beh['Gender'] = beh['Gender'].replace({0: 'Male', 1: 'Female'})
beh['LoyaltyProgram'] = beh['LoyaltyProgram'].replace({0: 'no', 1: 'yes'})
beh['DiscountsAvailed'] = beh['DiscountsAvailed'].replace({0: 'zero', 1: 'very_small',2: 'small', 3: 'medium',4: 'big', 5: 'very_big'})
beh['PurchaseStatus'] = beh['PurchaseStatus'].replace({0: 'no', 1: 'yes'})
beh
```

	Age	Gender	AnnualIncome	NumberOfPurchases	ProductCategory	TimeSpentOnWebsite	LoyaltyProgram	DiscountsAvailed	PurchaseStatus
0	40	Female	66120.267939	8	Electronics	30.568601	no	very_big	yes
1	20	Female	23579.773583	4	Home Goods	38.240097	no	very_big	no
2	27	Female	127821.306432	11	Home Goods	31.633212	yes	zero	yes
3	24	Female	137798.623120	19	Beauty	46.167059	no	big	yes
4	31	Female	99300.964220	19	Clothing	19.823592	no	zero	yes
...	...	...	...	...	...	...	...	...	...
1495	39	Female	65048.141834	13	Electronics	34.590743	no	very_big	yes
1496	67	Female	28775.331069	18	Home Goods	17.625707	no	very_small	yes
1497	40	Female	57363.247541	7	Sports	12.206033	no	zero	no
1498	63	Male	134021.775532	16	Home Goods	37.311634	yes	zero	yes
1499	50	Male	52625.665974	13	Electronics	25.348017	yes	big	yes

1500 rows × 9 columns

εικόνα 62:μετατροπή δεδομένων

Ξεκινάμε την ανάλυση μας με την μεταβλητή age και όπως θα δούμε το κατάστημα έχει πελάτες ηλικίας από 18 έως 70 ετών με μέσο ορό τα 44 έτη και τυπική απόκλιση ίση με 15, για την δημιουργία των διαγραμμάτων χωρίζουμε τις ηλικίες σε ηλικιακές ομάδες οι οποίες είναι πέντε η πρώτη είναι από 18 ετών έως 30 η δεύτερη από 31 έως 40 η τρίτη από 41 έως 50 η τέταρτη από 51 έως 60 και η πέμπτη από 61 και πάνω. Ουσιαστικά δημιουργούμε νέες μεταβλητές όπου εάν το άτομο έχει ηλικία αναμεσά σε ένα από αυτά τα διαστήματα μπαίνει στην αντίστοιχη η τιμή 1 και σε όλα τα υπόλοιπά η τιμή 0. Θα καούμε την ιδία μετατροπής και στις υπόλοιπές αριθμητικές τιμές και θα δοκιμάσουμε αν τα μοντέλα μηχανικής μάθησης ταξινομούν τα δεδομένα καλυτέρα σαν dummies(ψευδομεταβλητες με 0 και 1) η όπως είναι στην αρχική τους κατάσταση. Όσον αφορά την μεταβλητή ηλικία αθροίζουμε τις τιμές κάθε ηλικιακής ομάδας και

δημιουργούμε ενα διάγραμμα πίτας, όπου βλέπουμε ότι κάθε ηλικιακή ομάδα είναι το 20-25% του συνόλου των πελατών δηλαδή οι ομάδες είναι ισόποσες.

```
print(np.sort(beh['Age']))
```

```
[18 18 18 ... 70 70 70]
```

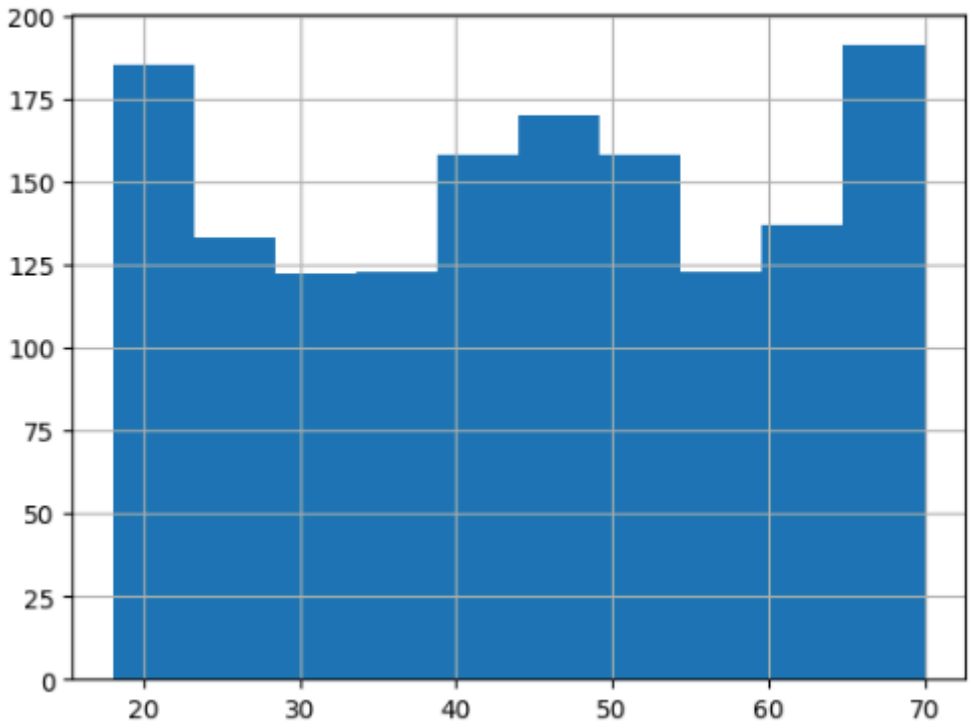
```
beh['Age'].describe()
```

	Age
count	1500.000000
mean	44.298667
std	15.537259
min	18.000000
25%	31.000000
50%	45.000000
75%	57.000000
max	70.000000

dtype: float64

```
beh['Age'].hist(bins=10)
```

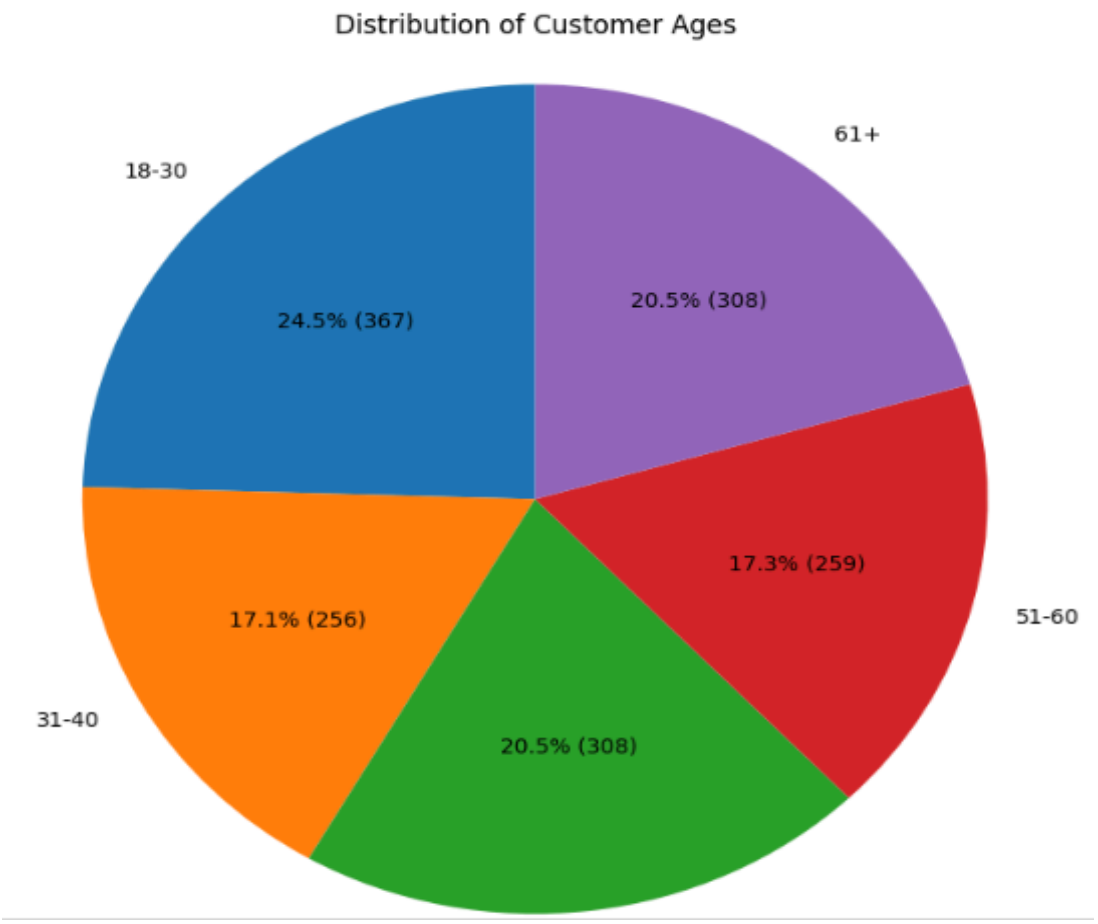
```
<Axes: >
```



Εικονα 63: ιστόγραμμα μεταβλητής age

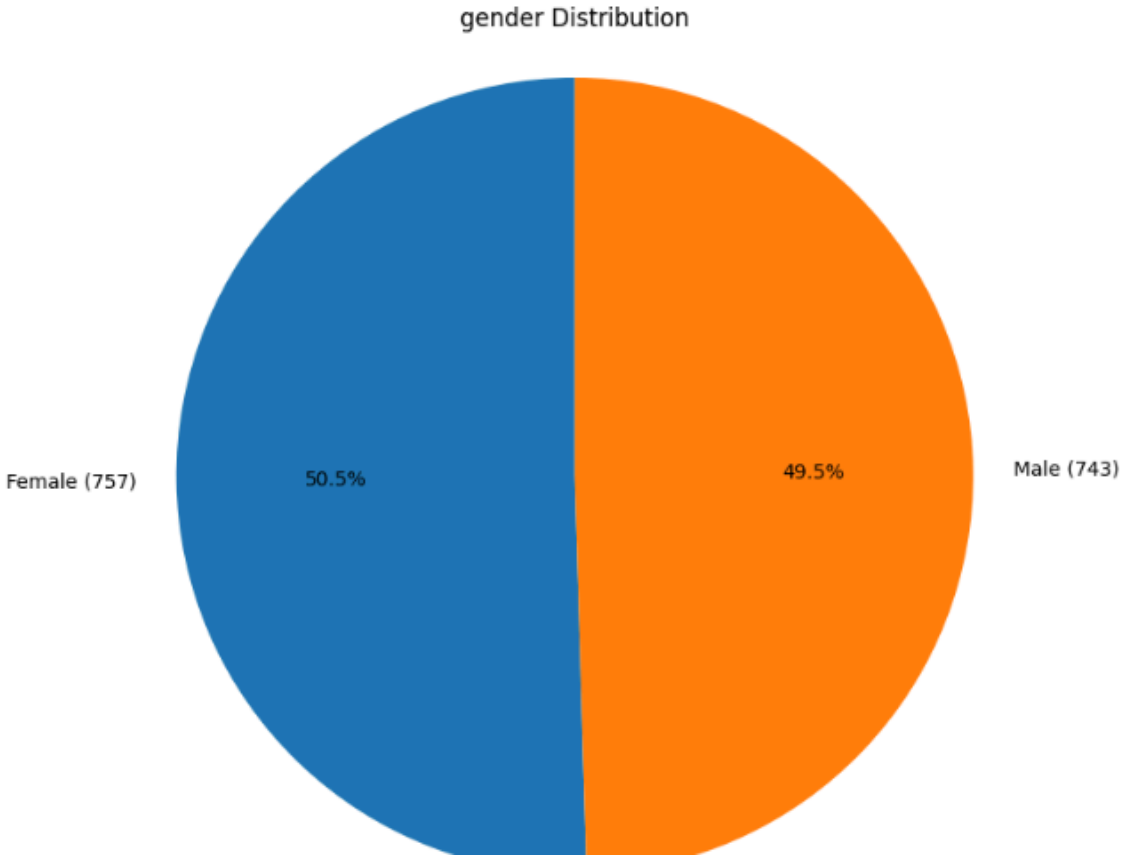
```
beh['age1830']=0
beh['age3140']=0
beh['age4150']=0
beh['age5160']=0
beh['age61above']=0
beh.loc[(beh['Age'] >= 18) & (beh['Age'] <= 30), 'age1830'] = 1
beh.loc[(beh['Age'] >= 31) & (beh['Age'] <= 40), 'age3140'] = 1
beh.loc[(beh['Age'] >= 41) & (beh['Age'] <= 50), 'age4150'] = 1
beh.loc[(beh['Age'] >= 51) & (beh['Age'] <= 60), 'age5160'] = 1
beh.loc[(beh['Age'] > 60) , 'age61above'] = 1
```

```
count1830 = beh['age1830'].value_counts().get(1, 0)
count3140 = beh['age3140'].value_counts().get(1, 0)
count4150 = beh['age4150'].value_counts().get(1, 0)
count5160 = beh['age5160'].value_counts().get(1, 0)
count61above = beh['age61above'].value_counts().get(1, 0)
age_groups = ['18-30', '31-40', '41-50', '51-60', '61+']
counts = [count1830, count3140, count4150, count5160, count61above]
total_count = sum(counts)
percentages = [(count / total_count) * 100 for count in counts]
plt.figure(figsize=(8, 8))
plt.pie(counts, labels=age_groups, autopct=lambda p: f'{p:.1f}% ({int(p * total_count / 100)})', startangle=90)
plt.title('Distribution of Customer Ages')
plt.axis('equal')
plt.show()
```



Εικονα 64: pie plot age

Στην συνέχεια φτιάχνουμε ένα διάγραμμα πίτας για την μεταβλητή φύλο και όπως παρατηρούμε και αυτές οι δυο ομάδες είναι ισόποσες έχοντας μόνο τέσσερις παραπάνω γυναίκες πελάτισσες.



εικόνα 65: διάγραμμα πίτας για το φύλο

Συνεχίζουμε την ανάλυση μας με την μεταβλητή του ετήσιου εισοδήματος, για να μπορέσουμε και αυτή την να την αναπαραστήσουμε γραφικά καούμε την ιδία μετατροπή με της ψευδομεταβλητες που είχαμε κάνει και στην μεταβλητή ηλικίας δηλαδή δημιουργώντας τέσσερεις ομάδες εισοδήματος. Στην πρώτη ομάδα ανήκουν οι πελάτες με εισόδημα από 20 έως 40 χιλιάδες δολάρια (το κατάστημα και οι πελάτες βρίσκονται στις ηνωμένες πολιτείες για αυτό και οι τιμές του εισοδήματος είναι τόσο μεγάλες) η δεύτερη κατηγορία αναφέρεται στου πελάτες που έχουν εισόδημα από 40000 έως 80000 δολάρια η τρίτη από 80000 μέχρι 120000 και η τελευταία περιέχει τους πελάτες με εισόδημα μεγαλύτερο των 120000 δολαρίων ετήσιος. Το μέσο εισόδημα είναι 84.249 και το μικρότερο είναι κοντά στα 20000 δολάρια ετήσιος ενώ το μεγαλύτερο είναι κοντά στις 150000. Στην συνέχεια υπολογίζουμε τον αριθμό των πελατών που βρίσκονται στην

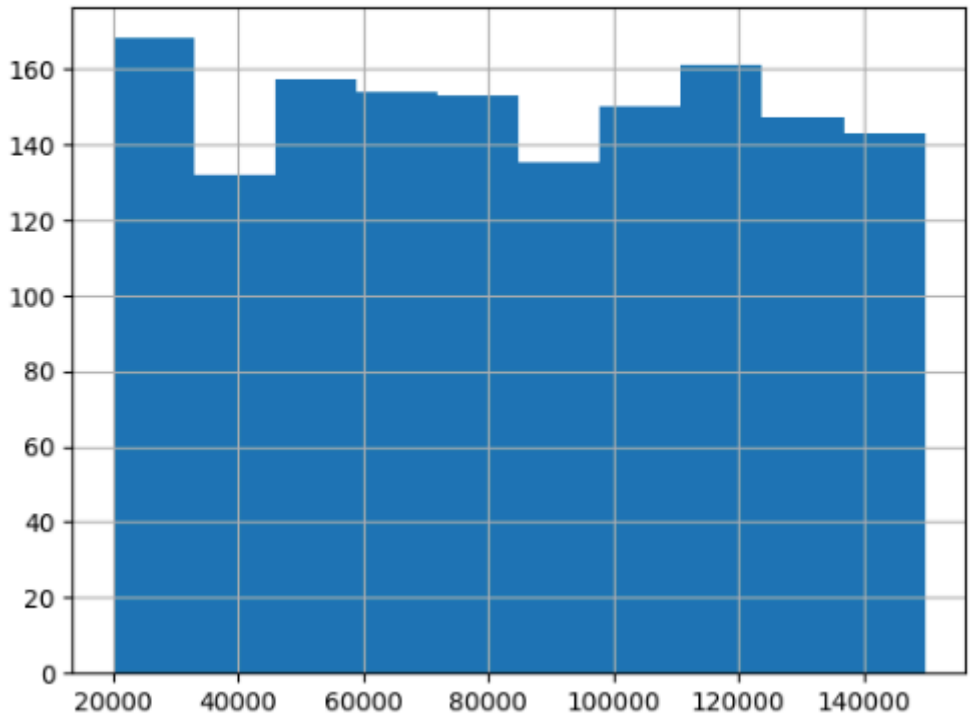
εκάστοτε ομάδα εισοδήματος και δημιουργούμε ένα διάγραμμα πίτας στο οποίο βλέπουμε ότι το 60% των πελατών έχει ετήσιο εισόδημα από 40 έως 120 χιλιάδες δολάρια.

```
beh['AnnualIncome'].describe()
```

	AnnualIncome
count	1500.000000
mean	84249.164338
std	37629.493078
min	20001.512518
25%	53028.979155
50%	83699.581476
75%	117167.772858
max	149785.176481

```
beh['AnnualIncome'].hist(bins=10)
```

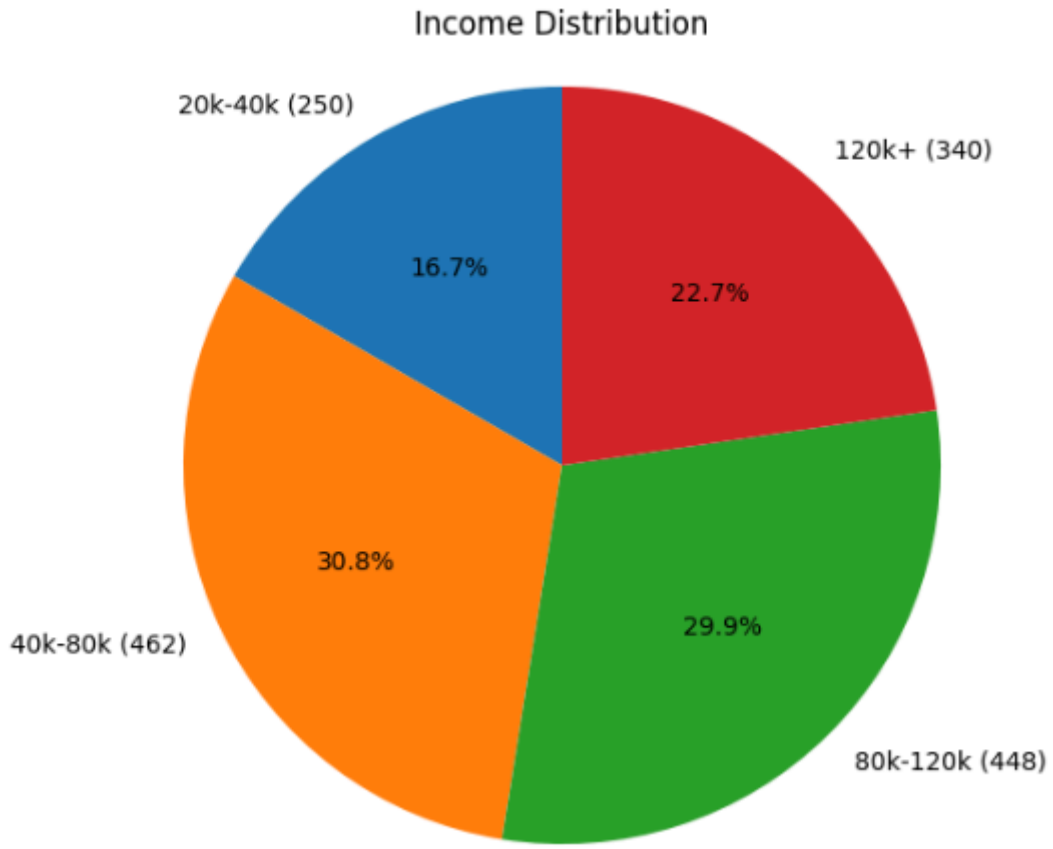
```
<Axes: >
```



```
beh['income2040k']=0
beh['income4080k']=0
beh['income80120k']=0
beh['income120kabove']=0
beh.loc[(beh['AnnualIncome'] >= 20000) & (beh['AnnualIncome'] <= 40000), 'income2040k'] = 1
beh.loc[(beh['AnnualIncome'] > 40000) & (beh['AnnualIncome'] <= 80000), 'income4080k'] =1
beh.loc[(beh['AnnualIncome'] > 80000) & (beh['AnnualIncome'] <= 120000), 'income80120k'] = 1
beh.loc[(beh['AnnualIncome'] > 120000) , 'income120kabove'] = 1
```

```
income2040k=beh['income2040k'].value_counts().get(1, 0)
income4080k=beh['income4080k'].value_counts().get(1, 0)
income80120=beh['income80120k'].value_counts().get(1, 0)
income120kabove=beh['income120kabove'].value_counts().get(1, 0)
print(income2040k)
print(income4080k)
print(income80120)
print(income120kabove)
```

```
250
462
448
340
```



εικόνα 66: ανάλυση και δημιουργία pie plot για το εισόδημα

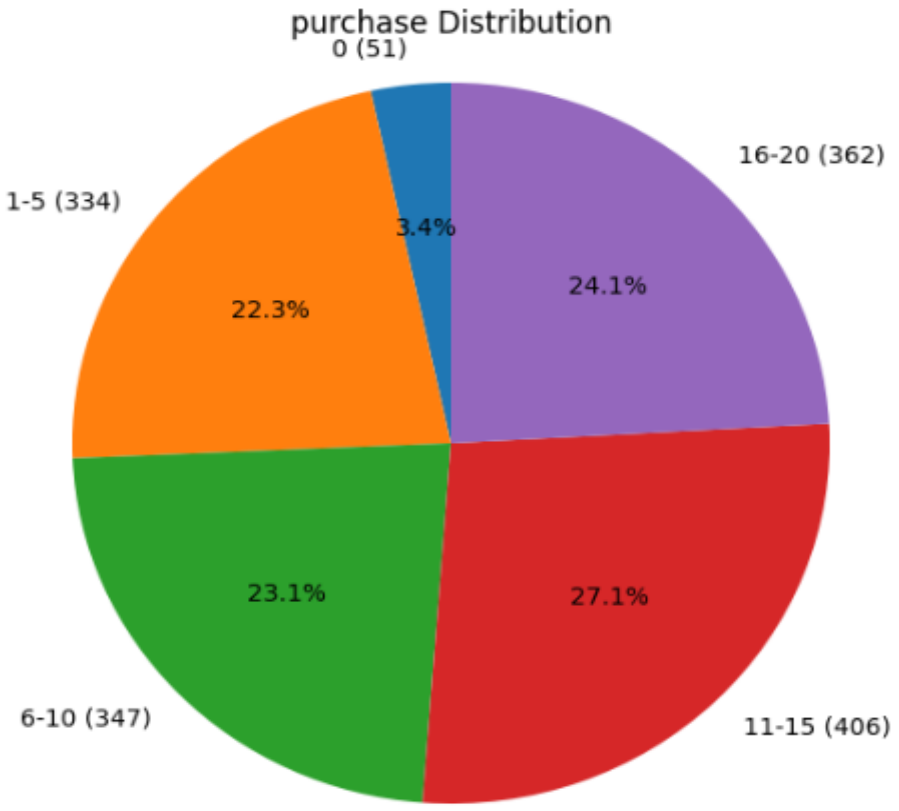
Στην συνέχεια εφαρμόζουμε την ιδία μετατροπή και στην μεταβλητή με τον αριθμό των προηγουμένων αγορών ανά καταναλωτή από το κατάστημα. Δημιουργούμε μια ομάδα αποκλειστικά για πελάτες όπου δεν έχουν ξαναγοράσει από το κατάστημα ώστε να τους αναλύσουμε διεξοδικά στην συνέχεια, η δεύτερη ομάδα είναι από μια προηγουμένη αγορά έως 5 η τρίτη από 6 έως 10 η τέταρτη από 11 έως 15 και η τελευταία από 16 και πάνω προηγούμενες αγορές. Ο μέσος ορός προηγουμένων αγορών συνολικά όλων των πελατών είναι 10,4 αγορές. Τέλος υπολογίζουμε το συνολο των πελατών ανά κατηγορία και δημιουργούμε το διάγραμμα πίτας , στο οποίο παρατηρούμε ότι οι ομάδες είναι ισομιρασμενες εκτός από τους πελάτες που αγοράζουν για πρώτη φορά από το κατάστημα που είναι 51 άτομα δηλαδή το 3,4% του συνόλου των πελατών.

```
beh['NumberOfPurchases'].describe()
```

	NumberOfPurchases
count	1500.000000
mean	10.420000
std	5.887391
min	0.000000
25%	5.000000
50%	11.000000
75%	15.000000
max	20.000000

```
beh['np0']=0
beh['np1_5']=0
beh['np6_10']=0
beh['np11_15']=0
beh['np16_20']=0
beh.loc[(beh['NumberOfPurchases'] <= 0) , 'np0'] = 1
beh.loc[(beh['NumberOfPurchases'] >= 1) & (beh['NumberOfPurchases'] <= 5), 'np1_5'] = 1
beh.loc[(beh['NumberOfPurchases'] >= 6) & (beh['NumberOfPurchases'] <= 10), 'np6_10'] = 1
beh.loc[(beh['NumberOfPurchases'] >= 11) & (beh['NumberOfPurchases'] <= 15), 'np11_15']=1
beh.loc[(beh['NumberOfPurchases'] >= 16) & (beh['NumberOfPurchases'] <= 20), 'np16_20']=1
purchase0=beh['np0'].value_counts().get(1, 0)
purchase1_5=beh['np1_5'].value_counts().get(1, 0)
purchase6_10=beh['np6_10'].value_counts().get(1, 0)
purchase11_15=beh['np11_15'].value_counts().get(1, 0)
purchase16_20=beh['np16_20'].value_counts().get(1, 0)
print(purchase0)
print(purchase1_5)
print(purchase6_10)
print(purchase11_15)
print(purchase16_20)
```

```
51
334
347
406
362
```



εικόνα 67: ανάλυση και δημιουργία pie plot για τα δεδομένα των προηγουμένων αγορών

Η ιδία μεθοδολογία ακολουθείτε και για την μεταβλητή του χρόνου όπου ο εκάστοτε πελάτης αφιέρωσε για να ψάξει και να δει τα προϊόντα μέσα στο ηλεκτρονικό κατάστημα.

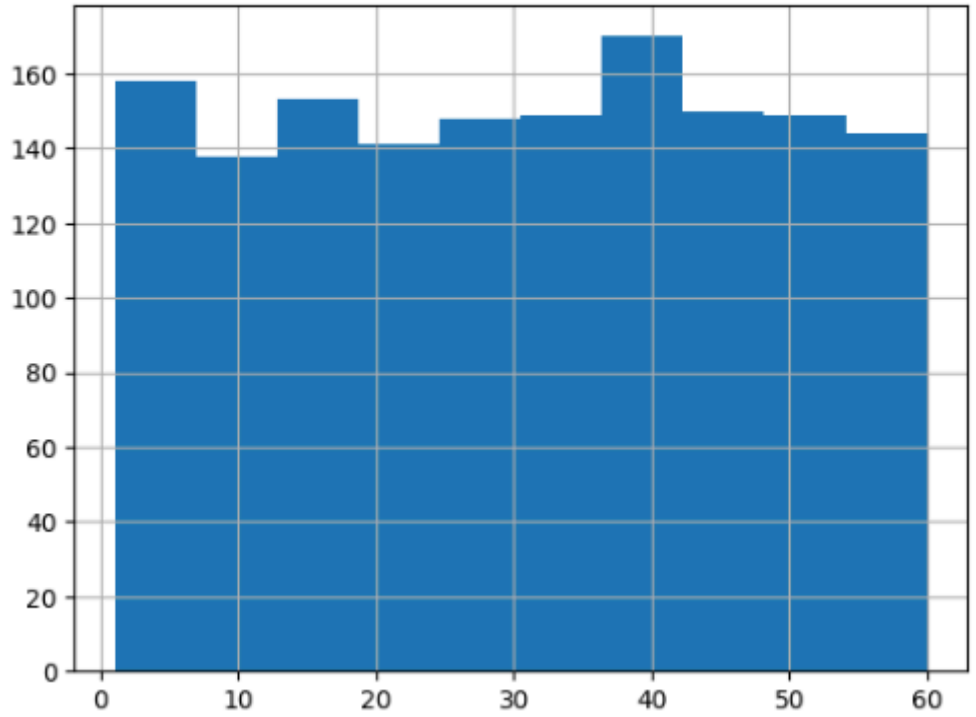
Ο μέσος χρόνος των 1500 πελατών είναι τα 30 λεπτά με ελάχιστο χρόνο το 1 λεπτό και μέγιστο χρόνο την μια ώρα (59 λεπτά). Για να δημιουργήσουμε το δίγραμμα πίτας χωρίζουμε τους πελάτες σε 4 ομάδες με βάση το χρόνο που διέθεσαν στο κατάστημα στην πρώτη ομάδα ανήκουν οι πελάτες που διέθεσαν από 0 έως 15 λεπτά η δεύτερη από 16 έως 30 λεπτά η τρίτη από 31 έως 45 και στην τέταρτη ανήκουν οι πελάτες που αφιέρωσαν περισσότερο από 45 λεπτά στο ισότοπο του ηλεκτρονικού καταστήματος. Και σε αυτό το διάγραμμα παρατηρούμε ότι οι ομάδες είναι ισόποσες με 23 έως 28% για την κάθε μια

```
beh['TimeSpentOnWebsite'].describe()
```

	TimeSpentOnWebsite
count	1500.000000
mean	30.469040
std	16.984392
min	1.037023
25%	16.156700
50%	30.939516
75%	44.369863
max	59.991105

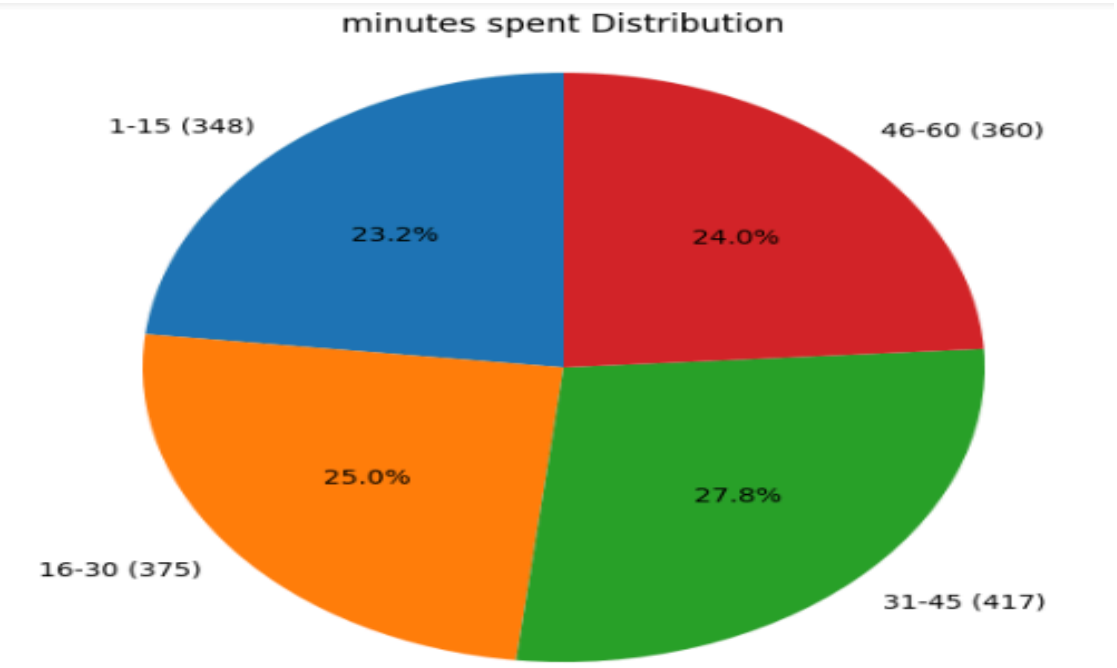
```
beh['TimeSpentOnWebsite'].hist(bins=10)
```

```
<Axes: >
```



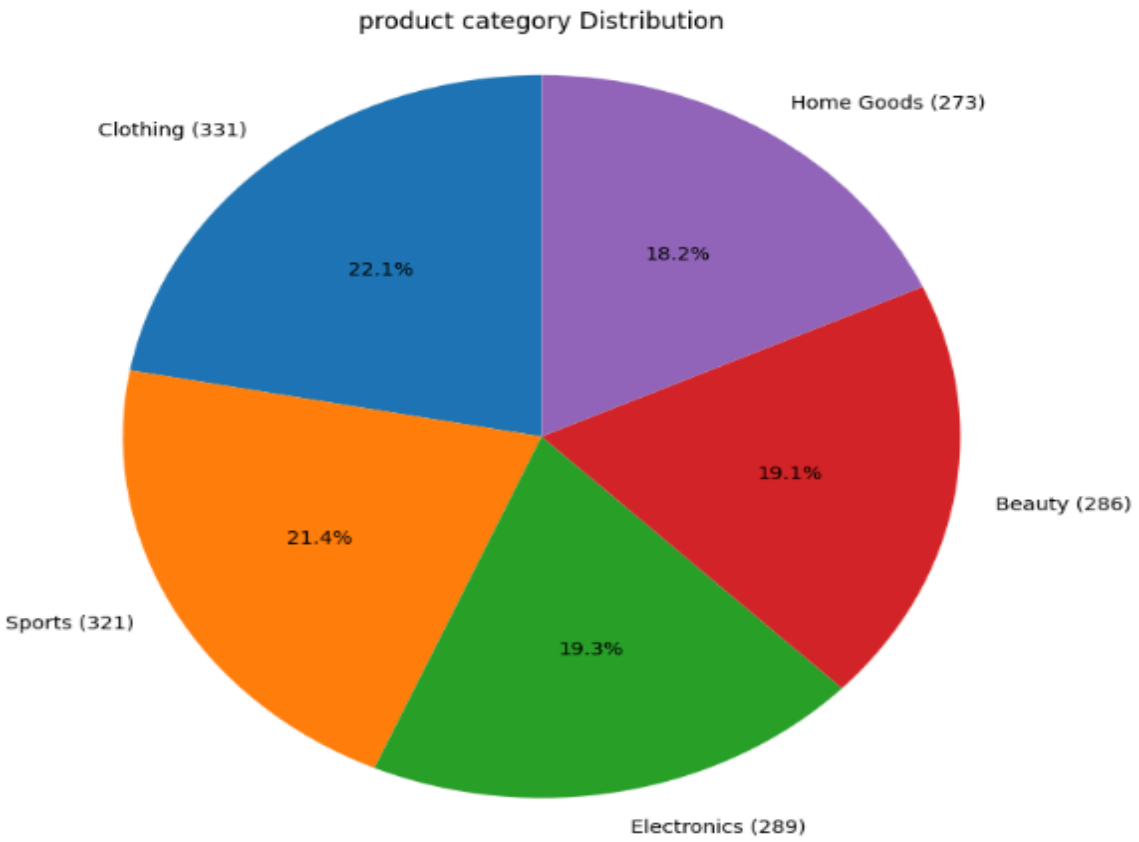

```
beh['min015']=0
beh['min1630']=0
beh['min3145']=0
beh['min4660']=0
beh.loc[(beh['TimeSpentOnWebsite'] >= 0) & (beh['TimeSpentOnWebsite'] <= 15), 'min015'] = 1
beh.loc[(beh['TimeSpentOnWebsite'] > 15) & (beh['TimeSpentOnWebsite'] <= 30), 'min1630'] = 1
beh.loc[(beh['TimeSpentOnWebsite'] > 30) & (beh['TimeSpentOnWebsite'] <= 45), 'min3145'] = 1
beh.loc[(beh['TimeSpentOnWebsite'] > 45) , 'min4660'] = 1
min015=beh['min015'].value_counts().get(1, 0)
min1630=beh['min1630'].value_counts().get(1, 0)
min3145=beh['min3145'].value_counts().get(1, 0)
min4660=beh['min4660'].value_counts().get(1, 0)
print(min015)
print(min1630)
print(min3145)
print(min4660)
```

```
348
375
417
360
```



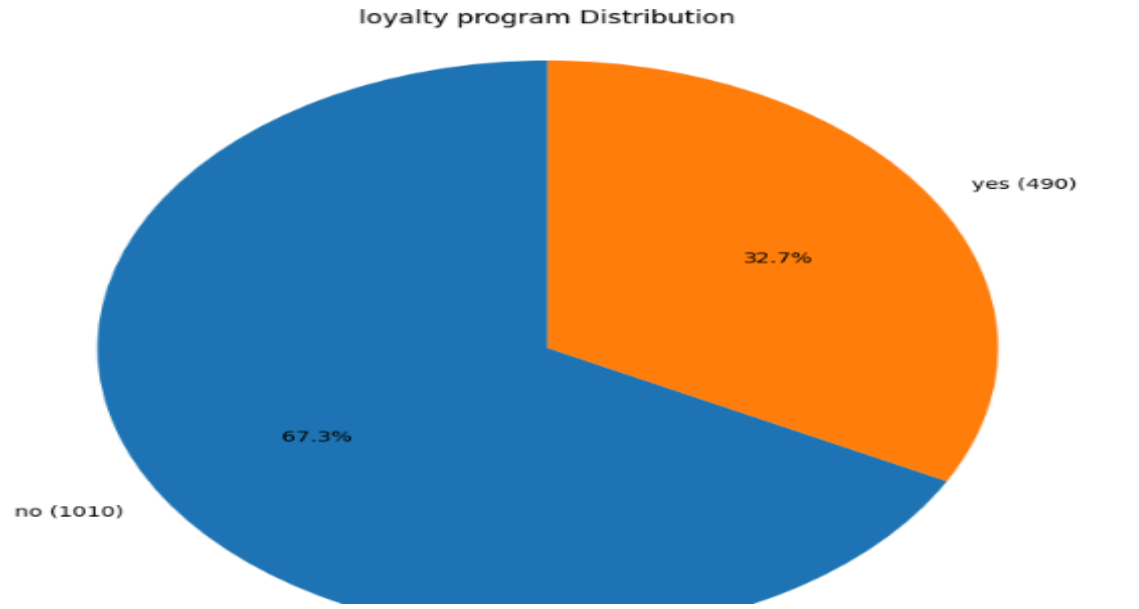
εικόνα 68: ανάλυση και δημιουργία pie plot για το χρόνο στο site

Συνεχίζουμε δημιουργώντας τα διαγράμματα πίτας για τα κατηγορηματικά δεδομένα της της κατηγορίας προϊόντος όπου επιλέγουν οι πελάτες να βάλουν στο καλάθι αγορών τους. Όπως βλέπουμε η επιλογή είδους προϊόντων είναι και αυτή ισομιρασμενη παρόλα αυτά οι περισσότεροι βάζουν στο καλάθι τους προϊόντα ρουχισμού και άθλησης.



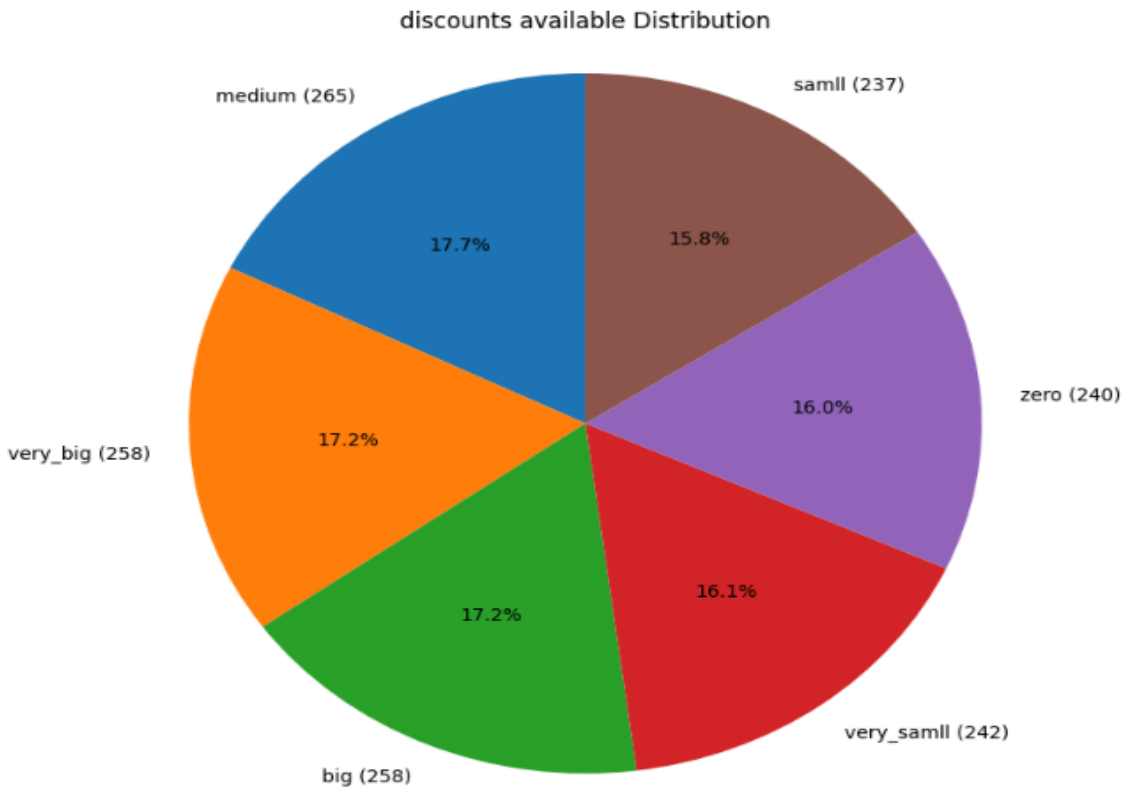
Εικονα 69: pie plot product category

Όσον αφορά το ποσοστό των πελατών που ψωνίζουν από το κατάστημα και έχουν κάρτα μέλους είναι πολύ λιγότεροι σε σχέση με εκείνους που μπαίνουν στο ηλεκτρονικό κατάστημα και δεν έχουν κάρτα μέλους.



Εικονα 70: pie plot product loyalty program

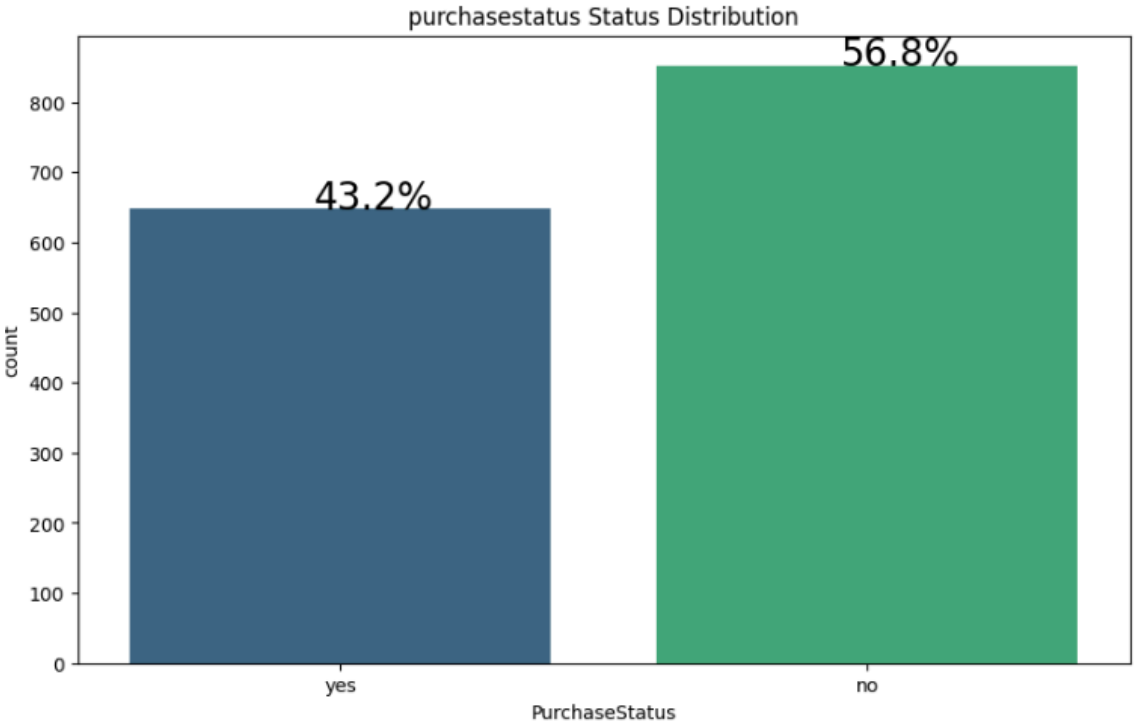
Αναφορικά με το ποσοστό έκπτωσης κατά την αγορά βλέπουμε ότι είναι ισομιρασμενα. Το μεγαλύτερο ποσοστό αντιστοιχεί στην έκπτωση medium δηλαδή στο 3% της συνολικής τιμής, Και ακολουθούν το very_big(5% έκπτωση) και big (4%) αντίστοιχα, μόνο το 16% του συνόλου των πελατών έχουν μηδενική έκπτωση.



Εικονα 71: pie plot ανα διαθέσιμη έκπτωση

Προχωράμε στην ανάλυση της κάθε μεταβλητής σε σχέση με το αν ο πελάτης αγόρασε η όχι από το κατάστημα(purchase status). Αρχικά δημιουργούμε ένα διάγραμμα count plot με την χρήση της βιβλιοθήκης seaborn για την συγκεκριμένη μεταβλητή και παρατηρούμε ότι το 57% του συνόλου των πιθανών αγοραστών των προϊόντων τελικά δεν αγόρασε το προϊόν που είχε βάλει στο καλάθι του.

```
plt.figure(figsize=(10, 6))
ax = sns.countplot(x='PurchaseStatus', data=beh, palette='viridis')
plt.title('purchasestatus Status Distribution')
total = len(beh)
for p in ax.patches:
    percentage = '{:.1f}%'.format(100 * p.get_height()/total)
    x = p.get_x() + p.get_width() / 2 - 0.05
    y = p.get_y() + p.get_height()
    ax.annotate(percentage, (x, y), size = 20)
plt.show()
```

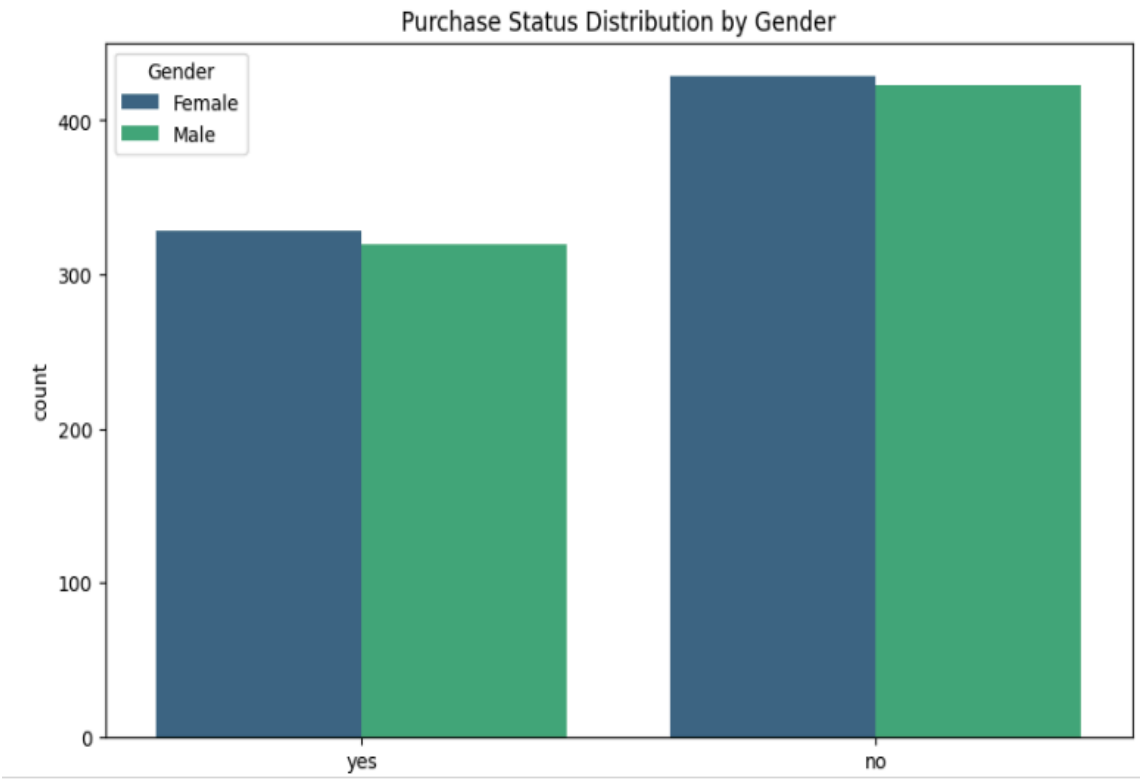


Εικονα 72: count plot purchase status

Στην συνέχεια δημιουργούμε τα αντίστοιχα count plot με βάση και τις μεταβλητές που είδαμε παραπάνω όπως για παράδειγμα το φύλο του πελάτη, για να γίνει αυτό διαχωρίζουμε το dataset ανάλογα με το αν ο πελάτης είναι άνδρας ή γυναικά, με αυτό τον τρόπο μπορούμε να υπολογίσουμε ποσοστά όπως πόσοι άνδρες αγοράσανε το προϊόν που είχαν βάλει στο καλάθι τους σε σχέση με το ποσοστό των ανδρών που δεν το αγοράσανε, το ίδιο μπορείς να γίνει και για τις γυναίκες. Η ιδία μεθοδολογία ακολουθείτε και στα υπόλοιπα διαγράμματα του ιδίου τύπου. Αυτό που βλέπουμε με βάση το διάγραμμα είναι ότι και οι γυναίκες και οι άνδρες αγοράζουν ή απορρίπτουν την αγορά ενός προϊόντος σε κοινό ποσοστό δηλαδή το φύλο δεν παίζει ρολό στην αγορά των προϊόντων γενικά, στην συνέχεια θα ψάξουμε σε βάθος τι προϊόντα προτιμάει το κάθε φύλο.

```
plt.figure(figsize=(10, 6))
ax = sns.countplot(x='PurchaseStatus', hue='Gender', data=beh, palette='viridis')
plt.title('Purchase Status Distribution by Gender')
male_data = beh[beh['Gender'] == 'Male']
male_yes = len(male_data[male_data['PurchaseStatus'] == 'yes'])
male_no = len(male_data[male_data['PurchaseStatus'] == 'no'])
total_male = male_yes + male_no
Female_data = beh[beh['Gender'] == 'Female']
Female_yes = len(Female_data[Female_data['PurchaseStatus'] == 'yes'])
Female_no = len(Female_data[Female_data['PurchaseStatus'] == 'no'])
total_Female = Female_yes + Female_no
Female_yes_percentage = (Female_yes / total_Female) * 100
Female_no_percentage = (Female_no / total_Female) * 100
print(f"Female - Purchase Status Yes: {Female_yes_percentage:.2f}%")
print(f"Female - Purchase Status No: {Female_no_percentage:.2f}%")
male_yes_percentage = (male_yes / total_male) * 100
male_no_percentage = (male_no / total_male) * 100
print(f"Male - Purchase Status Yes: {male_yes_percentage:.2f}%")
print(f"Male - Purchase Status No: {male_no_percentage:.2f}%")
plt.show()
```

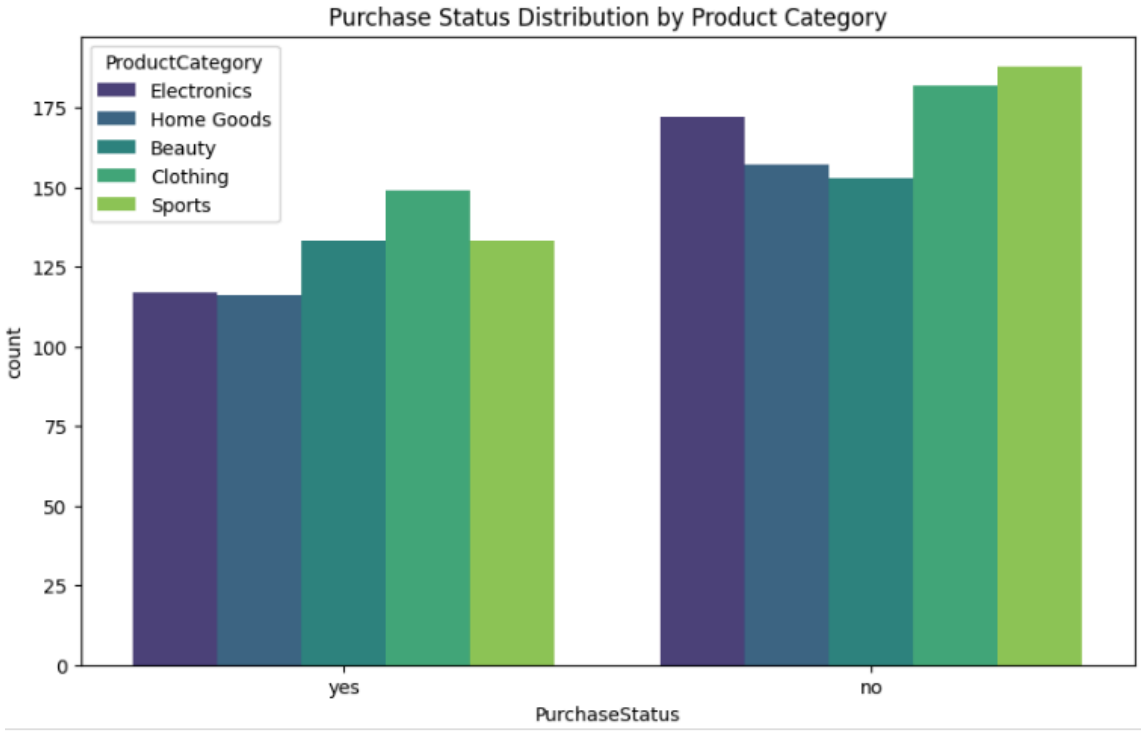
```
Female - Purchase Status Yes: 43.33%
Female - Purchase Status No: 56.67%
Male - Purchase Status Yes: 43.07%
Male - Purchase Status No: 56.93%
```



Εικονα 73: count plot purchase status and gender

Η επόμενη μεταβλητή που αναλύουμε είναι η κατηγορία προϊόντος. Όπως παρατηρούμε τα προϊόντα της κατηγορίας ομορφιάς είναι εκείνα που έχουν το μικρότερο ποσοστό απόρριψης(μη αγοράς) ίσως αυτό να έχει να κάνει με την τιμή τους μια μεταβλητή που όμως δεν έχουμε στην διάθεση μας. Ενώ από την άλλη πλευρά τα προϊόντα που μπαίνουν στο καλάθι του πελάτη αλλά τελικά δεν αγοράζονται είναι τα ηλεκτρονικά προϊόντα τα οποία ακολουθούν τα προϊόντα αθλητισμού και τα είδη σπιτιού.

```
Electronics - Purchase Status Yes: 40.48%
Electronics - Purchase Status No: 59.52%
Home Goods - Purchase Status Yes: 42.49%
Home Goods - Purchase Status No: 57.51%
Beauty - Purchase Status Yes: 46.50%
Beauty - Purchase Status No: 53.50%
Clothing - Purchase Status Yes: 45.02%
Clothing - Purchase Status No: 54.98%
Sports - Purchase Status Yes: 41.43%
Sports - Purchase Status No: 58.57%
```

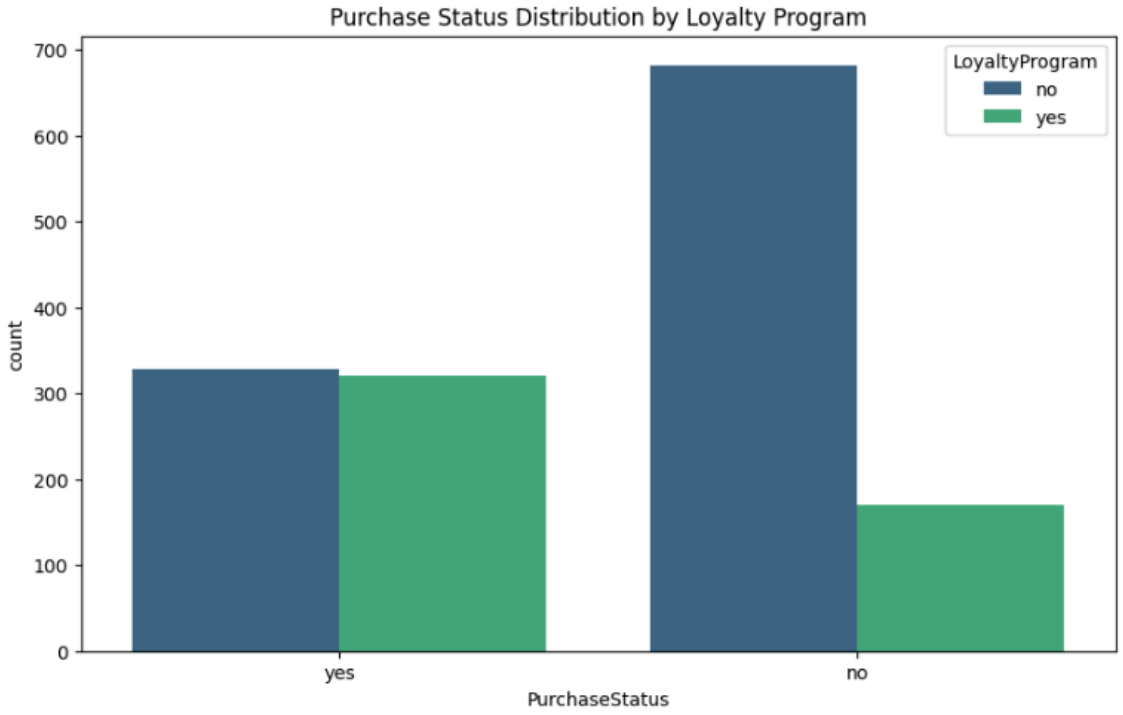


Εικονα 74: count plot purchase status and product category

Εν συνέχεια δημιουργούμε τα αντίστοιχα ποσοστά όσον αφορά τους πελάτες που έχουν ή δεν έχουν κάρτα μέλους. Όπως παρατηρούμε η συγκεκριμένη μεταβλητή είναι πολύ σημαντική στην απόφαση του πελάτη για το αν θα αγοράσει ή όχι τα προϊόντα στο καλάθι τους καθώς όπως βλέπουμε παρότι οι περισσότεροι πελάτες δεν έχουν κάρτα μέλους (δεν

αναγράφεται πως μπορεί να αποκτήσει κάποιος πελάτης την κάρτα μέλους αν είναι δωρεάν ή όχι αν πρέπει να κάνει κάποιο αριθμό αγορών πρώτα και μετα να την αποκτήσει ή τι παροχές δίνει στους πελάτες) εάν ο πελάτης έχει κάρτα μέλους υπάρχει 65% πιθανότατα να αγοράσει τελικά τα προϊόντα ενώ αν δεν έχει η πιθανότητα αγοράς είναι περίπου 32% .

```
loyalty no - Purchase Status Yes: 32.48%
loyalty no - Purchase Status No: 67.52%
loyalty yes - Purchase Status Yes: 65.31%
loyalty yes - Purchase Status No: 34.69%
```



Εικονα 75: count plot purchase status and loyalty program

Αναφορικά με το ποσοστό έκπτωσης και την πιθανότητα ολοκλήρωσης της αγοράς βλέπουμε ότι όσο αυξάνεται το ποσοστό της έκπτωσης τόσο πιο πιθανό είναι και ο πελάτης να αγοράσει το προϊόν για παράδειγμά το ποσοστό αγοράς με ποσοστό έκπτωσης 5%(very big) είναι 60% ενώ το ποσοστό αγοράς με ποσοστό έκπτωσης μηδέν, πολύ μικρό καθώς είναι σχεδόν 25%, δηλαδή παρατηρούμε ότι στους πελάτες όπου θα τους δοθεί έκπτωση άνω του 2% είναι αρκετά πιθανό να πραγματοποιήσουν την αγορά.

```
Discount very_big - Purchase Status Yes: 59.69%
Discount very_big - Purchase Status No: 40.31%
Discount zero - Purchase Status Yes: 27.50%
Discount zero - Purchase Status No: 72.50%
Discount big - Purchase Status Yes: 63.95%
Discount big - Purchase Status No: 36.05%
Discount samll - Purchase Status Yes: 23.21%
Discount samll - Purchase Status No: 76.79%
Discount medium - Purchase Status Yes: 55.09%
Discount medium - Purchase Status No: 44.91%
Discount very_samll - Purchase Status Yes: 25.62%
Discount very_samll - Purchase Status No: 74.38%
```

Εικονα 76: count plot purchase status and discount

Για να δημιουργήσουμε το ίδιο διάγραμμα με της ηλικιακές ομάδες ,τις ομάδες εισοδήματος ,της ομάδες με βάση τον χρόνο στην ιστοσελίδα και τον αριθμό των προηγουμένων αγορών , πρέπει αντί για της ψευδομεταβλητες με ένα και μηδέν να δημιουργήσουμε μια νέα στήλη με κατηγορηματικές μεταβλητές που θα αλλάζει το όνομα της τιμής ανάλογα με το διάστημα(οροί) που ανήκει η εκάστοτε τιμή (ηλικία, εισόδημα κλπ.) ακολουθεί ένα παράδειγμα για την μεταβλητή ηλικία όπως βλέπουμε η νέα στήλη ονομάζεται agecategory και ανάλογα το διάστημα ηλικιών που ανήκει η ηλικία του εκάστοτε πελάτη στην νέα μεταβλητή μπαίνει μια λέξη που δείχνει σε πια ηλικιακή ομάδα ανήκει για παράδειγμα αν η ηλικία ενός ατόμου είναι 24 τότε στην μεταβλητή

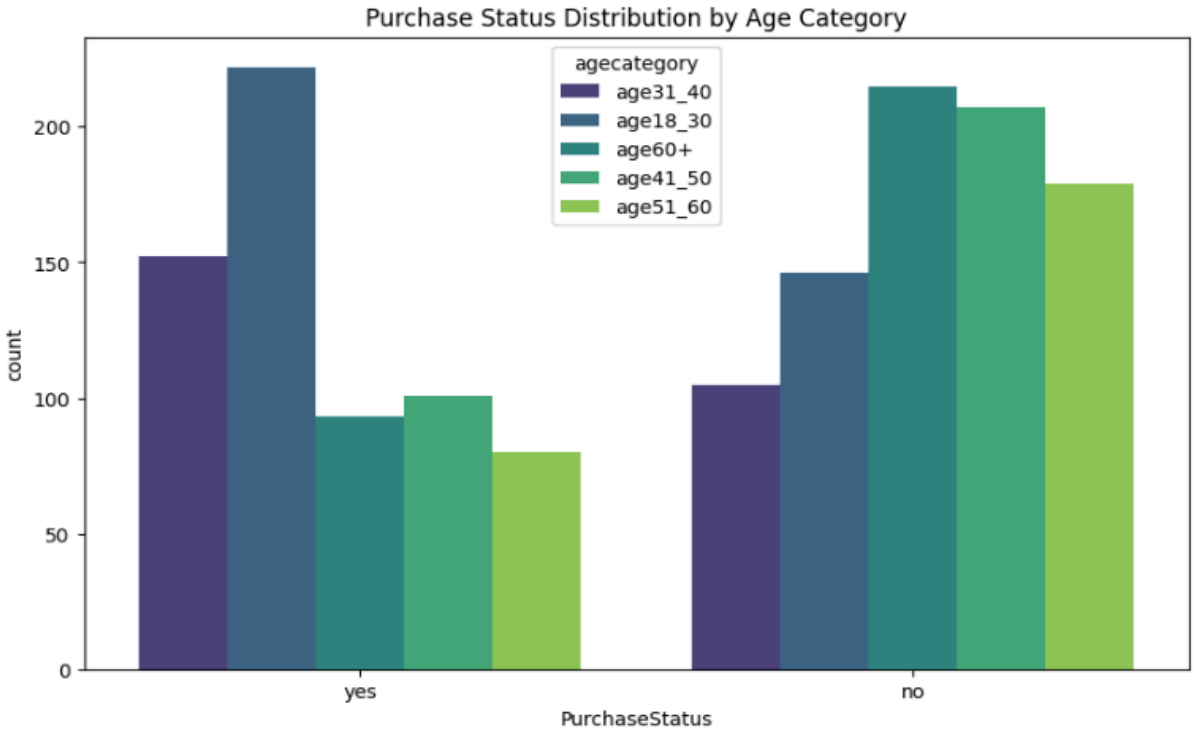
agecategory θα μπει η τιμή age18_30.οι μεταβλητές που έχουν κατηγορηματικές μεταβλητές δεν μπορούν να χρησιμοποιηθούν στα μοντέλα μηχανικής μάθησης και πρέπει να μετατραπούν ή σε ψευδομεταβλητες με διαστήματα όπως κάναμε παραπάνω ή να χρησιμοποιηθούν οι κανονικές αρχικές τιμές της ηλικίας.

```
beh['agecategory']='a'
beh.loc[(beh['Age'] >= 18) & (beh['Age'] <= 30), 'agecategory'] = 'age18_30'
beh.loc[(beh['Age'] >= 31) & (beh['Age'] <= 40), 'agecategory'] = 'age31_40'
beh.loc[(beh['Age'] >= 41) & (beh['Age'] <= 50), 'agecategory'] = 'age41_50'
beh.loc[(beh['Age'] >= 51) & (beh['Age'] <= 60), 'agecategory'] = 'age51_60'
beh.loc[(beh['Age'] > 60) , 'agecategory'] = 'age60+'
```

Εικονα 77: δημιουργία ψευδομεταβλητων

Όπως παρατηρούμε από τα διαγράμματα που δημιουργήσαμε οι ηλικιακές ομάδες που κατά κύριο λόγο ολοκληρώνουν την αγορά τους στο ηλεκτρονικό κατάστημα είναι οι πελάτες ηλικίας από 18 έως 30 και από 31 έως 40 με ποσοστό αγοράς ίσο με 60% ενώ από την άλλη οι μεγαλύτερης ηλικίας έχουν πιθανότητα να ολοκληρώσουνε την διαδικτυακή αγορά σε ποσοστό μόλις 30% το οποίο σημαίνει ότι είτε το κατάστημα πουλάει προϊόντα σε πιο νέα άτομα ή ότι οι άνθρωποι άνω των 41 δεν εμπιστεύονται της ηλεκτρονικές αγορές (δεν έχουμε πληροφορίες για το αν το κατάστημα λειτουργεί μόνο ηλεκτρονικά ή μπορεί ένας πελάτης να πραγματοποιήσει την αγορά του από το φυσικό κατάστημα της εταιρείας)

```
age31_40 - Purchase Status Yes: 59.14%
age31_40 - Purchase Status No: 40.86%
age18_30 - Purchase Status Yes: 60.33%
age18_30 - Purchase Status No: 39.67%
age60+ - Purchase Status Yes: 30.19%
age60+ - Purchase Status No: 69.81%
age41_50 - Purchase Status Yes: 32.79%
age41_50 - Purchase Status No: 67.21%
age51_60 - Purchase Status Yes: 30.89%
age51_60 - Purchase Status No: 69.11%
```

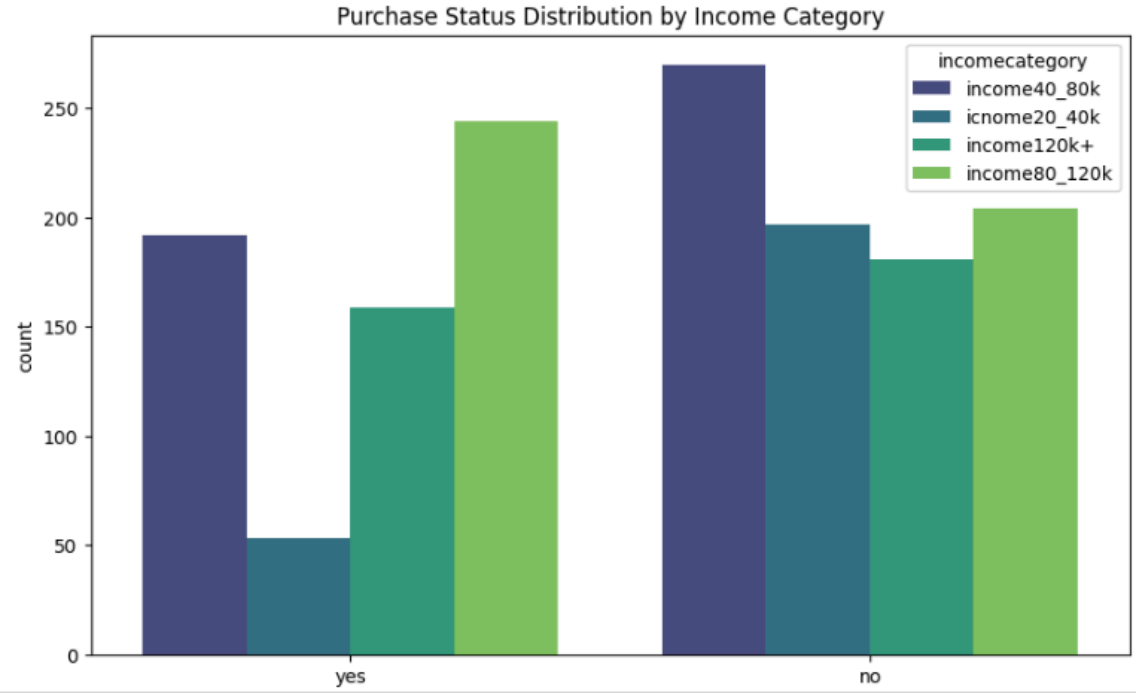


Εικονα 78: count plot purchase status and age

Όπως δημιουργήσαμε την μεταβλητή με της κατηγορηματικές μεταβλητές για της ηλικιακές ομάδες κάνουμε ακριβός το ίδιο και για τη ομάδες με βάση το εισόδημα , και δημιουργούμε τα αντίστοιχα ποσοστά και διαγράμματα. Όπως μπορούμε να παρατηρήσουμε το ύψος του εισοδήματος δεν παίζει πολύ σημαντικό λόγο για την τελική απόφαση της αγοράς ή μη καθώς μόνο στην ομάδα με εισόδημα μετάξι 20 και 40 χιλιάδων παρατηρείται πολύ μικρό ποσοστό ολοκλήρωσης της αγοράς , ενώ στα υψηλοτέρα εισοδήματα το ποσοστό αυξάνεται αλλά στην ομάδα με εισόδημα άνω των 120 χιλιάδων το ποσοστό αγοράς μειώνεται αυτό ίσως σηματοδοτεί ότι τα προϊόντα που εμπορεύεται η εταιρεία δεν είναι αρκετά premium (υψηλής ποιότητας) και αρά τα άτομα με πολύ υψηλό

εισόδημα να τα απορρίπτουν. Πάντως οι περισσότεροι πελάτες τις εταιρείας όπου ολοκληρώνουν την αγορά έχουν εισόδημα 80 με 120 χιλιάδες δολάρια ανά έτος.

```
income40_80k - Purchase Status Yes: 41.56%
income40_80k - Purchase Status No: 58.44%
icnome20_40k - Purchase Status Yes: 21.20%
icnome20_40k - Purchase Status No: 78.80%
income120k+ - Purchase Status Yes: 46.76%
income120k+ - Purchase Status No: 53.24%
income80_120k - Purchase Status Yes: 54.46%
income80_120k - Purchase Status No: 45.54%
```



Εικονα 79: count plot purchase status and income

Η επόμενη μεταβλητή που θα αναλύσουμε είναι η σχέση μεταξύ του αριθμού των προηγουμένων αγορών που πραγματοποίησε ο πελάτης στο ηλεκτρονικό κατάστημα σε σχέση με το αν αγόρασε ή όχι το προϊόν που έβαλε στο καλάθι αγορών του. Όπως παρατηρούμε οι πελάτες που μπαίνουν για πρώτη φορά στην ιστοσελίδα δεν ολοκληρώνουν την αγορά σε ποσοστό 78,4% το ίδιο συμβαίνει και με τους πελάτες που έχουν πραγματοποιήσει 1 έως 5 αγορές από εκεί και πέρα οι πελάτες με παραπάνω από 5 αγορές έχουν πιθανότητα να ολοκληρώσουν την αγορά κατά 50%, αρά η συγκεκριμένη μεταβλητή ίσως να μην διαδραματίζει σημαντικό ρολό και αρά να μην υπάρχει κάποιο brand loyalty δηλαδή οι πελάτες να αγοράζουν με βάση την τιμή του προϊόντος και την

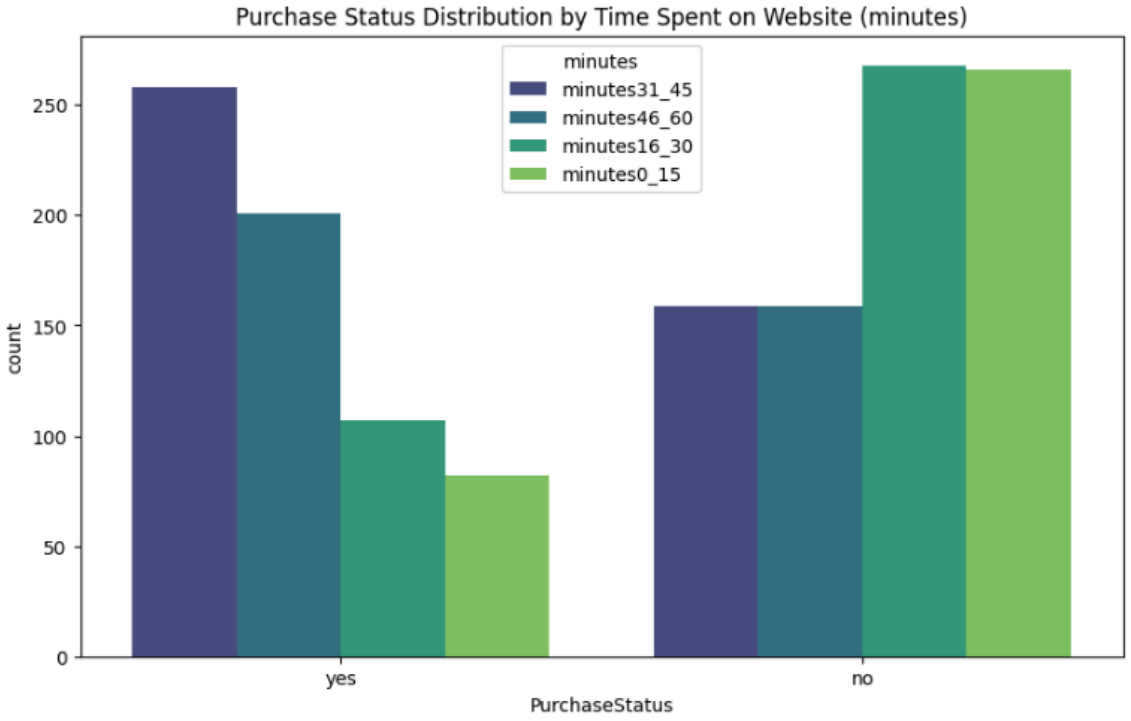
έκπτωση και όχι λόγο της εξυπηρέτησης και του after sales marketing κάτι που είναι λογικό όταν αναφερόμαστε σε πωλήσεις μέσο διαδικτύου όπου δεν υπάρχει επικοινωνία πελάτη με πωλητή.

```
6_10orders - Purchase Status Yes: 49.28%
6_10orders - Purchase Status No: 50.72%
1_5orders - Purchase Status Yes: 19.16%
1_5orders - Purchase Status No: 80.84%
11_15orders - Purchase Status Yes: 53.20%
11_15orders - Purchase Status No: 46.80%
16_20orders - Purchase Status Yes: 51.38%
16_20orders - Purchase Status No: 48.62%
zero_orders - Purchase Status Yes: 21.57%
zero_orders - Purchase Status No: 78.43%
```

Εικονα 80: count plot purchase status and number of previous orders

Και η τελευταία μεταβλητή είναι ο χρόνος που αφιέρωσε ο εκάστοτε πελάτης πριν την απόρριψη ή αγορά του προϊόντος. Όπως βλέπουμε από το διάγραμμα οι πελάτες που αφιέρωσαν 0 έως 15 λεπτά ολοκληρώνουν την αγορά σε ποσοστό 26% από εκεί και πέρα καθώς ο χρόνος αυξάνεται το ποσοστό αγοράς επίσης αυξάνεται.

```
minutes31_45 - Purchase Status Yes: 61.87%
minutes31_45 - Purchase Status No: 38.13%
minutes46_60 - Purchase Status Yes: 55.83%
minutes46_60 - Purchase Status No: 44.17%
minutes16_30 - Purchase Status Yes: 28.53%
minutes16_30 - Purchase Status No: 71.47%
minutes0_15 - Purchase Status Yes: 23.56%
minutes0_15 - Purchase Status No: 76.44%
```



Εικονα 81: count plot purchase status and time spent

Στην συνέχεια κάνουμε ανάλυση απλής γραμμικής παλινδρόμησης με κάποιες από τις παραπάνω μεταβλητές και υπολογίζουμε το συντελεστή συσχέτισης και το slope (συντελεστής β) του εκάστοτε μοντέλου. Η πρώτη γραμμική παλινδρόμηση είναι μεταξύ της μεταβλητής ηλικίας(ανεξάρτητη μεταβλητή) και εισοδήματος(εξαρτημένη μεταβλητή)

```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from scipy.stats import pearsonr
x = beh['Age']
y = beh['AnnualIncome']
slope, intercept = np.polyfit(x, y, 1)
correlation_coefficient, p_value = pearsonr(x, y)
print(f"Slope: {slope}")
print(f"Intercept: {intercept}")
print(f"Correlation Coefficient: {correlation_coefficient}")
plt.figure(figsize=(8, 6))
sns.regplot(x=x, y=y, data=beh, ci=None, line_kws={'color': 'red'})
plt.title('Linear Regression: Annual Income vs. Age')
plt.xlabel('Age')
plt.ylabel('Annual Income')
plt.show()
```

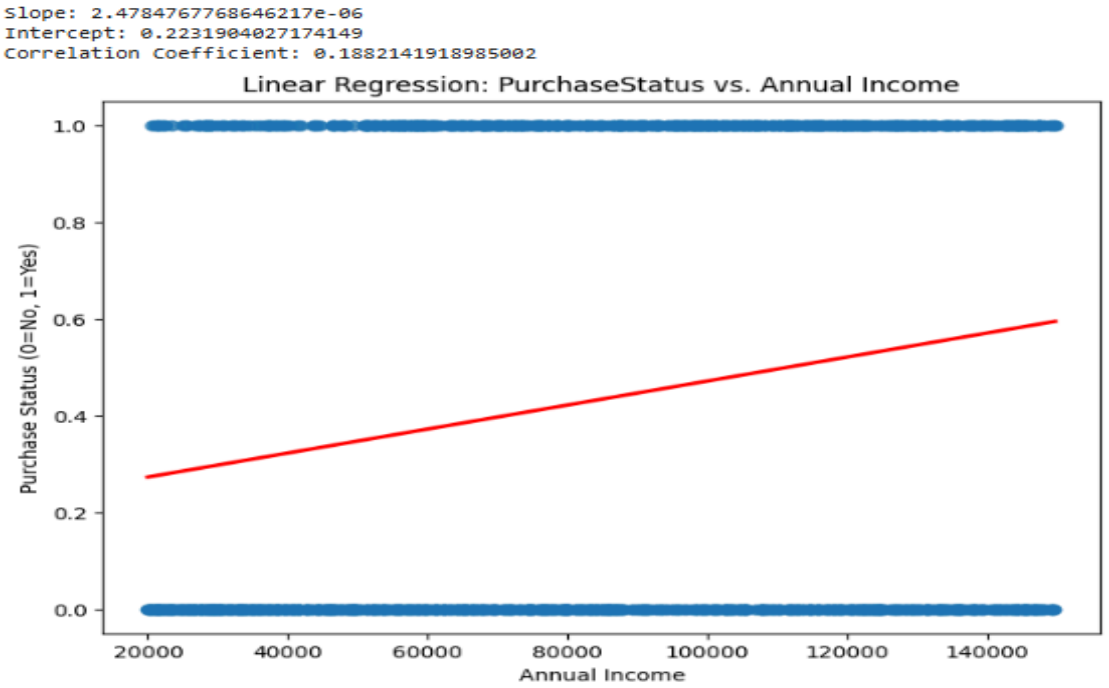
Εικονα 82 : γραμμική παλινδρόμηση age annual income

Όπως βλέπουμε δεν υπάρχει καμία συσχέτιση μεταξύ εισοδήματος και ηλικίας δηλαδή καθώς αυξάνεται ή ηλικία το εισόδημα κινείται ανεξάρτητα από την ηλικία , και αυτό επιβεβαιώνεται καθώς υπάρχουν άτομα ηλικίας 18 ετών όπου έχουν ετήσιο εισόδημα άνω των 100 χιλιάδων. Το παραπάνω αποτέλεσμα προκύπτει λόγο του ότι τα δεδομένα δεν είναι πραγματικά , υπό κανονικές συνθήκες καθώς ένα άτομο μεγαλώνει αυξάνεται και το εισόδημα του.

```
Slope: 39.81167718117771
Intercept: 82485.56012135265
Correlation Coefficient: 0.0164382847096486
```

Εικονα 83: διάγραμμα γραμμικής παλινδρόμησης income and age

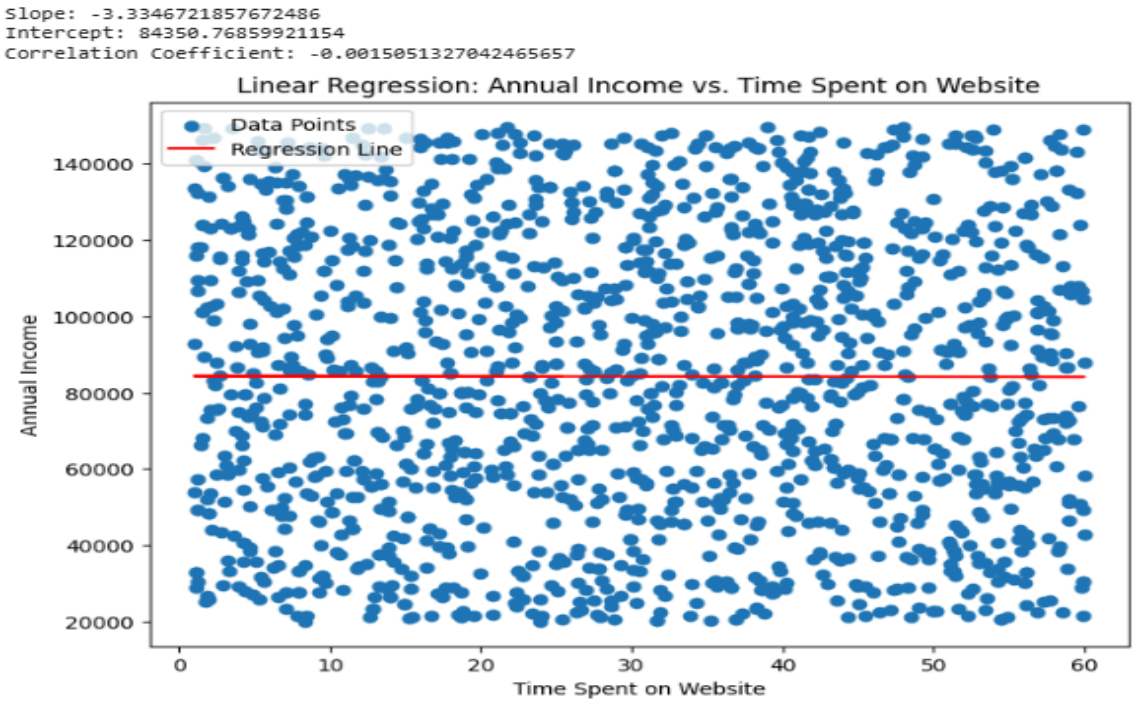
Η επόμενη παλινδρόμηση έχει ως εξαρτημένη μεταβλητή την απόφαση αγοράς ή μη του προϊόντος στο καλάθι σε σχέση με την ερμηνευτική μεταβλητή ετήσιο εισόδημα , από το μοντέλο προκύπτει ότι υπάρχει μια μικρή αλλά θετική συσχέτιση μεταξύ απόφασης αγοράς και εισοδήματος, δηλαδή καθώς το εισόδημα αυξάνεται , αυξάνεται και η πιθανότητα αγοράς των προϊόντων.



Εικονα 84: διάγραμμα γραμμικής παλινδρόμησης income and purchace status

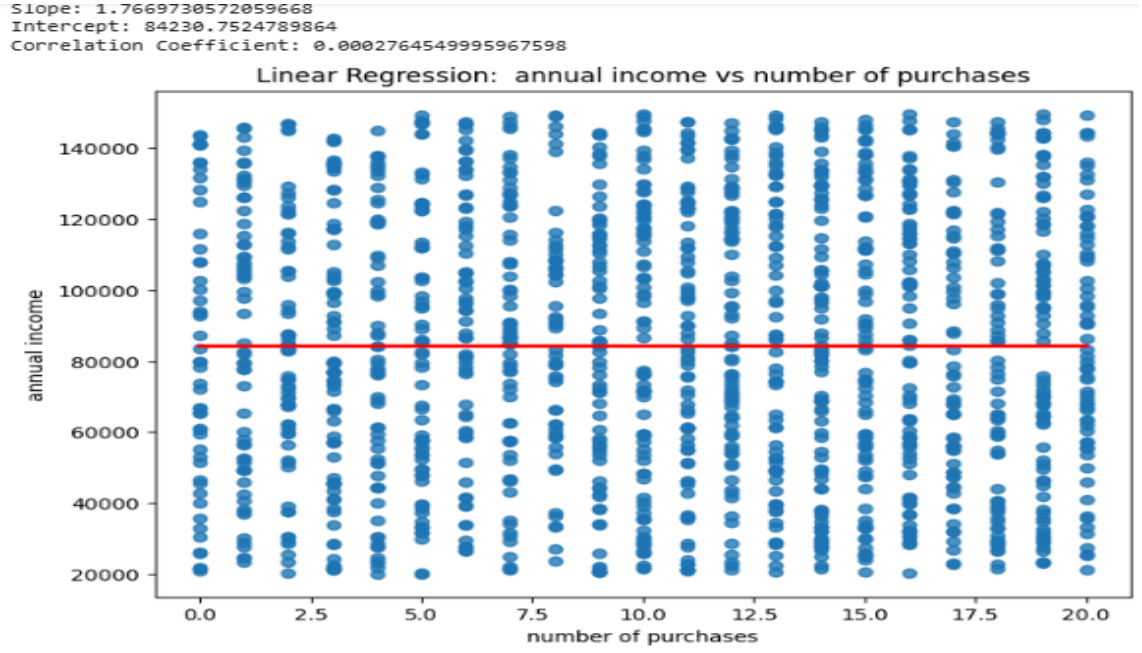
Στην συνέχεια πραγματοποιούμε μια γραμμική παλινδρόμηση μεταξύ εισοδήματος ως εξαρτημένη μεταβλητή και χρόνου που διέθεσε ο πελάτης στην ιστοσελίδα. Όπως παρατηρούμε οι δυο αυτές μεταβλητές είναι ανεξάρτητες δηλαδή κάποιος θα έλεγε ότι ίσως άτομα με υψηλό εισόδημα να έχουν πήγες παθητικού εισοδήματος που να τους επιτρέπουν. να έχουν περισσότερο ελεύθερο χρόνο , κάτι τέτοιο δεν επιβεβαιώνεται από τα

δεδομένα.



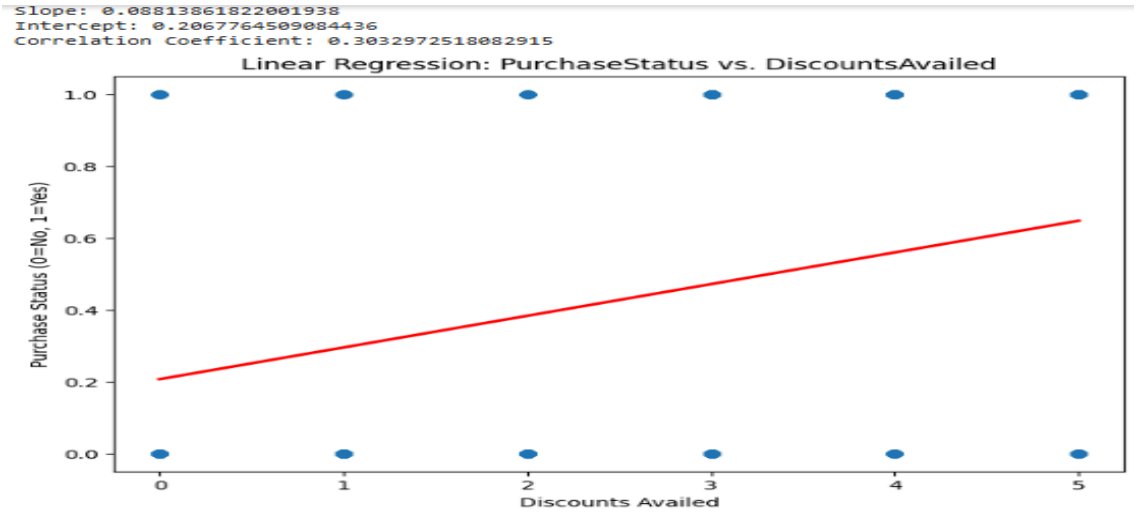
Εικονα 85: διάγραμμα γραμμικής παλινδρόμησης income and time spent

Στην συνέχεια αναλύουμε την σχέση μεταξύ ετήσιου εισοδήματος και τον αριθμό προηγουμένων παραγγελιών και σε σκύτη την περίπτωση δεν υπάρχει συσχέτιση μεταξύ των δυο μεταβλητών , για κάθε αριθμό προηγουμένων αγορών τα εισοδήματα των πελατών είναι μεταξύ 20 και 140 χιλιάδων δολαρίων.



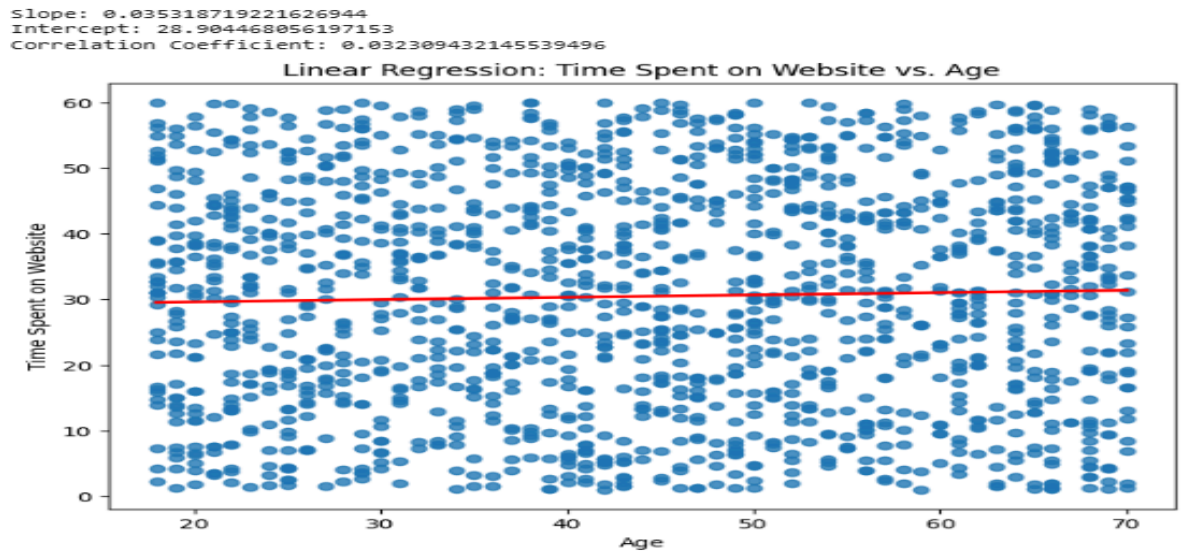
Εικονα 83: διάγραμμα γραμμικής παλινδρόμησης income and number of purchases

Στην συνέχεια έχουμε μια ενδιαφέρουσα ανάλυση μεταξύ πιθανότητας αγοράς ως εξαρτημένη μεταβλητή και ποσοστό έκπτωσης, όπως βλέπουμε υπάρχει θετική συσχέτιση δηλαδή καθώς αυξάνεται το ποσοστό της έκπτωσης αυξάνεται και η πιθανότητα αγοράς, αρά εάν το κατάστημα θέλει να αυξήσει την ζήτηση ίσως θα έπρεπε να αυξήσει το ποσοστό εκπτώσεων ή να μειώσει την τιμή.



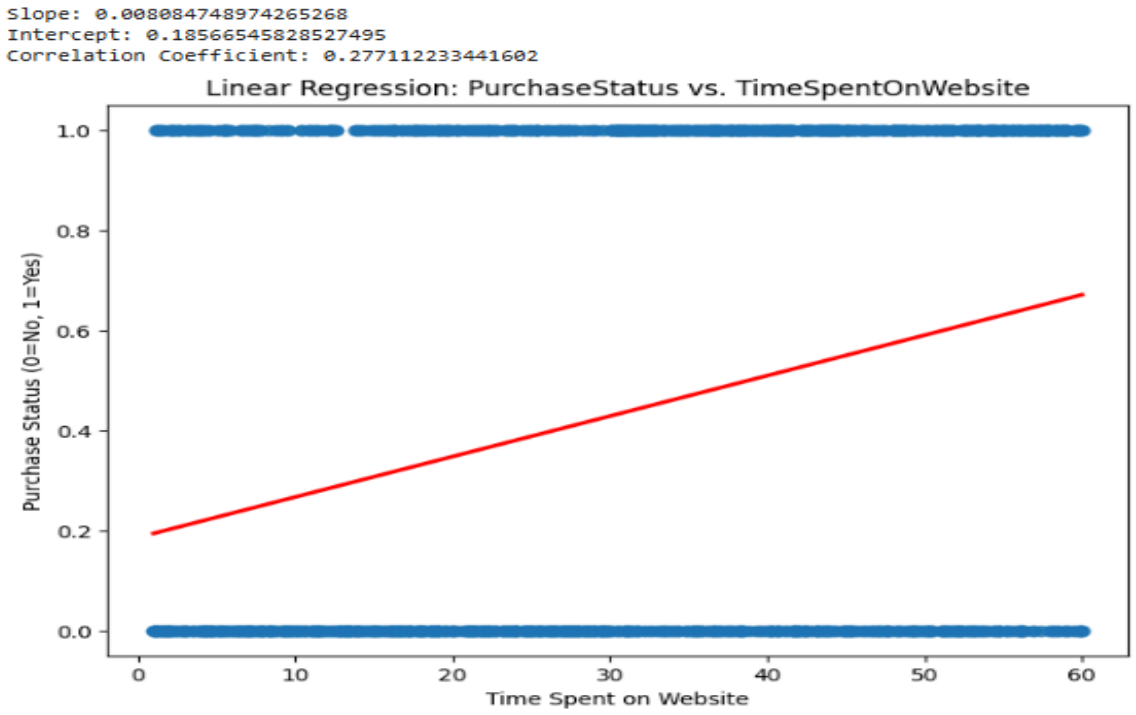
Εικονα 86: διάγραμμα γραμμικής παλινδρόμησης purchase status and discount

Όπως και στην παλινδρόμηση μεταξύ εισοδήματος και χρόνου στον ισότοπο έτσι και εδώ δεν υπάρχει καμία συσχέτιση μεταξύ χρόνου και ηλικίας , δηλαδή κάποιος θα περίμενε ότι ίσως άτομα μεγαλύτερης ηλικίας που έχουν συνταξιοδοτηθεί ή ακόμα και φοιτητές που δεν εργάζονται να είχαν περισσότερο ελεύθερο χρόνου να διαθέσουν για αγορές κάτι το οποίο δεν ισχύει με βάση τα δεδομένα.



Εικονα 83: διάγραμμα γραμμικής παλινδρόμησης time spent and age

Και τέλος βλέπουμε μια ξεκάθαρα θετική συσχέτιση μεταξύ πιθανότητας ολοκλήρωσης αγοράς σε σχέση με τον χρόνου που αφιερώνει ο πελάτης στην ιστοσελίδα του καταστήματος.



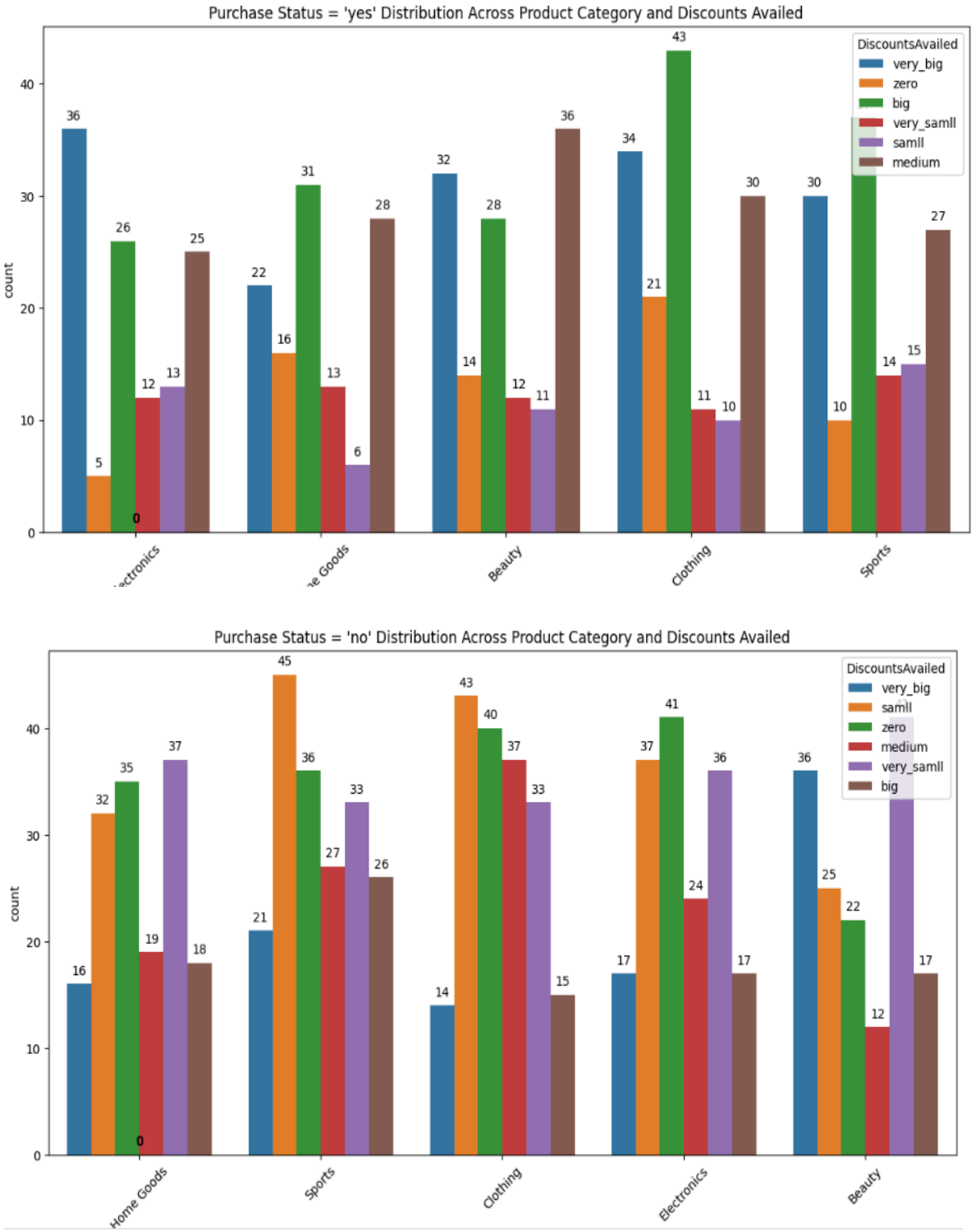
Εικονα 83: διάγραμμα γραμμικής παλινδρόμησης purchase status and time spent

Συνεχίζουμε την ανάλυση των δεδομένων συνδυάζοντας δυο μεταβλητές ταυτόχρονα σε σχέση με την πιθανότητα αγοράς του προϊόντος που είναι τοποθετημένο στο καλάθι. Ξεκινάμε την ανάλυση με τις μεταβλητές κατηγορία προϊόντος και διαθέσιμο ποσοστό έκπτωσης. Μερικές παρατηρήσεις σχετικά με τα διαγράμματα είναι ότι η πολύ μεγάλη έκπτωση(5%) αυξάνει το ποσοστό ολοκλήρωσης της αγοράς στα ηλεκτρονικά προϊόντα και στα προϊόντα ρουχισμού. Από την άλλη πλευρά η μηδενική έκπτωση έχει ως αποτέλεσμα το ποσοστό αγοράς ηλεκτρονικών ειδών να είναι ίσο με 10% το ίδιο χαμηλό ποσοστό αγοράς δημιουργείται και στα αθλητικά προϊόντα ενώ αναφορικά με την μεσαία(3%) και μεγάλη(4%) έκπτωση αυτές αυξάνουν την πιθανότητα αγοράς όλων των προϊόντων, σε σύγκρισή με την χαμηλή(2%) και πολύ χαμηλή(1%) έκπτωση.

```
plt.figure(figsize=(14, 7))
ax = sns.countplot(x='ProductCategory', hue='DiscountsAvailed', data=beh[beh['PurchaseStatus'] == 'yes'])
plt.title("Purchase Status = 'yes' Distribution Across Product Category and Discounts Availed")
plt.xticks(rotation=45)


for p in ax.patches:
    ax.annotate(f'{int(p.get_height())}', (p.get_x() + p.get_width() / 2., p.get_height()),
                ha='center', va='center', xytext=(0, 10), textcoords='offset points')

plt.show()
```

```
purchase_percentages = beh.groupby(['ProductCategory', 'DiscountsAvailed'])['PurchaseStatus'].value_counts(normalize=True).unstack() * 100
print(purchase_percentages)
purchase_percentages.plot(kind='bar', figsize=(14, 7))
plt.title('Purchase Status Percentage by Product Category and Discount')
plt.xlabel('Product Category and Discounts Availed')
plt.ylabel('Percentage of Purchase Status')
plt.xticks(rotation=45)
plt.legend(title='Purchase Status')
plt.tight_layout()
plt.show()
```

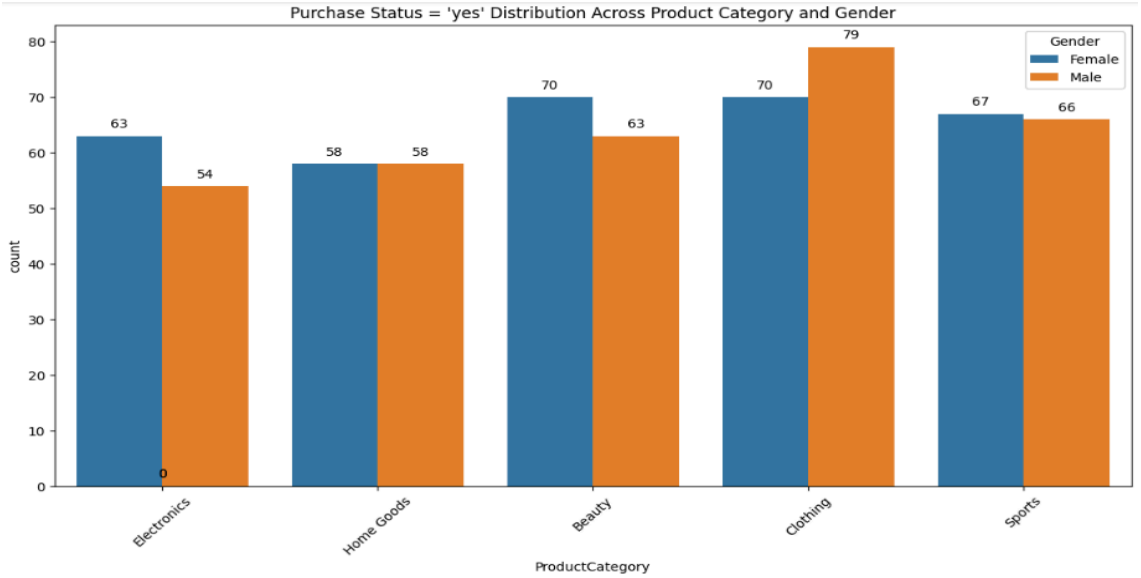
PurchaseStatus		no	yes
ProductCategory	DiscountsAvailed		
Beauty	big	37.777778	62.222222
	medium	25.000000	75.000000
	samll	69.444444	30.555556
	very_big	52.941176	47.058824
	very_samll	77.358491	22.641509
	zero	61.111111	38.888889
Clothing	big	25.862069	74.137931
	medium	55.223881	44.776119
	samll	81.132075	18.867925
	very_big	29.166667	70.833333
	very_samll	75.000000	25.000000
	zero	65.573770	34.426230
Electronics	big	39.534884	60.465116
	medium	48.979592	51.020408
	samll	74.000000	26.000000
	very_big	32.075472	67.924528
	very_samll	75.000000	25.000000
	zero	89.130435	10.869565
Home Goods	big	36.734694	63.265306
	medium	40.425532	59.574468
	samll	84.210526	15.789474
	very_big	42.105263	57.894737
	very_samll	74.000000	26.000000
	zero	68.627451	31.372549
Sports	big	41.269841	58.730159
	medium	50.000000	50.000000
	samll	75.000000	25.000000
	very_big	41.176471	58.823529
	very_samll	70.212766	29.787234
	zero	78.260870	21.739130

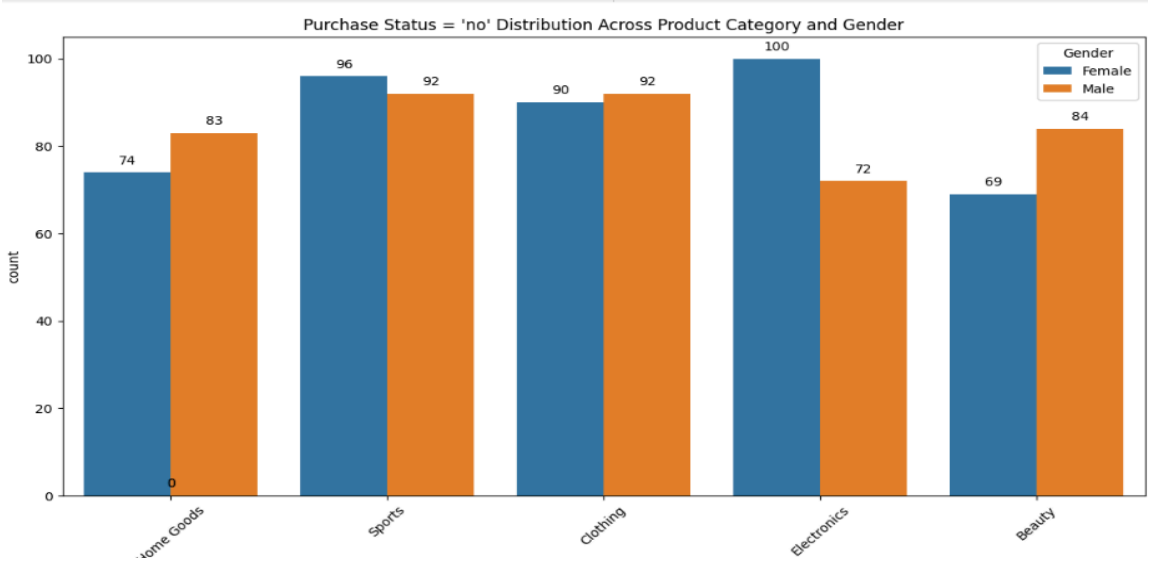


Εικονα 84: count plot purchase status ,product category and discount

Η επόμενη ανάλυση είναι μεταξύ των διάφορων κατηγοριών προϊόντων και του φύλου των πελατών σχετικά με την απόφαση αγοράς ή όχι των προϊόντων. Τα συμπεράσματα που εξάγονται είναι ότι οι γυναίκες πελάτες του καταστήματος είναι πιο πιθανό να ολοκληρώσουν την αγορά προϊόντων ομορφιάς με ποσοστό 50% ενώ η κατηγορία με το

μεγαλύτερο ποσοστό μη ολοκλήρωσης αγοράς είναι τα προϊόντα τεχνολογίας, από την άλλη οι άντρες είναι πιο πιθανό να αγοράσουν προϊόντα ρουχισμού ενώ όλες οι άλλες κατηγορίες έχουν πιθανότητα περίπου 40% να ολοκληρωθεί η αγορά τους.



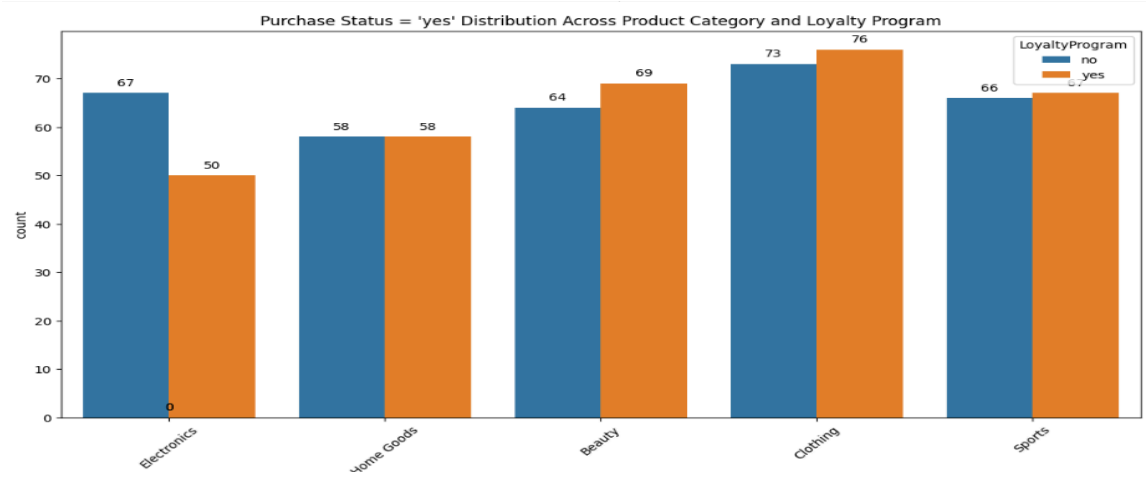


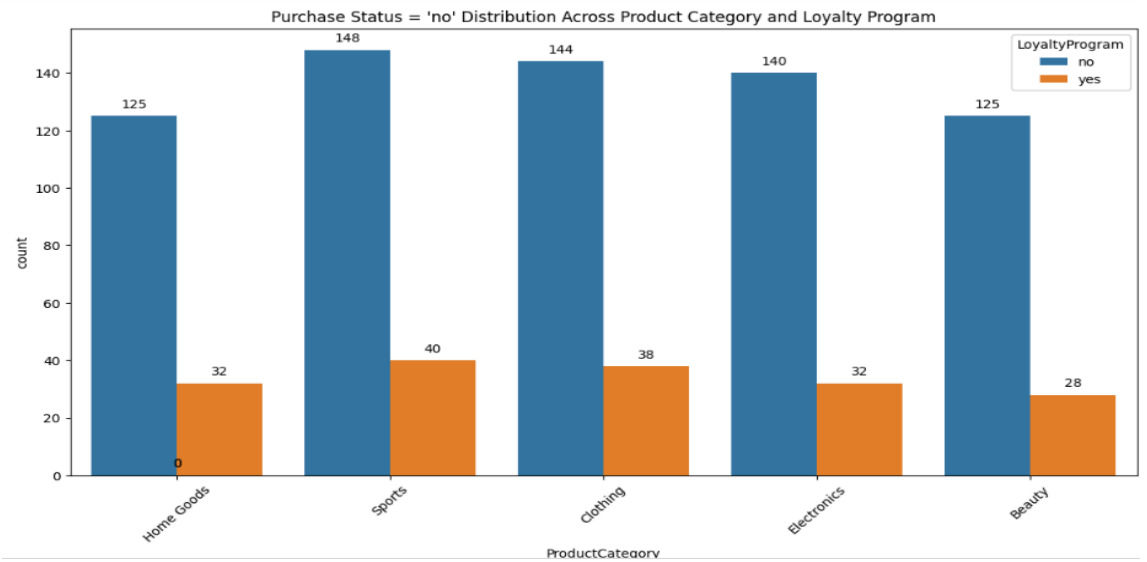
PurchaseStatus		no	yes
Gender	ProductCategory		
Female	Beauty	49.64	50.36
	Clothing	56.25	43.75
	Electronics	61.35	38.65
	Home Goods	56.06	43.94
	Sports	58.90	41.10
Male	Beauty	57.14	42.86
	Clothing	53.80	46.20
	Electronics	57.14	42.86
	Home Goods	58.87	41.13
	Sports	58.23	41.77

Εικονα 85: count plot purchase status ,product category and gende

Η ιδία ανάλυση συνεχίζεται με την σχέση της μεταβλητής κατηγορίας προϊόντων και το αν ο πελάτης έχει κάρτα μέλους η όχι. Όπως έχουμε ήδη αναφέρει η πιθανότητα ολοκλήρωσης της αγοράς αυξάνεται δραματικά με την κατοχή κάρτας μέλους σε σχέση με την μη κατοχή της. Τώρα με το παρακάτω διάγραμμα μπορούμε να δούμε πιο συγκεκριμένα πια προϊόντα κατά πασά πιθανότητα θα αγοραστούν από τα μέλη. Βλέπουμε ότι τα προϊόντα που επιλέγουν οι κάτοχοι κάρτας είναι τα προϊόντα ομορφιάς με πιθανότητα ολοκλήρωσης της αγοράς ίση με 70% στην συνέχεια ακολουθούν τα προϊόντα ρουχισμού και τα είδη σπιτιού.

PurchaseStatus		no	yes
LoyaltyProgram	ProductCategory		
no	Beauty	66.14	33.86
	Clothing	66.36	33.64
	Electronics	67.63	32.37
	Home Goods	68.31	31.69
	Sports	69.16	30.84
yes	Beauty	28.87	71.13
	Clothing	33.33	66.67
	Electronics	39.02	60.98
	Home Goods	35.56	64.44
	Sports	37.38	62.62

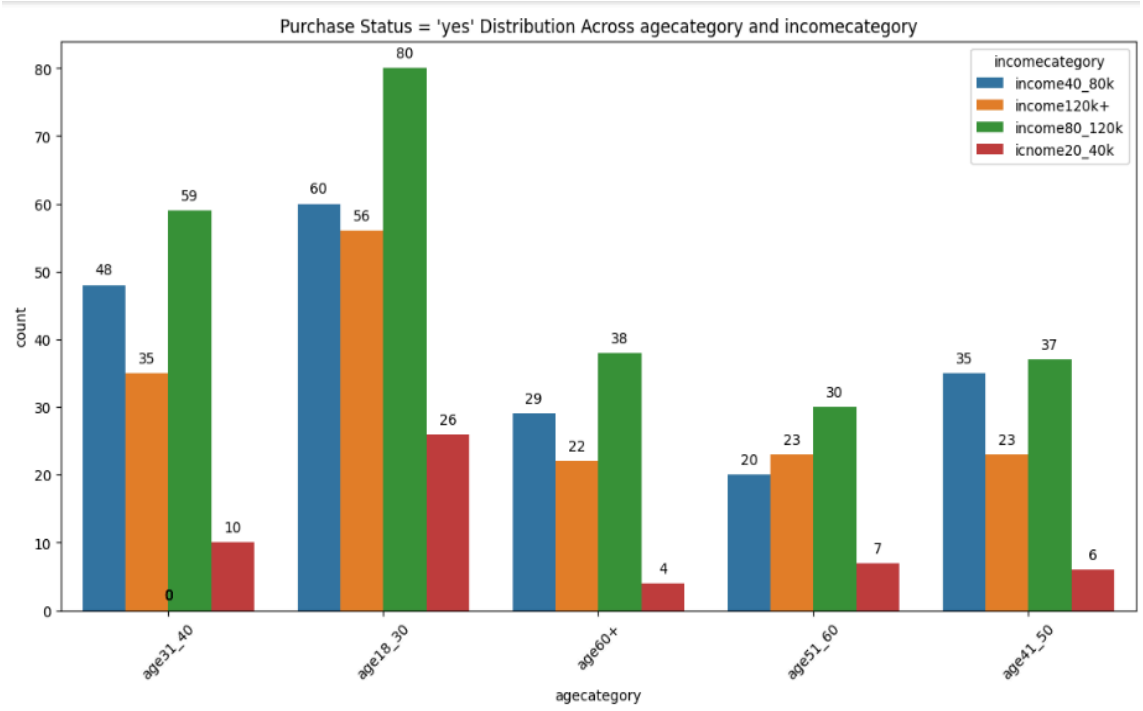


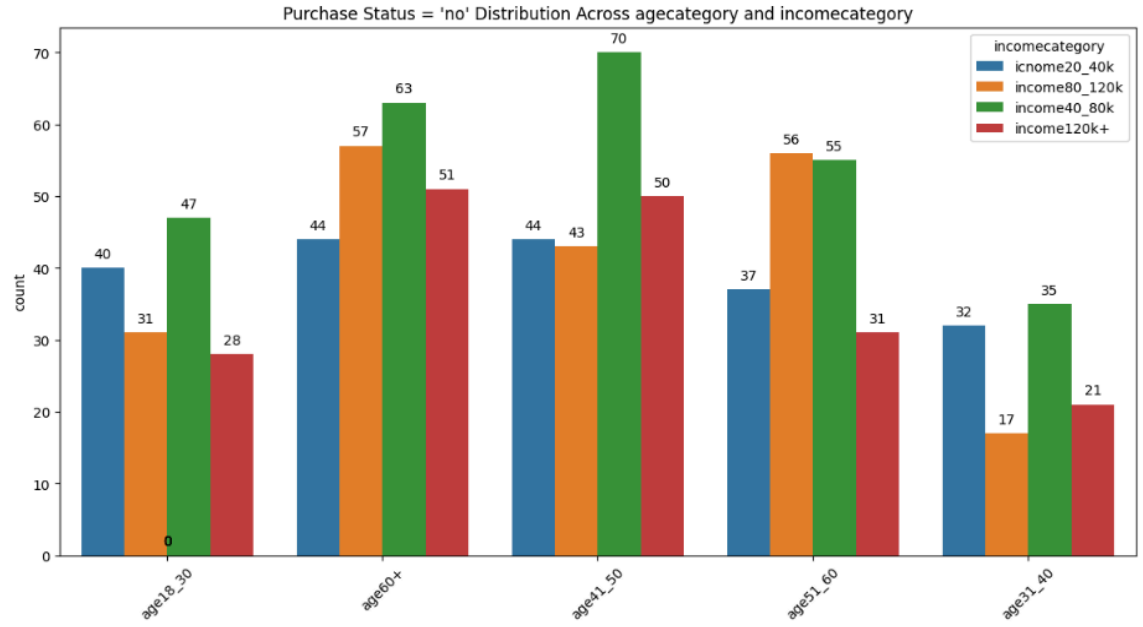


Εικονα 86: count plot purchase status ,product category and loyalty program

Η επόμενη σύγκριση είναι μεταξύ ηλικιακής ομάδας και ομάδας εισοδήματος , γενικά αυτό που παρατηρούμε με βάση τα δεδομένα είναι ότι καθώς αυξάνεται η ηλικία η πιθανότητα ολοκλήρωσης της αγοράς μειώνεται ανεξάρτητος του ετήσιου εισοδήματος. Πιο αναλυτικά το μεγαλύτερο ποσοστό ολοκλήρωσης αγοράς υπάρχει στην ηλικιακή ομάδα 41 έως 50 με εισόδημα 80 με 120 χιλιάδες ευρώ ενώ το μικρότερο ποσοστό υπάρχει στην ηλικιακή ομάδα των ατόμων άνω των 60 ετών με εισόδημα μεταξύ των 20 και 40

χιλιάδων, παρόλα αυτά συνολικά η ηλικιακή ομάδα με της περισσότερές αγορές ανεξάρτητος εισοδήματος είναι η ομάδα των 18 έως 30 ετών.

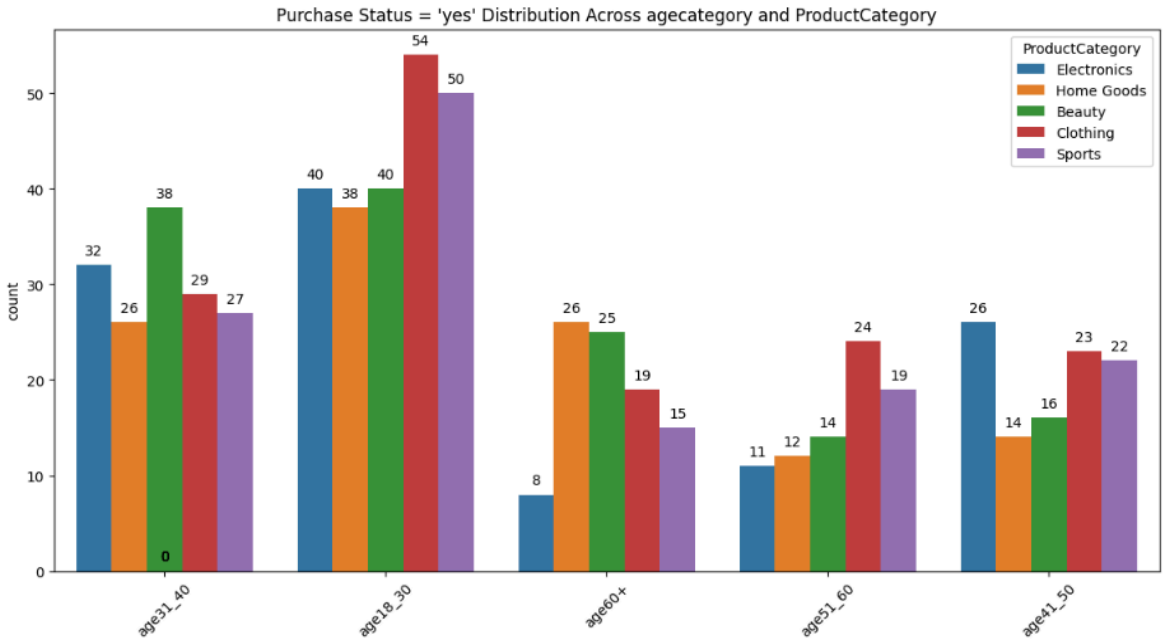


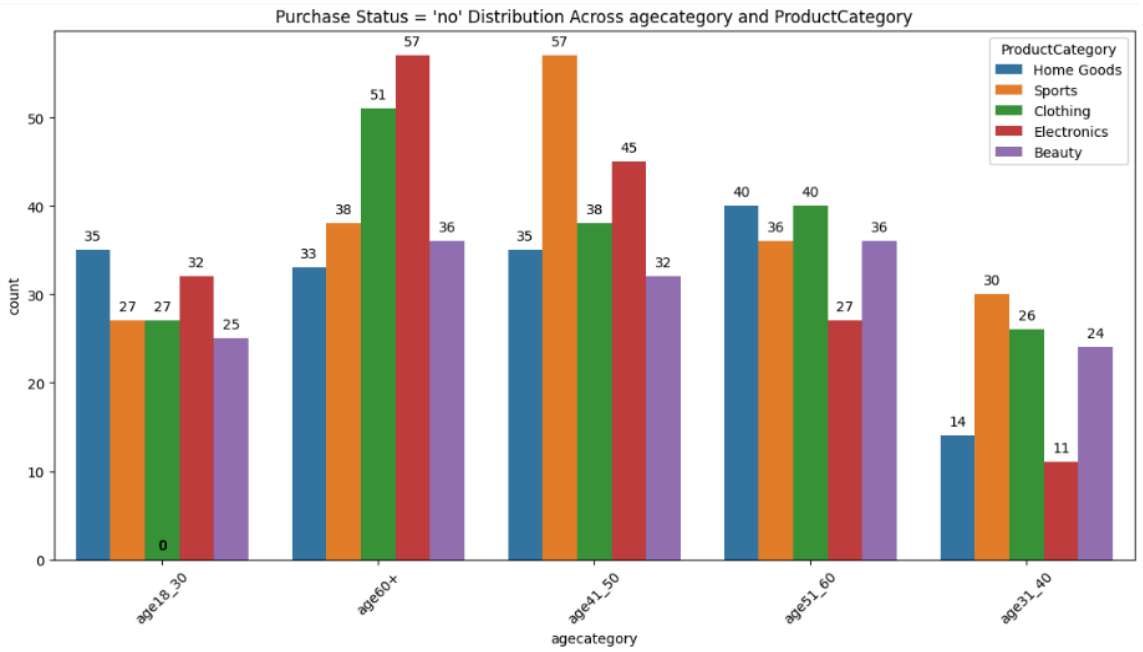


PurchaseStatus		no	yes
agecategory	incomecategory		
age18_30	icnome20_40k	60.61	39.39
	income120k+	33.33	66.67
	income40_80k	43.93	56.07
	income80_120k	27.93	72.07
age31_40	icnome20_40k	76.19	23.81
	income120k+	37.50	62.50
	income40_80k	42.17	57.83
	income80_120k	22.37	77.63
age41_50	icnome20_40k	88.00	12.00
	income120k+	68.49	31.51
	income40_80k	66.67	33.33
	income80_120k	53.75	46.25
age51_60	icnome20_40k	84.09	15.91
	income120k+	57.41	42.59
	income40_80k	73.33	26.67
	income80_120k	65.12	34.88
age60+	icnome20_40k	91.67	8.33
	income120k+	69.86	30.14
	income40_80k	68.48	31.52
	income80_120k	60.00	40.00

Εικονα 87: count plot purchase status ,age category and income

Ακόλουθος βλέπουμε ποια κατηγορία προϊόντων επιλεγεί η κάθε ηλικιακή ομάδα. Συγκεκριμένα οι πελάτες από 18 έως 30 όταν βάλουν στο καλάθι τους προϊόντα ρουχισμού υπάρχει πιθανότητα να τα αγοράσουν της τάξης του 66% , οι πελάτες από 31 έως 40 ολοκληρώνουν με ποσοστό 74% τις αγορές ηλεκτρονικών ειδών . το ποσοστό αγοράς στις επόμενες ηλικιακές ομάδες όπως έχουμε δει μειώνεται σε μεγάλο βαθμό παρόλα αυτά οι πελάτες από 41 έως 50 ολοκληρώνουν την αγορά ρούχων σε ποσοστό 37% η ιδία κατηγορία επιλέγεται και από την ηλικιακή ομάδα των 51 έως 60 και τέλος τα άτομα άνω των 60 ολοκληρώνουν την αγορά κύριος προϊόντων σπιτιού σε ποσοστό 41%

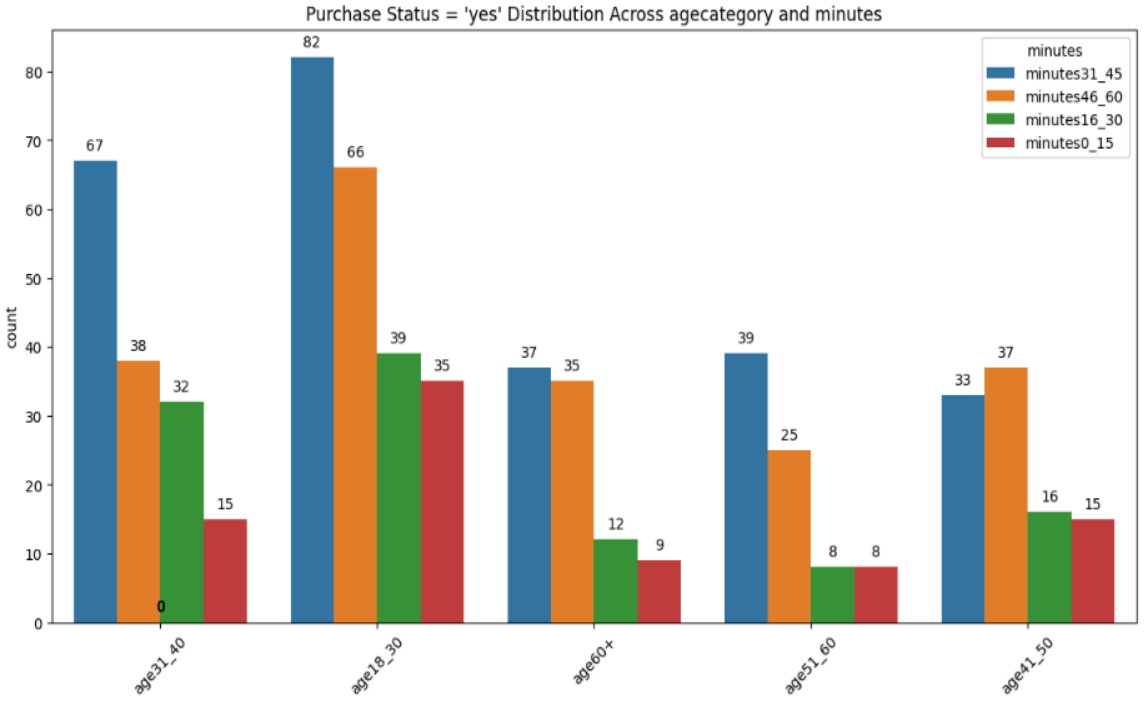

Purchase Status = 'yes' Distribution Across agecategory and ProductCategory
ProductCategory
Electronics
Home Goods
Beauty
Clothing
Sports
count
0
10
20
30
40
50
32
26
38
0
29
27
40
38
40
54
50
8
26
25
19
15
11
12
14
24
19
26
14
16
23
22
age31_40
age18_30
age60+
age51_60
age41_50

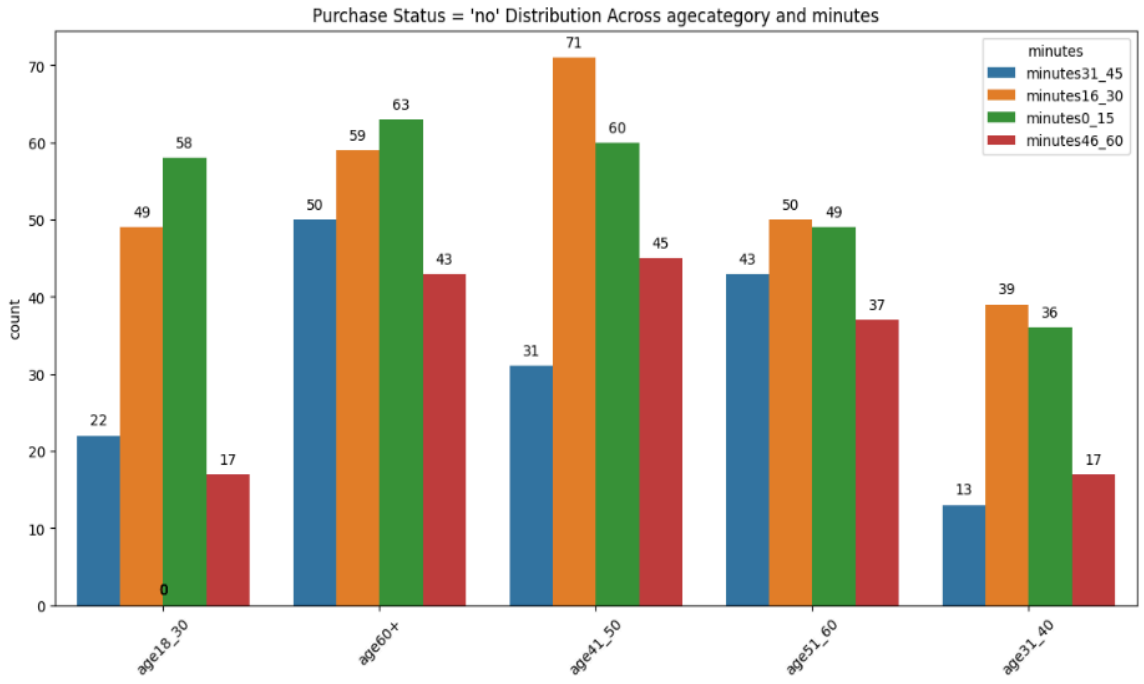


PurchaseStatus		no	yes
agecategory	ProductCategory		
age18_30	Beauty	38.46	61.54
	Clothing	33.33	66.67
	Electronics	44.44	55.56
	Home Goods	47.95	52.05
	Sports	35.06	64.94
age31_40	Beauty	38.71	61.29
	Clothing	47.27	52.73
	Electronics	25.58	74.42
	Home Goods	35.00	65.00
	Sports	52.63	47.37
age41_50	Beauty	66.67	33.33
	Clothing	62.30	37.70
	Electronics	63.38	36.62
	Home Goods	71.43	28.57
	Sports	72.15	27.85
age51_60	Beauty	72.00	28.00
	Clothing	62.50	37.50
	Electronics	71.05	28.95
	Home Goods	76.92	23.08
	Sports	65.45	34.55
age60+	Beauty	59.02	40.98
	Clothing	72.86	27.14
	Electronics	87.69	12.31
	Home Goods	55.93	44.07
	Sports	71.70	28.30

Εικονα 88: count plot purchase status ,age category and product category

Στη συνέχεια βλέπουμε την σχέση μεταξύ ηλικιακής ομάδας και χρόνου σε λεπτά που αφιέρωσαν οι πελάτες εντός της ιστοσελίδας του καταστήματος, όπως έχουμε είδη αναφέρει όσο περισσότερο ψάχνει ο πελάτης μέσα στην ιστοσελίδα τόσο πιο πιθανό είναι να αγοράσει τα προϊόντα ανεξάρτητος ηλικίας τα δυο μεγαλύτερα ποσοστά ολοκλήρωσης της αγοράς τα έχουν τα άτομα ηλικίας 31 έως 40 που αφιερώνουν 30 έως 45 λεπτά στο κατάστημα (ποσοστό αγοράς 84%) και τω δεύτερο μεγαλύτερο οι πελάτες ηλικίας 18 έως 30 που αφιερώνουν 45 έως 60 λεπτά με ποσοστό αγοράς 79,5%.

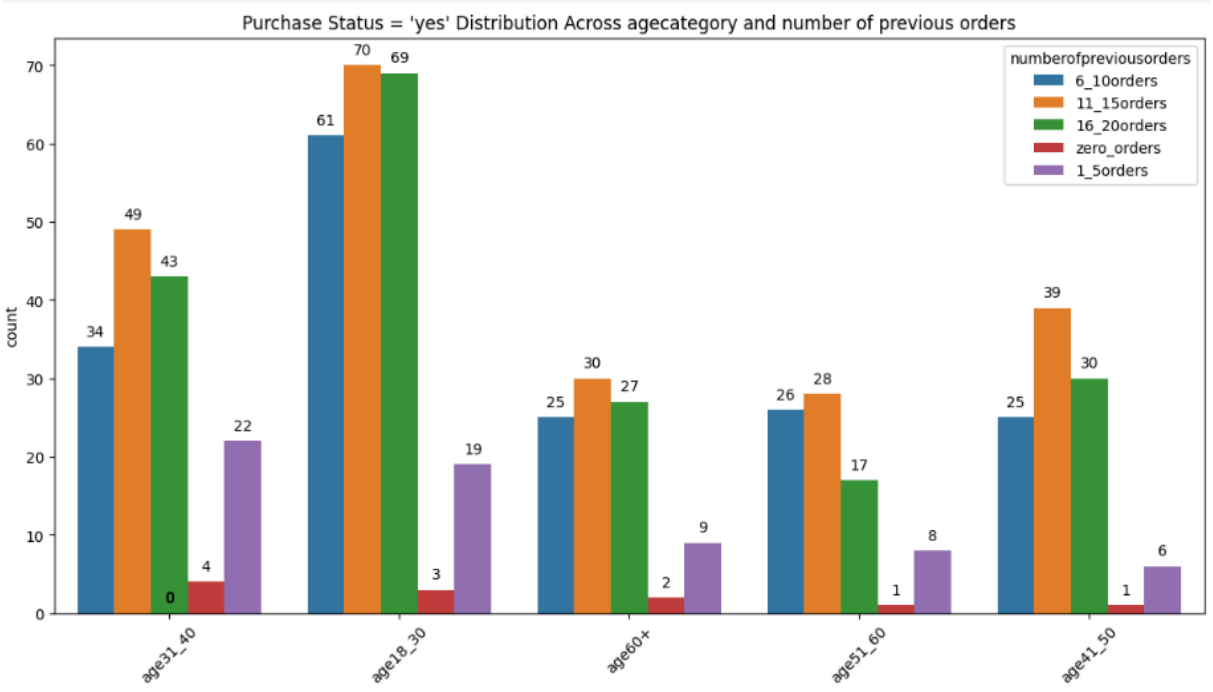


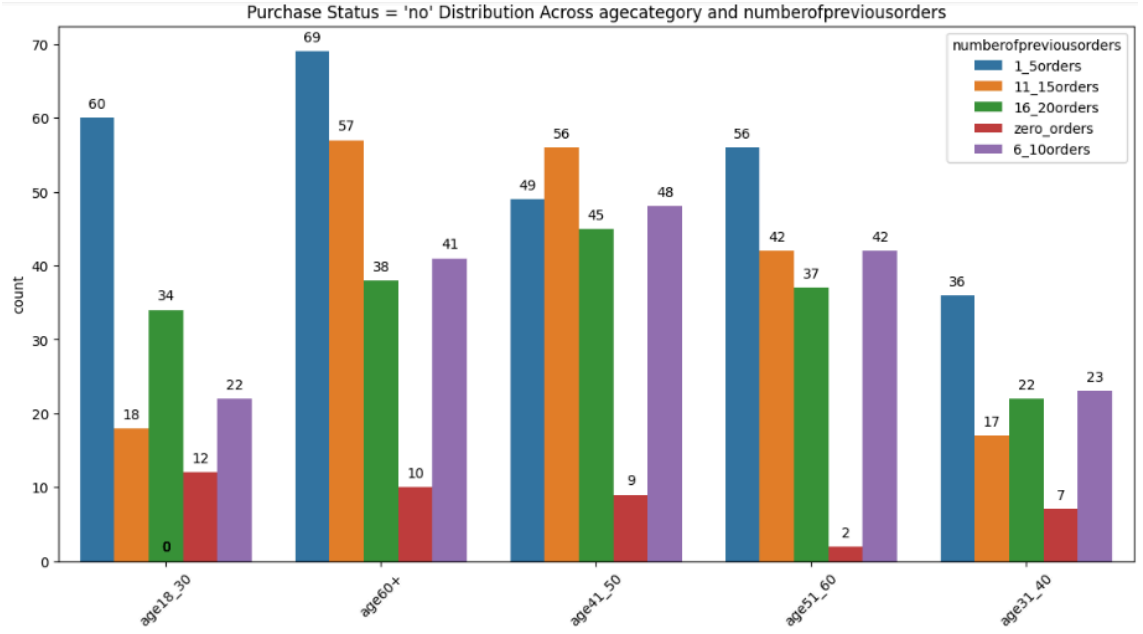


PurchaseStatus		no	yes
agecategory	minutes		
age18_30	minutes0_15	62.37	37.63
	minutes16_30	55.68	44.32
	minutes31_45	21.15	78.85
	minutes46_60	20.48	79.52
age31_40	minutes0_15	70.59	29.41
	minutes16_30	54.93	45.07
	minutes31_45	16.25	83.75
	minutes46_60	30.91	69.09
age41_50	minutes0_15	80.00	20.00
	minutes16_30	81.61	18.39
	minutes31_45	48.44	51.56
	minutes46_60	54.88	45.12
age51_60	minutes0_15	85.96	14.04
	minutes16_30	86.21	13.79
	minutes31_45	52.44	47.56
	minutes46_60	59.68	40.32
age60+	minutes0_15	87.50	12.50
	minutes16_30	83.10	16.90
	minutes31_45	57.47	42.53
	minutes46_60	55.13	44.87

Εικονα 89: count plot purchase status ,age category and time spent

Μετέπειτα βλέπουμε την σχέση μεταξύ ηλικιακής ομάδας και τον αριθμό των προηγουμένων αγορών. Ενδιαφέρον σε αυτή την ανάλυση έχει το γεγονός ότι την μεγαλύτερη πιθανότητα ολοκλήρωσης της πρώτης αγοράς(άτομα που δεν έχουν ξανά παραγγείλει) από το κατάστημα έχουν τα άτομα ηλικίας 31 έως 40 .Από τους πελάτες που έχουν κάνει πάνω από 16 αγορές δηλαδή άτομα που έχουν εμπιστευτεί πάρα πολλές φορές το κατάστημα , οι πιο πιθανοί να ολοκληρώσουν ακόμα μια αγορά είναι τα άτομα ηλικίας 18 έως 30.

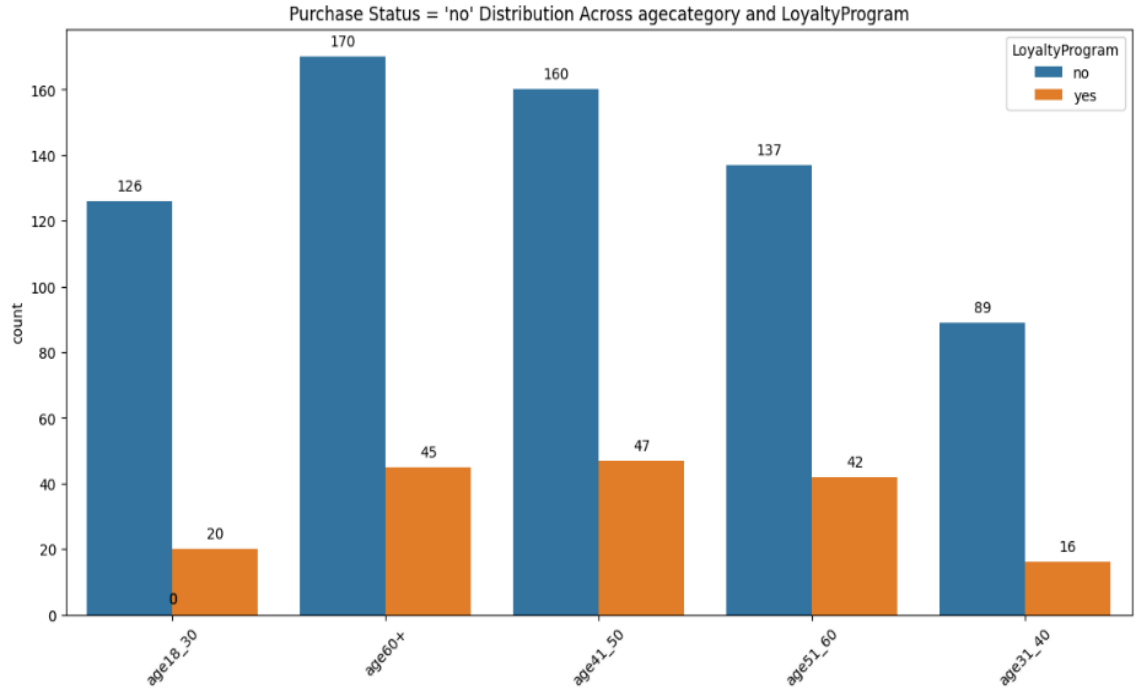

Purchase Status = 'yes' Distribution Across agecategory and number of previous orders
numberofpreviousorders
6_10orders
11_15orders
16_20orders
zero_orders
1_5orders
count
age31_40
age18_30
age60+
age51_60
age41_50

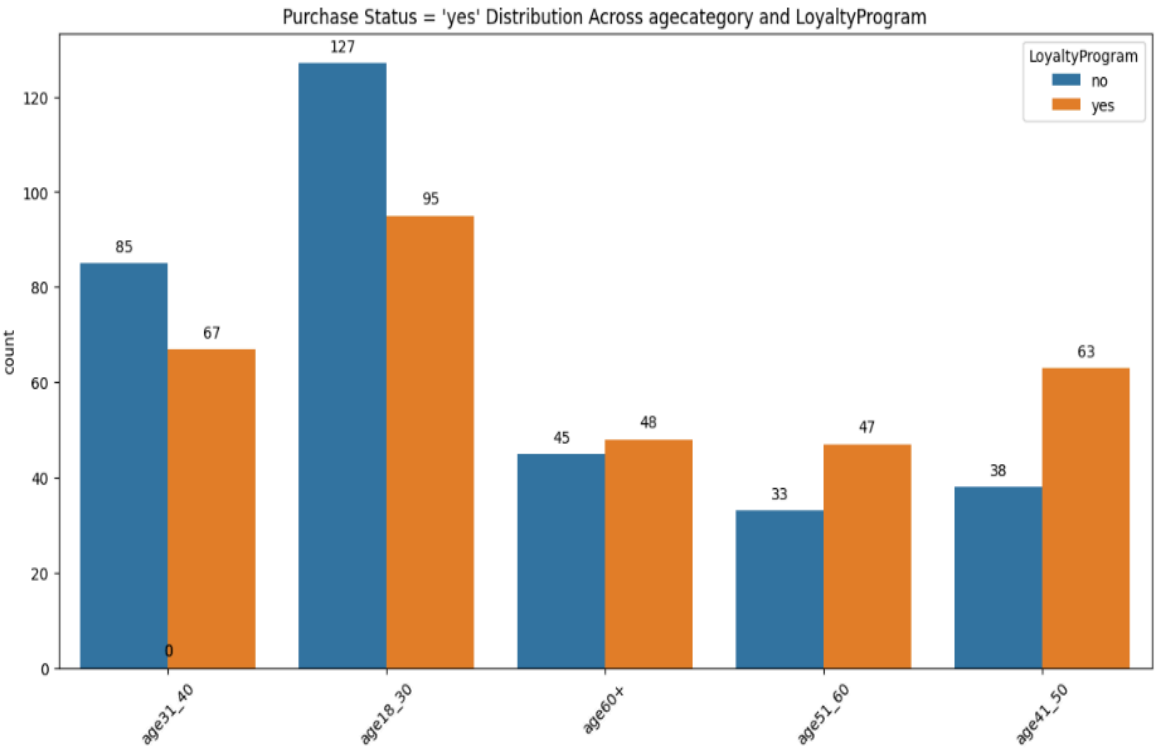

Purchase Status = 'no' Distribution Across agecategory and numberofpreviousorders
numberofpreviousorders
1_5orders
11_15orders
16_20orders
zero_orders
6_10orders
count
age18_30
age60+
age41_50
age51_60
age31_40

PurchaseStatus		no	yes
agecategory	numberofpreviousorders		
age18_30	11_15orders	20.45	79.55
	16_20orders	33.01	66.99
	1_5orders	75.95	24.05
	6_10orders	26.51	73.49
	zero_orders	80.00	20.00
age31_40	11_15orders	25.76	74.24
	16_20orders	33.85	66.15
	1_5orders	62.07	37.93
	6_10orders	40.35	59.65
	zero_orders	63.64	36.36
age41_50	11_15orders	58.95	41.05
	16_20orders	60.00	40.00
	1_5orders	89.09	10.91
	6_10orders	65.75	34.25
	zero_orders	90.00	10.00
age51_60	11_15orders	60.00	40.00
	16_20orders	68.52	31.48
	1_5orders	87.50	12.50
	6_10orders	61.76	38.24
	zero_orders	66.67	33.33
age60+	11_15orders	65.52	34.48
	16_20orders	58.46	41.54
	1_5orders	88.46	11.54
	6_10orders	62.12	37.88
	zero_orders	83.33	16.67

Εικονα 90: count plot purchase status ,age category and number of previous orders

Προχωράμε στην ανάλυση μεταξύ ηλικιακής ομάδας και κάρτας μέλους. Όπως μπορούμε να παρατηρήσουμε τα άτομα ηλικίας 18 με 30 ακόμα και αν δεν έχουν κάρτα μέλους η πιθανότητα ολοκλήρωσης της αγοράς είναι 50% το ίδιο υψηλό ποσοστό έχουν και οι πελάτες ηλικίας 31 έως 40 χωρίς κάρτα μέλους, βέβαια τα ποσοστά πέφτουν δραματικά στα άτομα μεγαλύτερης ηλικίας όπου το ποσοστό ολοκλήρωσης της αγοράς χωρίς κάρτα μέλους είναι 20%.Επισης η κάρτα μέλους δεν παίζει πολύ σημαντικό ρολό στην ολοκλήρωση της αγοράς για τα άτομα ηλικίας 41 ετών και άνω , σε αντίθεση με τα άτομα κάτω των 41 με κάρτα όπου το ποσοστό είναι περίπου 80%.

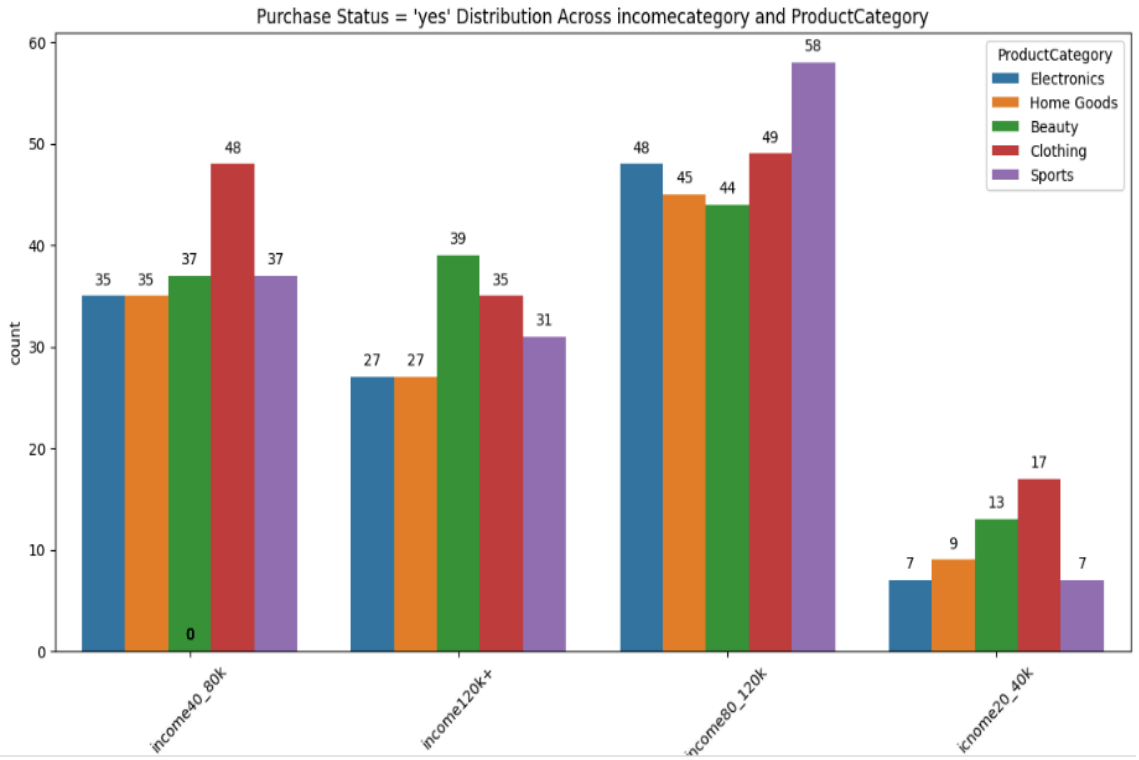

Purchase Status = 'no' Distribution Across agecategory and LoyaltyProgram
LoyaltyProgram
no
yes
count
0
20
40
60
80
100
120
140
160
126
20
0
170
45
160
47
137
42
89
16
age18_30
age60+
age41_50
age51_60
age31_40

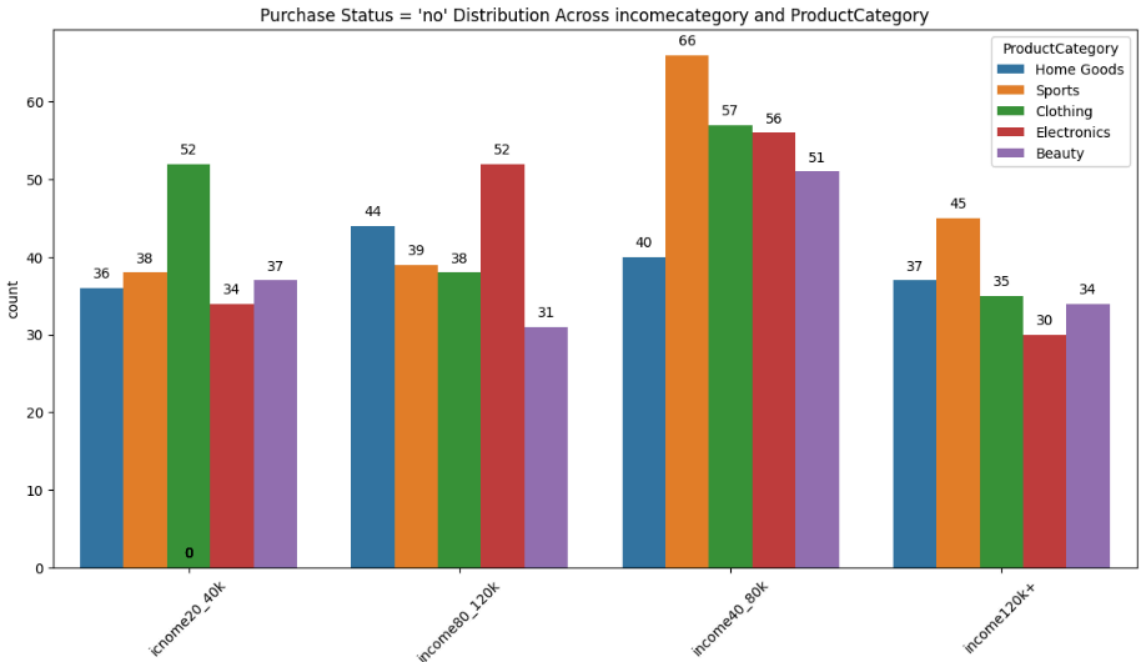

Purchase Status = 'yes' Distribution Across agecategory and LoyaltyProgram
LoyaltyProgram
no
yes
count
0
20
40
60
80
100
120
85
67
0
127
95
45
48
33
47
38
63
age31_40
age18_30
age60+
age51_60
age41_50

PurchaseStatus		no	yes
agecategory	LoyaltyProgram		
age18_30	no	49.80	50.20
	yes	17.39	82.61
age31_40	no	51.15	48.85
	yes	19.28	80.72
age41_50	no	80.81	19.19
	yes	42.73	57.27
age51_60	no	80.59	19.41
	yes	47.19	52.81
age60+	no	79.07	20.93
	yes	48.39	51.61

Εικονα 91: count plot purchase status ,age category and loyalty program

Μια επίσης ενδιαφέρουσα ανάλυση είναι αυτή για την αναγνώριση των αγοραστικών συνήθειων των ατόμων ανάλογα με το ετήσιο εισόδημα τους. Τα άτομα με εισόδημα μεταξύ 20 και 40 χιλιάδων δεν ολοκληρώνουν τις αγορές τους , το μεγαλύτερο ποσοστό αγοράς αυτής της εισοδηματικής κατηγορίας το έχουν τα προϊόντα ομορφιάς με 26%. Τα άτομα με εισόδημα 40 με 80 χιλιάδες ολοκληρώνουν κατά κύριο λόγο τις αγορές ειδών σπιτιού και τέλος τα άτομα με εισόδημα άνω των 120 χιλιάδων αν βάλουν στο καλάθι τους προϊόντα ομορφιάς τα αγοράζουν με ποσοστό περίπου 60%.

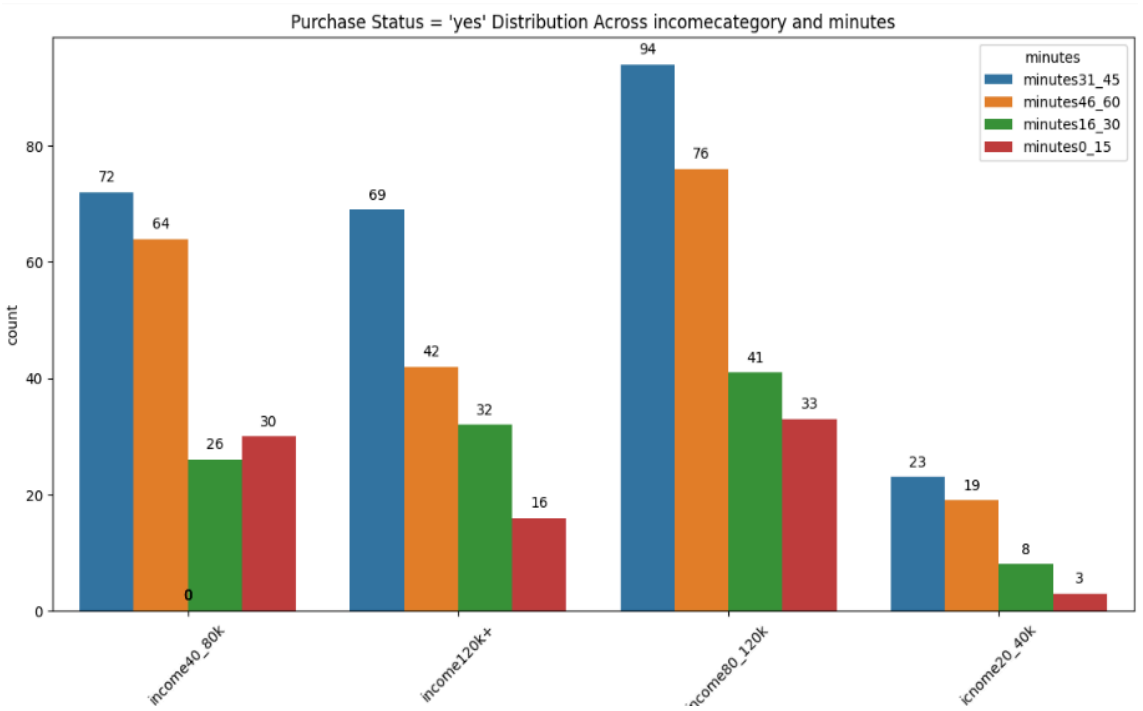


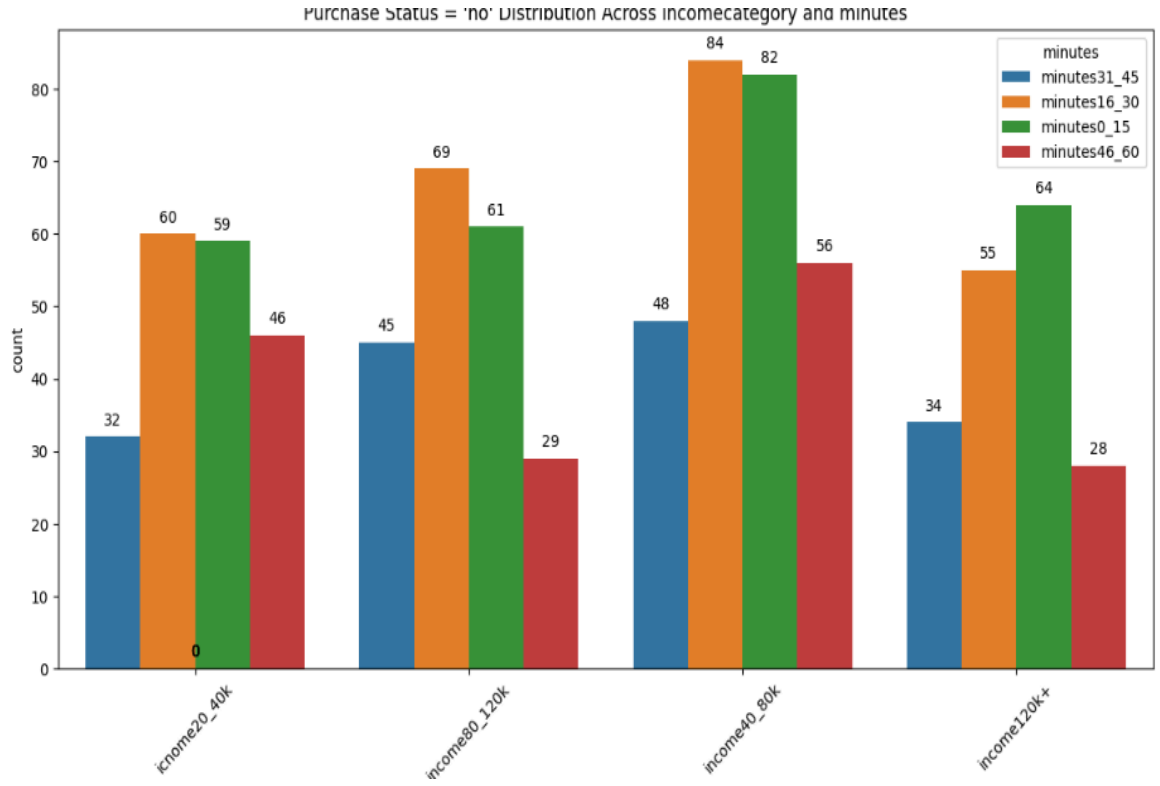


PurchaseStatus		no	yes
incomecategory	ProductCategory		
icnome20_40k	Beauty	74.00	26.00
	Clothing	75.36	24.64
	Electronics	82.93	17.07
	Home Goods	80.00	20.00
	Sports	84.44	15.56
income120k+	Beauty	46.58	53.42
	Clothing	50.00	50.00
	Electronics	52.63	47.37
	Home Goods	57.81	42.19
	Sports	59.21	40.79
income40_80k	Beauty	57.95	42.05
	Clothing	54.29	45.71
	Electronics	61.54	38.46
	Home Goods	53.33	46.67
	Sports	64.08	35.92
income80_120k	Beauty	41.33	58.67
	Clothing	43.68	56.32
	Electronics	52.00	48.00
	Home Goods	49.44	50.56
	Sports	40.21	59.79

Εικονα 92: count plot purchase status ,product category and income

Στο επόμενο διάγραμμα αναπαριστάτε η σχέση εισοδήματος και χρόνου σε λεπτά που αφιέρωσε ο πελάτης στο ηλεκτρονικό κατάστημα , όπως όμως έχουμε είδη δει στη γραμμική παλινδρομική δεν υπάρχει κάποια σχέση μεταξύ των δυο αυτών μεταβλητών. Παρόλα αυτά τα διαγράμματα και τα ποσοστά παρατίθενται παρακάτω.

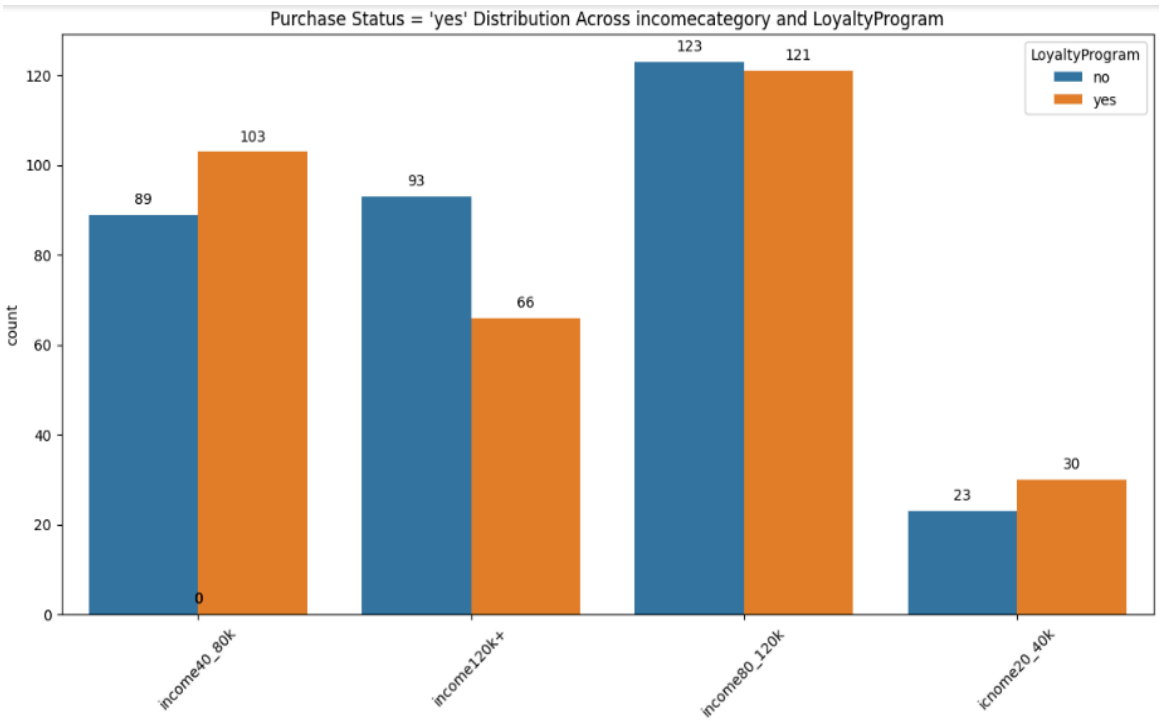

Purchase Status = 'yes' Distribution Across incomecategory and minutes
minutes
minutes31_45
minutes46_60
minutes16_30
minutes0_15
count
72
64
26
30
69
42
32
16
94
76
41
33
23
19
8
3
0
income40_80k
income120k+
income80_120k
income20_40k

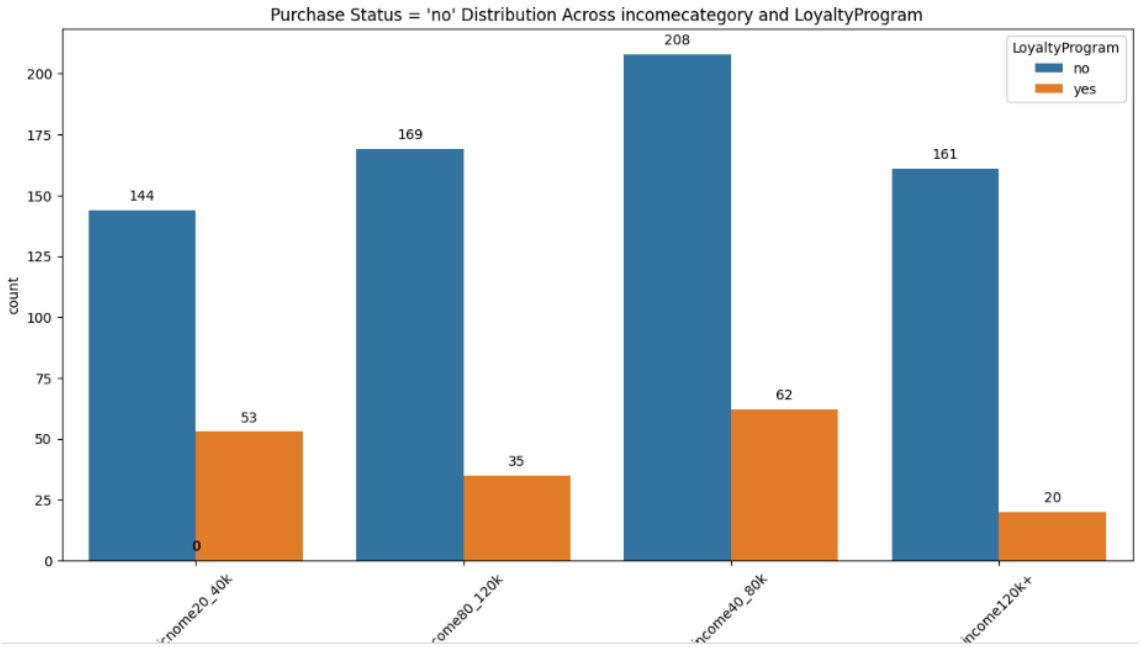

Purchase Status = 'no' Distribution Across incomecategory and minutes
minutes
minutes31_45
minutes16_30
minutes0_15
minutes46_60
count
32
60
59
46
45
69
61
29
48
84
82
56
34
55
64
28
0
income20_40k
income80_120k
income40_80k
income120k+

PurchaseStatus		no	yes
incomecategory	minutes		
icnome20_40k	minutes0_15	95.16	4.84
	minutes16_30	88.24	11.76
	minutes31_45	58.18	41.82
	minutes46_60	70.77	29.23
income120k+	minutes0_15	80.00	20.00
	minutes16_30	63.22	36.78
	minutes31_45	33.01	66.99
	minutes46_60	40.00	60.00
income40_80k	minutes0_15	73.21	26.79
	minutes16_30	76.36	23.64
	minutes31_45	40.00	60.00
	minutes46_60	46.67	53.33
income80_120k	minutes0_15	64.89	35.11
	minutes16_30	62.73	37.27
	minutes31_45	32.37	67.63
	minutes46_60	27.62	72.38

Εικονα 93: count plot purchase status ,income category and time spent

Στην συνέχεια έχουμε τα διαγράμματα μεταξύ ομάδας εισοδήματος και κάρτας μέλους. Αυτό που παρατηρούμε είναι ότι ανεξάρτητος κάρτας μέλους οι πελάτες με εισόδημα από 20 έως 40 χιλιάδες δολάρια δεν ολοκληρώνουν σε μεγάλο πόστο τις αγορές τους, ενώ σε όλες τις υπόλοιπές εισοδηματικές κατηγορίες η κατοχή κάρτας μέλους αυξάνει σε πολύ μεγάλο βαθμό την ολοκλήρωση της αγοράς.

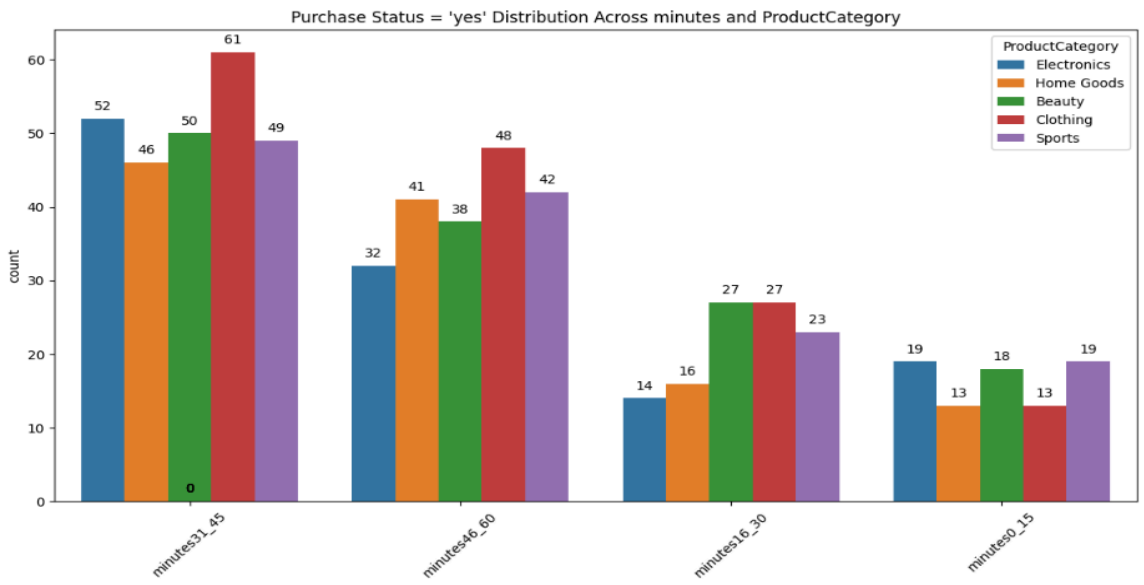




PurchaseStatus		no	yes
incomecategory	LoyaltyProgram		
icnome20_40k	no	86.23	13.77
	yes	63.86	36.14
income120k+	no	63.39	36.61
	yes	23.26	76.74
income40_80k	no	70.03	29.97
	yes	37.58	62.42
income80_120k	no	57.88	42.12
	yes	22.44	77.56

Εικονα 94: count plot purchase status ,income category and loyalty program

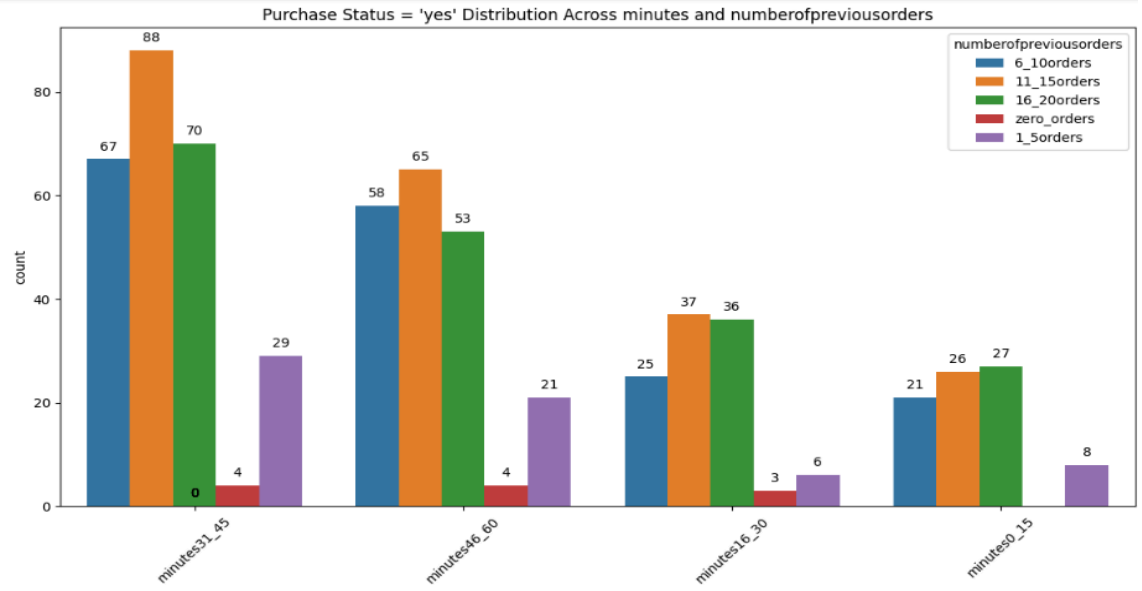
Μια επίσης ενδιαφέρουσα ανάλυση είναι μεταξύ του χρόνου που διέθεσε ο πελάτης στην ιστοσελίδα σε σχέση με τις κατηγορίες προϊόντων που αγοραστήκαν. Όπως παρατηρούμε τα άτομα που διέθεσαν 1 έως 15 λεπτά απέρριψαν την αγορά , η μονή κατηγορία προϊόντων με υψηλό ποσοστό σε αυτόν τον χρόνου είναι τα προϊόντα ομορφιάς το ίδιο ισχύει και για την κατηγορία των 16 έως 30 λεπτών όπου και εκεί την πρώτη θέση έχουν τα προϊόντα ομορφιάς. Από το διάγραμμα φαίνεται επίσης ότι οι πελάτες που ολοκληρώνουν την αγορά τους σε προϊόντα ρουχισμού αφιερώνουν μεταξύ 31 και 45 λεπτών και το ποσοστό ολοκλήρωσης είναι ίσο με 67%. Και τέλος τα άτομα που αφιερώνουν 45 έως 60 λεπτά είναι λίγο πάνω από το 50% η πιθανότητα να ολοκληρώσουν την αγορά ανεξάρτητος κατηγορίας προϊόντος.

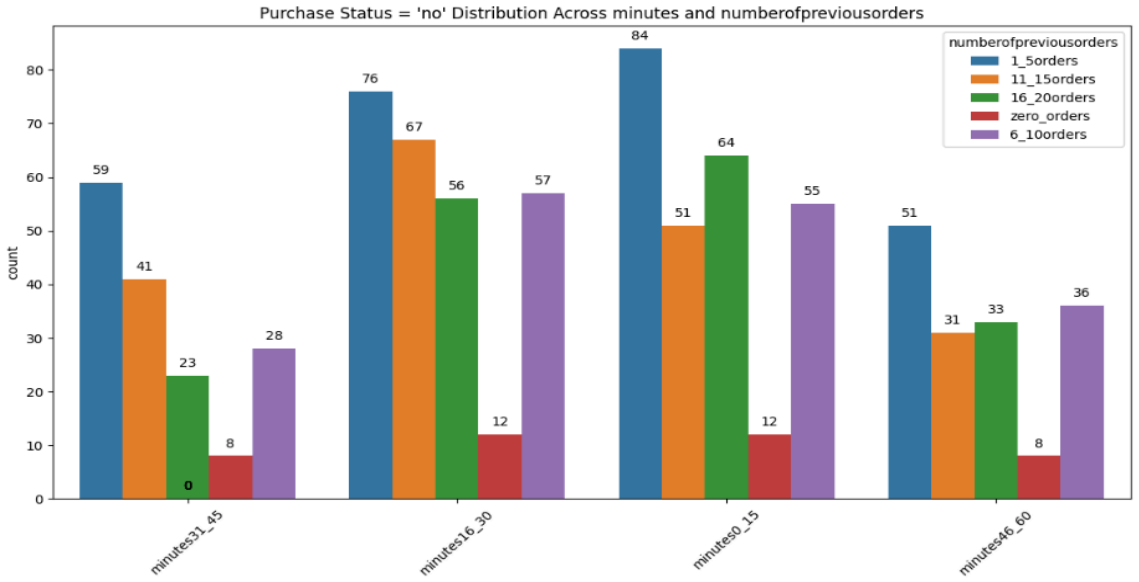


PurchaseStatus		no	yes
minutes	ProductCategory		
minutes0_15	Beauty	71.43	28.57
	Clothing	80.30	19.70
	Electronics	73.61	26.39
	Home Goods	78.69	21.31
	Sports	77.91	22.09
minutes16_30	Beauty	61.97	38.03
	Clothing	67.07	32.93
	Electronics	81.33	18.67
	Home Goods	76.12	23.88
	Sports	71.25	28.75
minutes31_45	Beauty	41.18	58.82
	Clothing	32.97	67.03
	Electronics	36.59	63.41
	Home Goods	38.67	61.33
	Sports	41.67	58.33
minutes46_60	Beauty	43.28	56.72
	Clothing	47.83	52.17
	Electronics	46.67	53.33
	Home Goods	41.43	58.57
	Sports	40.85	59.15

Εικονα 95: count plot purchase status ,product category and time spent

Στα παρακάτω διαγράμματα απεικονίζεται η σχέση μεταξύ του αριθμού των προηγουμένων αγορών σε σύγκριση με τον χρόνου που διαθέτουν οι πελάτες στο ηλεκτρονικό κατάστημα. Αυτό που παρατηρείται είναι ότι σε κάθε κατηγορία χρόνου στο κατάστημα όσο μεγαλύτερος είναι ο αριθμός των προηγουμένων αγορών τόσο μεγαλύτερη είναι και η πιθανότητα ολοκλήρωσης της αγοράς.





PurchaseStatus		no	yes
minutes	numberofpreviousorders		
minutes0_15	11_15orders	66.23	33.77
	16_20orders	70.33	29.67
	1_5orders	91.30	8.70
	6_10orders	72.37	27.63
	zero_orders	100.00	0.00
minutes16_30	11_15orders	64.42	35.58
	16_20orders	60.87	39.13
	1_5orders	92.68	7.32
	6_10orders	69.51	30.49
	zero_orders	80.00	20.00
minutes31_45	11_15orders	31.78	68.22
	16_20orders	24.73	75.27
	1_5orders	67.05	32.95
	6_10orders	29.47	70.53
	zero_orders	66.67	33.33
minutes46_60	11_15orders	32.29	67.71
	16_20orders	38.37	61.63
	1_5orders	70.83	29.17
	6_10orders	38.30	61.70
	zero_orders	66.67	33.33

Εικονα 96: count plot purchase status ,number of previous orders and time spent

Ολοκληρώνοντας τα παραπάνω προετοιμάζουμε τα δεδομένα για να εκτελέσουμε τους αλγορίθμους μηχανικής μάθησης για ταξινόμηση . Για να πραγματοποιηθεί το παραπάνω θα πρέπει τα δεδομένα να είναι σε αριθμητική μορφή και όχι κατηγορηματική (δηλαδή οι αλγόριθμοι δέχονται μόνο αριθμούς και όχι χαρακτήρες). Η πρώτη αλλαγή που καούμε είναι να επαναφέρουμε την στήλη με τα φυλά των πελατών από female και male σε 1 και 0 αντίστοιχα, το ίδιο καούμε και για την εξαρτημένη μεταβλητή δηλαδή αυτή όπου θέλουμε να ταξινομήσουμε το purchasestatus δηλαδή την αγορά η όχι του προϊόντος. Στην συνέχεια δημιουργούμε ψευδομεταβλητες για την κατηγορία προϊόντων και για το ποσοστό προσφοράς. Όπως αναφέραμε και πιο πάνω θα δοκιμάσουμε τους αλγορίθμους σε δυο μορφές των ιδίων δεδομένων το ένα dataset ονομάζεται behnobins και έχει μέσα τις μεταβλητές στην κανονική τους μορφή δηλαδή τις ηλικίες από το 18 έως το 70 , τις κανονικές τιμές του ετήσιου εισοδήματος από 20000 έως 150000 κλπ. ενώ στο dataset behbins έχουμε όλες τις αριθμητικές μεταβλητές σε μορφή ψευδομεταβλητων με βάση τα ιστογράμματα που είδαμε στην αρχή του κεφαλαίου. Οι μεταβλητές που επιλέχτηκαν για το κάθε dataset φαίνονται στην παρακατω εικόνα.

```
beh['Gender'] = beh['Gender'].replace({'Female': 1, 'Male': 0})
beh['PurchaseStatus']=beh['PurchaseStatus'].replace({'yes':1,'no':0})
```

```
<ipython-input-113-ba27bf98e4a5>:4: FutureWarning: Downcasting behavior in `replace` is deprecated and will
  beh['Gender'] = beh['Gender'].replace({'Female': 1, 'Male': 0})
<ipython-input-113-ba27bf98e4a5>:5: FutureWarning: Downcasting behavior in `replace` is deprecated and will
  beh['PurchaseStatus']=beh['PurchaseStatus'].replace({'yes':1,'no':0})
```

```
product_category_dummies = pd.get_dummies(beh['ProductCategory'], prefix='ProductCategory').astype(int)
beh = pd.concat([beh, product_category_dummies], axis=1)
discounts_availed_dummies = pd.get_dummies(beh['DiscountsAvailed'], prefix='DiscountsAvailed').astype(int)
beh = pd.concat([beh, discounts_availed_dummies], axis=1)
```

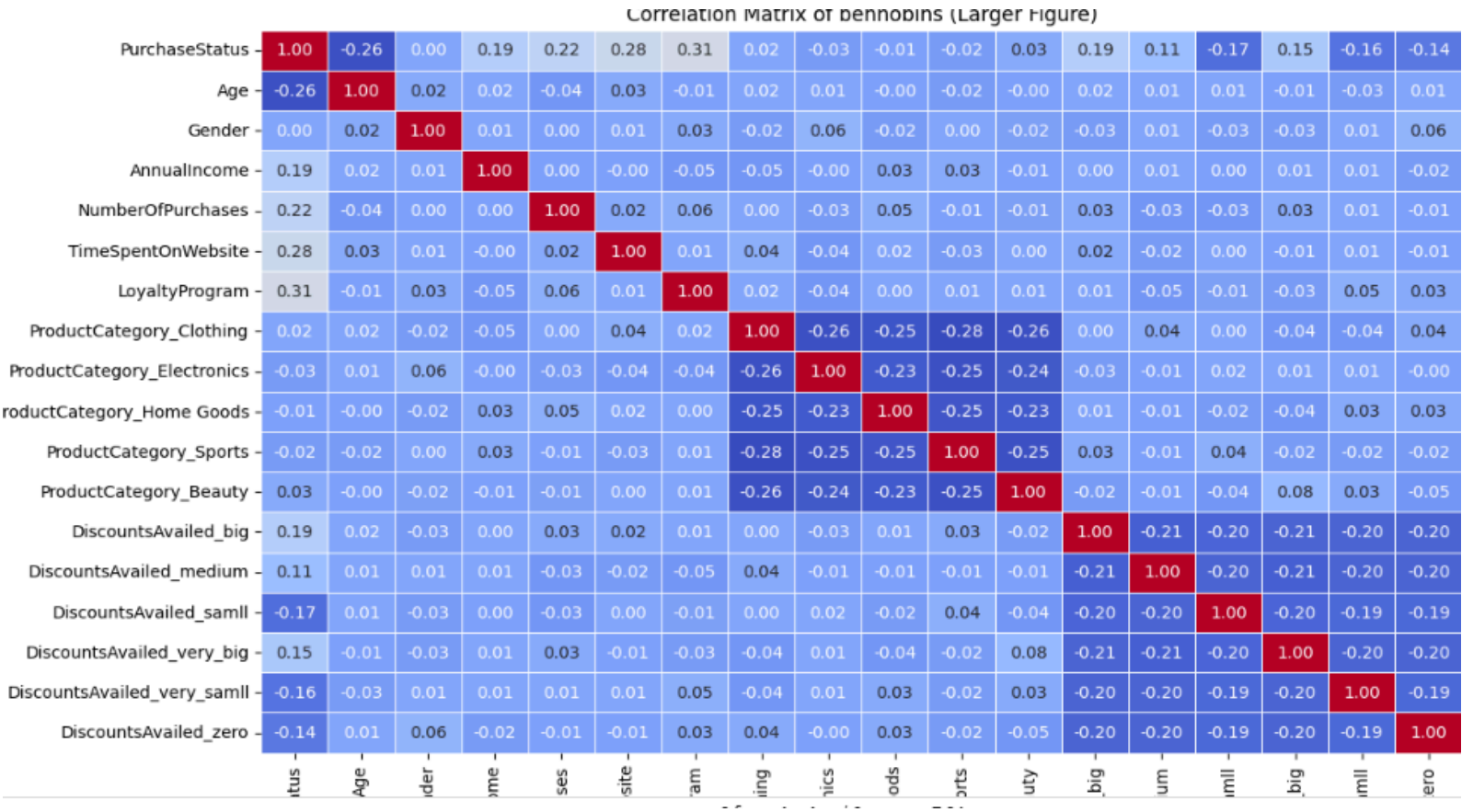
```
behnobins=beh[['PurchaseStatus','Age','Gender','AnnualIncome','NumberOfPurchases','TimeSpentOnWebsite','LoyaltyProgram',
               'ProductCategory_Clothing','ProductCategory_Electronics','ProductCategory_Home Goods','ProductCategory_Sports',
               'ProductCategory_Beauty',
               'DiscountsAvailed_big','DiscountsAvailed_medium','DiscountsAvailed_samll','DiscountsAvailed_very_big'
               ,'DiscountsAvailed_very_samll','DiscountsAvailed_zero']]
```

```
behbins=beh[['PurchaseStatus','Gender','age1830', 'age3140', 'age4150', 'age5160','age61above', 'income2040k', 'income4080k', 'income80120k',
             'income120kabove', 'np0', 'np1_5', 'np6_10', 'np11_15', 'np16_20','min015', 'min1630', 'min3145', 'min4660', 'ProductCategory_Electronics',
              'ProductCategory_Home Goods','ProductCategory_Clothing', 'ProductCategory_Beauty',
              'ProductCategory_Sports', 'DiscountsAvailed_big','DiscountsAvailed_medium', 'DiscountsAvailed_samll','DiscountsAvailed_very_big',
              'DiscountsAvailed_very_samll','DiscountsAvailed_zero']]
```

Εικονα 96: δημιουργία dataset για εισαγωγή του στα μοντέλα μηχανικής μάθησης

Έχοντας δημιουργήσει τα dataset και όλες οι μεταβλητές είναι σε αριθμητική μορφή μπορούμε να δούμε την συσχέτιση μεταξύ της μεταβλητής αγοράς ή όχι (purchase status) με τις υπόλοιπες μεταβλητές του behnobins καθώς το dataset με τις ψευδομεταβλητες έχει τόσες πολλές στήλες που είναι πολύ δύσκολη η απεικόνιση του. Όπως βλέπουμε η μεταβλητή έχει αρνητική συσχέτιση σε σχέση με την ηλικία δηλαδή τα άτομα μεγαλύτερης ηλικίας τείνουν να μην ολοκληρώνουν την αγορά τους, το φύλο του πελάτη δεν παίζει ρολό στην απόφαση αγοράς. Το ετήσιο εισόδημα ο αριθμός των προηγουμένων αγορών το εάν ο πελάτης έχει κάρτα μέλους η μέτρια η μεγάλη και η πολύ μεγάλη έκπτωση έχουν θετική συσχέτιση , ενώ η κατηγορία προϊόντος όπως φαίνεται δεν έχει κάποια συσχέτιση με την απόφαση αγοράς (ο συντελεστής συσχέτισης είναι σχεδόν σε όλες ίσος με το μηδέν βέβαια μεταξύ των πέντε κατηγοριών υπάρχει μικρή θετική συσχέτιση με τα προϊόντα ομορφιάς). ενώ αρνητική συσχέτιση δηλαδή οι μεταβλητές όπου μειώνουν την πιθανότητα αγοράς είναι η μηδενική ,η πολύ μικρή και η μικρή έκπτωση , όπου ίσως το κατάστημα θα μπορούσε να σταματήσει να παρέχει.

```
correlation_matrix = behbins.corr()
plt.figure(figsize=(20, 16))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".2f", linewidths=.5
plt.title('Correlation Matrix of behbins (Larger Figure)')
plt.show()
```



Εικονα 97: heatmap

Συνεχίζουμε την μοντελοποιηση και θα εφαρμόσουμε τα μοντέλα μηχανικής μάθησης αρχικά στα δεδομένα που είναι σε μορφή ψευδομεταβλητων , πριν από αυτό για να αποφευχθεί το dummy variable trap δηλαδή το πρόβλημα αναφορικά με την πολυσυγραμμικοτητα (multicollinearity) διαγράφουμε μια στήλη ψευδομεταβλητων ανά κατηγορία καθώς η πληροφορία αυτής εμπεριέχεται έμμεσος στις υπόλοιπές στήλες ψευδομεταυλητων. Για παράδειγμα διαγράφουμε την ψευδομεταβλητη με την ηλικιακή ομάδα 18 έως 30 την εισοδηματική κατηγορία 20 έως 40 χιλιάδων κλπ.

```
ehbins.drop(columns=['age1830','income2040k','np0','min015','ProductCategory_Electronics','DiscountsAvailed_big','ProductCategory_Sports','DiscountsAvailed_zero'],inplace=Tru
```

Έχοντας ολοκληρώσει τα παραπάνω μπορούμε να εφαρμόσουμε τους αλγορίθμους μηχανικής μάθησης αρχικά θέτουμε ως y την εξαρτημένη μεταβλητή purchasestatus

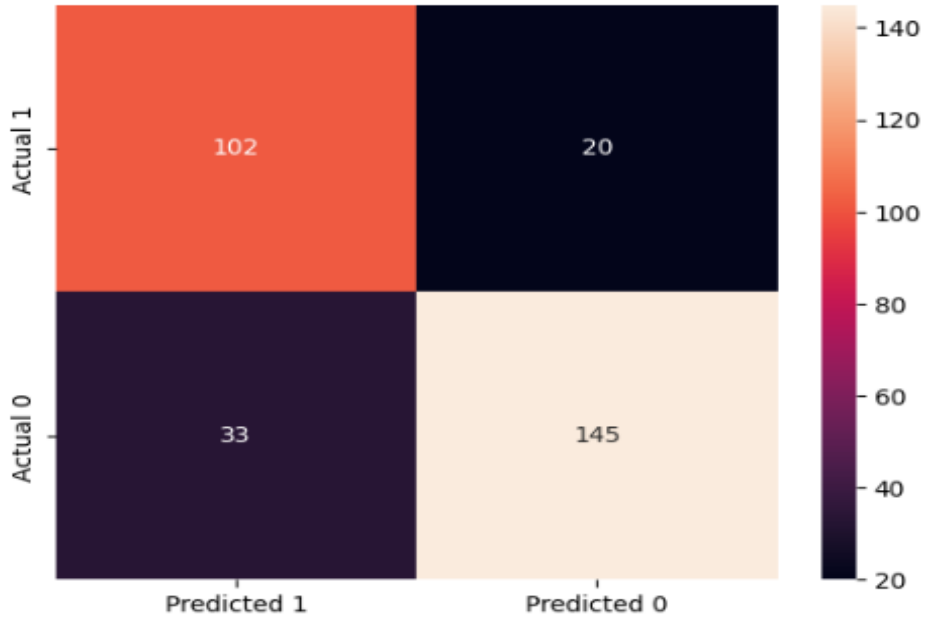
δηλαδή αυτήν όπου θέλουμε να μοντελοποιησουμε και ως χ όλες τις υπόλοιπές ερμηνευτικές μεταβλητές. Επίσης χωρίζουμε τα δεδομένα σε αυτά όπου θα χρησιμοποιήσει ο αλγόριθμος για να εκπαιδευτεί και να φτιάξει το μοντέλο και σε αυτά όπου θα δοκιμάσουμε την απόδοση του μοντέλου (δηλαδή τα δεδομένα όπου ο αναλυτής γνωρίζει την πραγματική τιμή δηλαδή το αν ο πελάτης αγόρασε η όχι τελικά αλλά ο αλγόριθμος δεν τα έχει ξανά δει). Συγκεκριμένα ο διαχωρισμός είναι 80% του συνόλου των δεδομένων δίνεται στον αλγόριθμο για να εκπαιδευτεί και το υπόλοιπο 20% δια κρατείται για να δοκιμαστεί η απόδοση του μοντέλου.

Αρχικά εφαρμόζουμε το μοντέλο λογιστικής παλινδρόμησης στα δεδομένα και δημιουργούμε τον confusion matrix, στον οποίο κανούμε κάποιες μετατροπές ώστε να εμφανίζονται οι τιμές όπως είδαμε στο κεφάλαιο 4 (συγκεκριμένα χωρίς την μετατροπή η python εμφανίζει τις τιμές true negative στην θέση των true positive και τις τιμές του false negative στην περιοχή των false positives προφανώς θα μπορούσε ο εκάστοτε αναλυτής να τις αφήσει όπως τις εξάγει η python).

Όπως παρατηρούμε η λογιστική παλινδρόμηση μοντελοπιει σε πολύ καλό βαθμό τα δεδομένα με αποτέλεσμα το accuracy να είναι ίσο με 82%. Πιο συγκείμενα η απόδοση του μοντέλου ελέγχθηκε με βάση 300 παρατηρήσεις (1500 (σύνολο παρατηρήσεων) *0,2 (test dataset size)=300) ,οι πελάτες που αγοράσαν όντως το προϊόν που έβαλαν στο καλάθι τους και το μοντέλο τους ταξινόμησε ότι όντως αγορά είναι ίσοι με 102(true positive), οι πελάτες οι οποίοι δεν αγοράσαν στην πραγματικότητα το προϊόν και ο αλγόριθμος τους ταξινόμησε σωστά ότι δεν αγοράσαν είναι 145(false positive) αρά και το accuracy είναι ίσο με 82% καθώς (102+145)/300=0,82. Όσον αφορά τις λανθασμένες ταξινομήσεις ο αλγόριθμος θεώρησε ότι 20 άτομα αγοράσανε το προϊόν που είχαν βάλει στο καλάθι τους αλλά στην πραγματικότητα δεν το αγοράσανε(false positive), ενώ 33 άτομα ο αλγόριθμος εκτίμησε ότι δεν αγοράσαν τα προϊόντα στο καλάθι τους αλλά στην πραγματικότητα πραγματοποίησαν την αγορά.

```
x=behbins.drop('PurchaseStatus',axis=1)
y=behbins['PurchaseStatus']
from sklearn.model_selection import train_test_split
xtrain,xtest,ytrain,ytest=train_test_split(x,y,test_size=0.2)
```

```
from sklearn.linear_model import LogisticRegression
logr=LogisticRegression()
logr.fit(xtrain,ytrain)
ypredlogr=logr.predict(xtest)
from sklearn.metrics import confusion_matrix
cm=confusion_matrix(ytest,ypredlogr,labels=[1,0])
cm[0, 1], cm[1, 0] = cm[1, 0], cm[0, 1]
sns.heatmap(cm, annot=True, fmt='d', xticklabels=['Predicted 1', 'Predicted 0'], yticklabels=['Actual 1', 'Actual 0'])
plt.show()
```




```
from sklearn.metrics import classification_report
print(classification_report(ytest,ypredlogr))
```

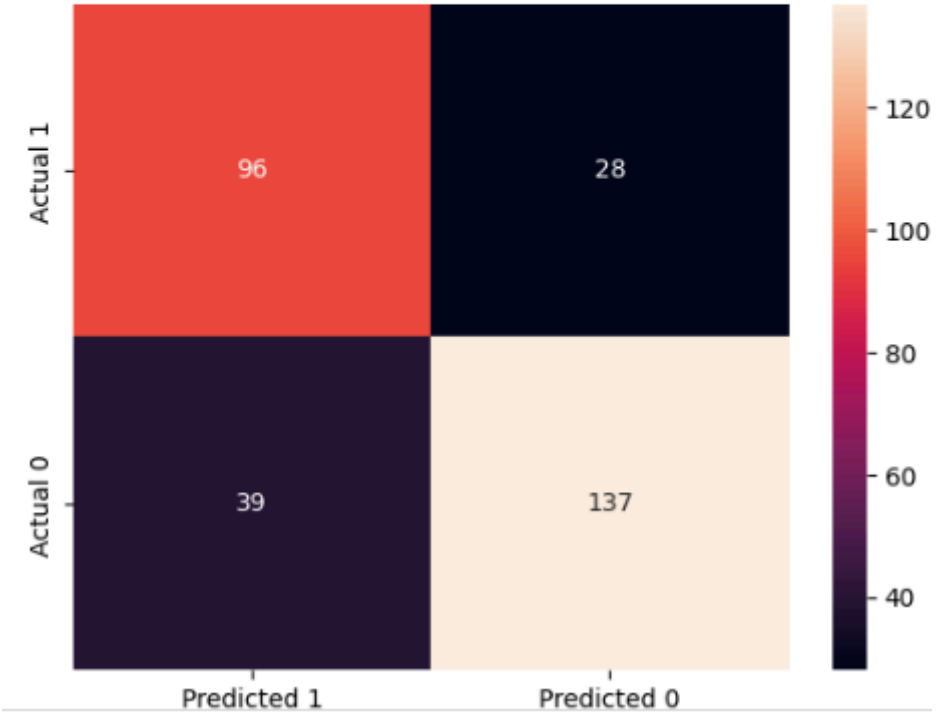
	precision	recall	f1-score	support
0	0.81	0.88	0.85	165
1	0.84	0.76	0.79	135
accuracy			0.82	300
macro avg	0.83	0.82	0.82	300
weighted avg	0.82	0.82	0.82	300

Εικονα 98: confusion matrix logistic regression

Ο επόμενος αλγόριθμος που χρησιμοποιήθηκε είναι το support vector machine με accuracy ίσο με 78% εδώ ο αλγόριθμος στα ιδία δεδομένα ενέχου(test set) κάνει περισσότερα λάθη με αποτέλεσμα και οι false negative και οι false positive εκτιμήσεις να αυξηθούν.

```
from sklearn.svm import SVC
svc=SVC()
svc.fit(xtrain,ytrain)
ypredsvc=svc.predict(xtest)
cm=confusion_matrix(ytest,ypredsvc,labels=[1,0])
cm[0, 1], cm[1, 0] = cm[1, 0], cm[0, 1]
sns.heatmap(cm, annot=True, fmt='d', xticklabels=['Predicted 1', '
```

```
<Axes: >
```



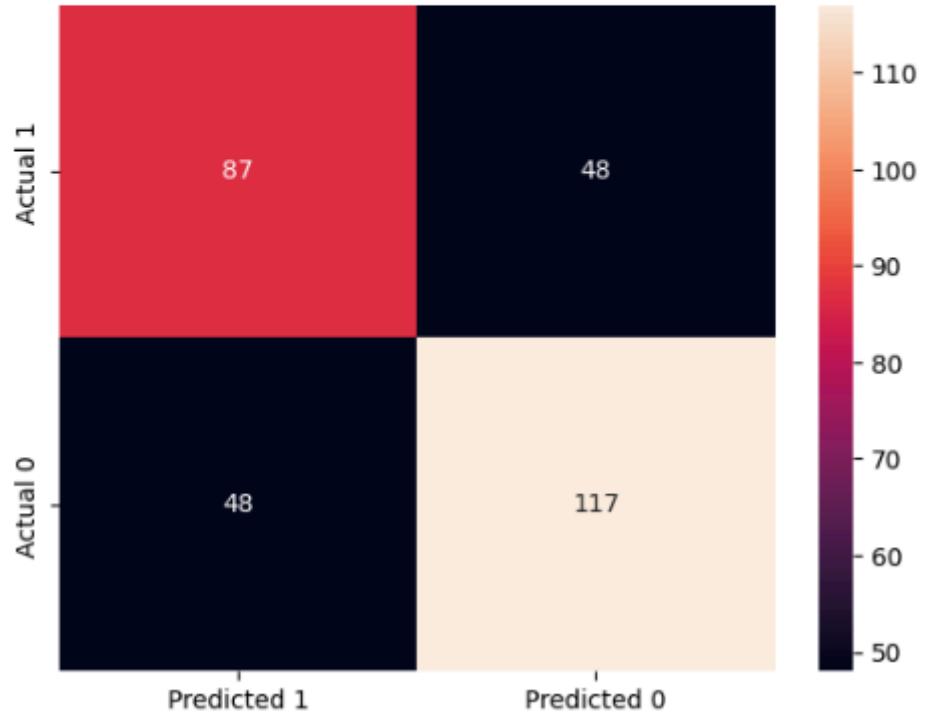

```
print(classification_report(ytest,ypredsvc))
```

```
              precision    recall  f1-score   support

           0       0.78      0.83      0.80       165
           1       0.77      0.71      0.74       135

    accuracy                           0.78       300
   macro avg       0.78      0.77      0.77       300
weighted avg       0.78      0.78      0.78       300
```

Εικονα 99: confusion matrix svc

Στην συνέχεια χρησιμοποιήθηκε ο αλγόριθμος κ κοντινότεροι γείτονες και ο τρόπος υπολογισμού της απόστασης μεταξύ των νέων τιμών υπολογίζεται με βάση την ευκλείδεια απόσταση και η απόφαση σχετικά με την ταξινόμηση της νέας παρατήρησης βγαίνει με βάση τους πέντε κοντινότερους γείτονες. Το μοντέλο αποδίδει χειροτέρα σε σχέση με τα δυο προηγούμενα καθώς το accuracy είναι ίσο με 68%.

```
from sklearn.neighbors import KNeighborsClassifier
knn=KNeighborsClassifier(n_neighbors=5,metric='minkowski',p=2)
knn.fit(xtrain,ytrain)
ypredknn=knn.predict(xtest)
cm=confusion_matrix(ytest,ypredknn,labels=[1,0])
cm[0, 1], cm[1, 0] = cm[1, 0], cm[0, 1]
sns.heatmap(cm, annot=True, fmt='d', xticklabels=['Predicted 1', 'Pr
plt.show()
```




```
print(classification_report(ytest,ypredknn))
```

	precision	recall	f1-score	support
0	0.71	0.71	0.71	165
1	0.64	0.64	0.64	135
accuracy			0.68	300
macro avg	0.68	0.68	0.68	300
weighted avg	0.68	0.68	0.68	300

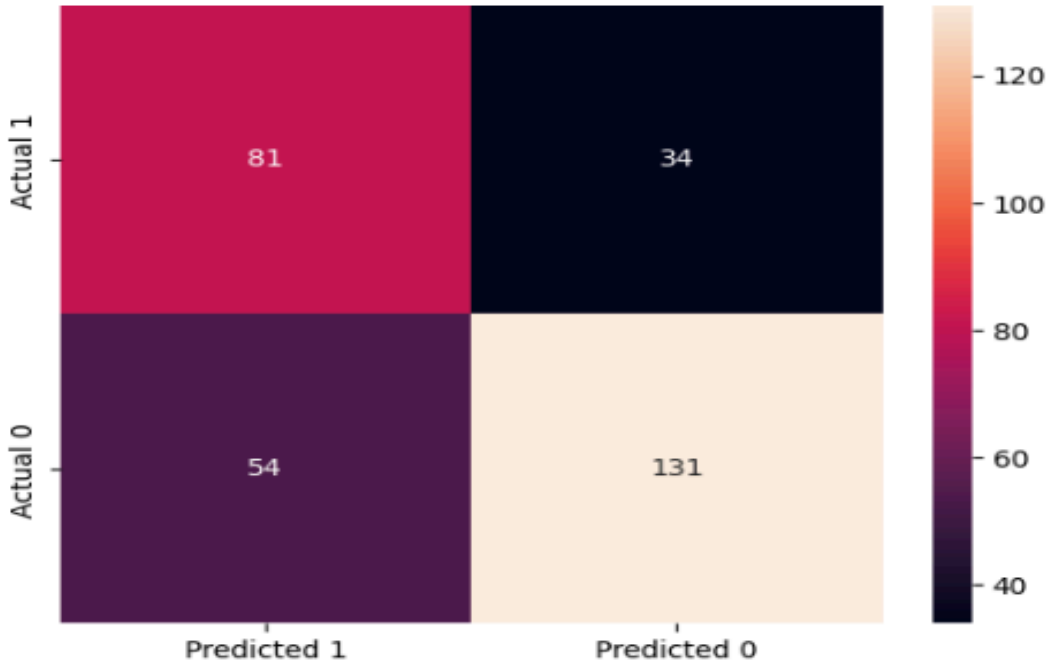
Εικονα 100: confusion matrix knn

Συνεχίζουμε με τα αποτελέσματα του δέντρου απόφασης. Ο συγκεκριμένος αλγόριθμος έχει accuracy 71% και αν τον χρησιμοποιήσει το ηλεκτρονικό κατάστημα λογικά θα καταλήξει με ελλείψεις αποθεμάτων καθώς το μοντέλο έχει κάνει τα περισσότερα λάθη false negative δηλαδή ταξινομεί πολλούς πελάτες ότι δεν θα αγοράσουν τα προϊόντα στο καλάθι τους ενώ τελικά εκείνοι στην πραγματικότητα τα αγοράσαν. Στην συνέχεια

ακολουθούν οι συντελεστές τις κάθε ερμηνευτικής μεταβλητής όπου με βάση αυτούς οδηγήθηκε το μοντέλο στις συγκεκριμένες ταξινομήσεις ,και όπως βλέπουμε οι συντελεστές σημαντικότητας είναι ισομιρασμενοι.

```
from sklearn.tree import DecisionTreeClassifier
tree=DecisionTreeClassifier()
tree.fit(xtrain,ytrain)
ypredt=tree.predict(xtest)
cm=confusion_matrix(ytest,ypredt,labels=[1,0])
cm[0, 1], cm[1, 0] = cm[1, 0], cm[0, 1]
sns.heatmap(cm, annot=True, fmt='d', xticklabels=['Predicted 1', 'Pr
```

```
<Axes: >
```




```
print(classification_report(ytest,ypredt))
```

	precision	recall	f1-score	support
0	0.71	0.79	0.75	165
1	0.70	0.60	0.65	135
accuracy			0.71	300
macro avg	0.71	0.70	0.70	300
weighted avg	0.71	0.71	0.70	300

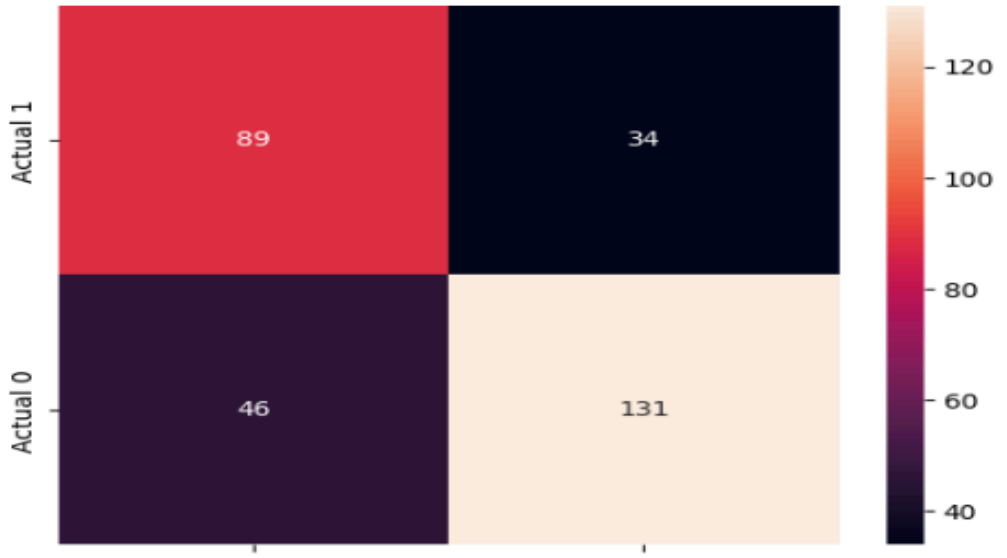
Εικονα 101: confusion matrix decision tree

```
feature_importances = tree.feature_importances_
for i, feature_name in enumerate(x.columns):
  print(f"{feature_name}: {feature_importances[i]}")
```

```
Gender: 0.06224740469549244
age3140: 0.02758132694664897
age4150: 0.06521617034731934
age5160: 0.04117443251360994
age61above: 0.04477257199160685
income4080k: 0.04015292304512032
income80120k: 0.063342342386387 74
income120kabove: 0.03458168988705881
np1_5: 0.06261190311095383
np6_10: 0.027856036828615933
np11_15: 0.04299714814843727
np16_20: 0.029342111901460695
min1630: 0.03773481266070405
min3145: 0.050947131151716646
min4660: 0.04841805485675898
ProductCategory_Home Goods: 0.041554248011800 27
ProductCategory_Clothing: 0.05680090433667645
ProductCategory_Beauty: 0.05073148190237921
DiscountsAvailed_medium: 0.04593990314584917
DiscountsAvailed_small: 0.04621601830211361
DiscountsAvailed_very_big: 0.028878410053741473
DiscountsAvailed_very_small: 0.050902973775548185
```

Ο τελευταίος αλγόριθμος που χρησιμοποιήθηκε για να ταξινομήσει τα δεδομένα είναι τα τυχαία δάση, συγκείμενα για τον υπολογισμό της τελικής απόφασης ταξινόμησης χρησιμοποιήθηκαν 200 δέντρα αποφάσεων. Ο αλγόριθμος έχει accuracy 73% που τον κάνει να είναι πιο αποδοτικό μοντέλο σε σχέση με το δέντρο απόφασης που είδαμε παραπάνω , ο αλγόριθμος έχει ταξινομήσει 34 πελάτες ότι αγοράσανε το προϊόν που βάλανε στο καλάθι τους αλλά στην πραγματικότητα αυτό δεν έγινε όσοι ήταν και οι αντίστοιχοι του δέντρου απόφασης , όσο αναφορά τις false negative ταξινομήσεις αυτές είναι λιγότερες σε σχέση με τον αλγόριθμο του δέντρου. Το συμπέρασμα συνολικά όλων των παραπάνω αλγορίθμων είναι ότι ο πιο αποδοτικός ήταν η λογιστική παλινδρόμηση , αρά χρησιμοποιώντας την το ηλεκτρονικό κατάστημα θα μπορέσει σε μεγάλο βαθμό να κρατήσει το σωστό επίπεδο αποθεμάτων χωρίς να χάνει πελάτες σε περιπτώσεις υψηλής ζήτησης αλλά και να μην διακρατει μεγάλο όγκο αποθεμάτων όπου με την πάροδο του χρόνου χάνει την αξία του αυξάνει τα κόστη αποθήκευσης και μειώνει τα κεφάλαια τα οποία θα μπορούσαν να είχαν χρησιμοποιηθεί σε άλλες επενδύσεις, καθώς πλέον ο επιχείρηση γνωρίζει πριν καν αποφασίσει ο πελάτης αν θα αγοράσει το προϊόν που έχει στο καλάθι του , αν όντως θα το πράξει με βάση τα χαρακτηριστικά του (τις ερμηνευτικές μεταβλητές). Παρόλα αυτά θα δοκιμάσουμε τους ιδίους αλγορίθμους και στα δεδομένα που δεν έχουν διαχωριστεί οι αριθμητικές μεταβλητές σε ψευδομεταβλητες.

```
from sklearn.ensemble import RandomForestClassifier
rf=RandomForestClassifier(n_estimators=200)
rf.fit(xtrain,ytrain)
ypredrf=rf.predict(xtest)
cm=confusion_matrix(ytest,ypredrf,labels=[1,0])
cm[0, 1], cm[1, 0] = cm[1, 0], cm[0, 1]
sns.heatmap(cm, annot=True, fmt='d', xticklabels=['Predicted 1', 'Pr
plt.show()
```




```
print(classification_report(ytest,ypredrf))
```

	precision	recall	f1-score	support
0	0.74	0.79	0.77	165
1	0.72	0.66	0.69	135
accuracy			0.73	300
macro avg	0.73	0.73	0.73	300
weighted avg	0.73	0.73	0.73	300

Εικονα 102: confusion matrix random forest

Στην συνέχεια χρησιμοποιούμε το dataset behnobins όπου όπως έχουμε αναφέρει οι κατηγορηματικές μεταβλητές όπως η κατηγορία προϊόντων έχει μετατραπεί σε ψευδομεταβλητες ενώ οι αριθμητικές έχουν παραμείνει στην αρχική τους κατάσταση. Όπως και στο προηγούμενο dataset διαγράφουμε μια dummy variable στήλη από κάθε σύνολο ψευδομεταβλητων ώστε να αποφύγουμε το dummy variable trap. Στην συγκεκριμένη περίπτωση διαγράψαμε την ψευδομεταβλητη με την κατηγορία προϊόντων ηλεκτρονικών ειδών και από τις εκπτώσεις την έκπτωση big(4%). Στην συνέχεια χορδίσαμε το dataset σε train και test set με 80% και 20% του συνόλου των δεδομένων αντίστοιχα(τα δεδομένα (γραμμές) εκπαίδευσης και ελέγχου δεν είναι ιδία με τα

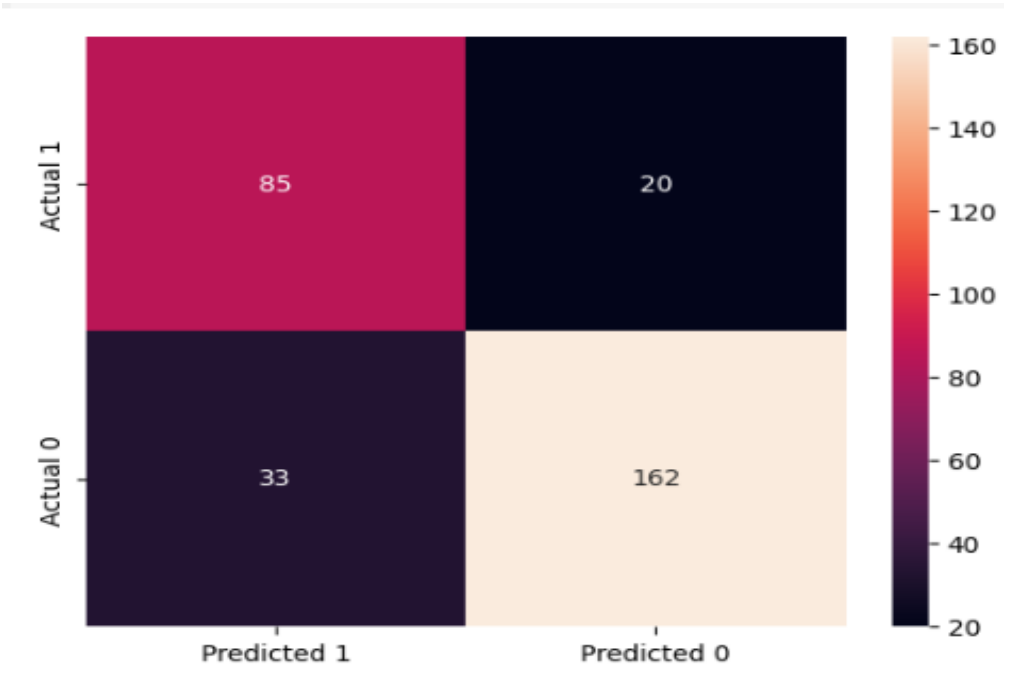
αντίστοιχα που χρησιμοποιήθηκαν στο dataset behbins). Ο πρώτος αλγόριθμος που θα εξετάσουμε είναι η λογιστική παλινδρόμηση η οποία έχει το ίδιο accuracy με το μοντέλο λογιστικής παλινδρόμησης με τις αριθμητικές μεταβλητές ως ψευδομεταβλητες.

```
behnobins.drop(columns=['ProductCategory_Electronics','DiscountsAvailed_big'],inplace=True)
```

Εμφάνιση μη ορατής εξόδου

```
x=behnobins.drop('PurchaseStatus',axis=1)
y=behnobins['PurchaseStatus']
from sklearn.model_selection import train_test_split
xtrain,xtest,ytrain,ytest=train_test_split(x,y,test_size=0.2)
```

```
from sklearn.linear_model import LogisticRegression
logr=LogisticRegression(max_iter=10000)
logr.fit(xtrain,ytrain)
ypredlogr=logr.predict(xtest)
from sklearn.metrics import confusion_matrix
cm=confusion_matrix(ytest,ypredlogr,labels=[1,0])
cm[0, 1], cm[1, 0] = cm[1, 0], cm[0, 1]
sns.heatmap(cm, annot=True, fmt='d', xticklabels=['Predicted 1', 'Predicted 0'], yticklabels=['Actual 1', 'Actual 0'])
plt.show()
```

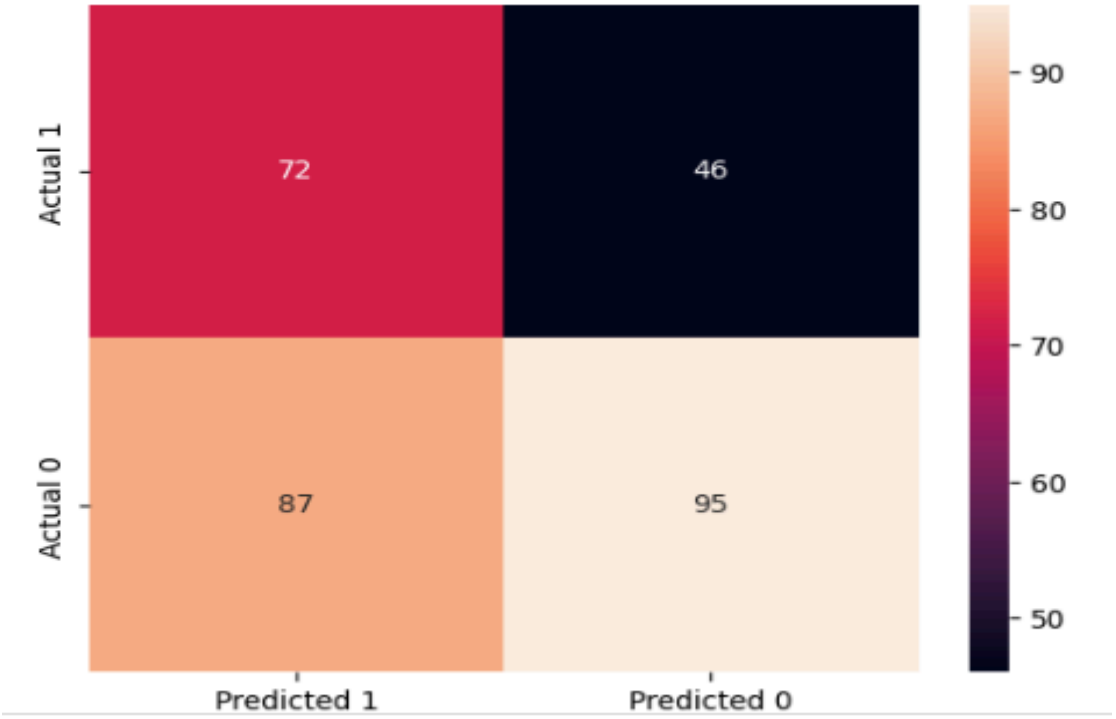


	precision	recall	f1-score	support
0	0.83	0.89	0.86	182
1	0.81	0.72	0.76	118
accuracy			0.82	300
macro avg	0.82	0.81	0.81	300
weighted avg	0.82	0.82	0.82	300

Εικονα 103: confusion matrix logistic regression

Το μοντέλο support vector machine ταξινομεί τα δεδομένα χειροτέρα σε σχέση με τον αντίστοιχο με τις αριθμητικές μεταβλητές ως ψευδομεταβλητες καθώς το accuracy του είναι 56% έναντι 78%.

```
from sklearn.svm import SVC
svc=SVC()
svc.fit(xtrain,ytrain)
ypredsvc=svc.predict(xtest)
cm=confusion_matrix(ypredsvc,ytest,labels=[1,0])
cm[0, 1], cm[1, 0] = cm[1, 0], cm[0, 1]
sns.heatmap(cm, annot=True, fmt='d', xticklabels=['Predicted 1', 'Pr
plt.show()
```




```
print(classification_report(ytest,ypredsvc))
```

	precision	recall	f1-score	support
0	0.67	0.52	0.59	182
1	0.45	0.61	0.52	118
accuracy			0.56	300
macro avg	0.56	0.57	0.55	300
weighted avg	0.59	0.56	0.56	300

Εικονα 104: confusion matrix svc

Τα αποτελέσματα ταξινόμησης συνεχίζουν να είναι χειροτέρα μεταξύ των αρχικών δεδομένων έναντι των δεδομένων με τις αριθμητικές μεταβλητές ως ψευδομεταβλητες καθώς και ο αλγόριθμος κ κοντινότερων γειτόνων έχει accuracy ίσο με 62% ενάτη 68% του πρώτού.

```
from sklearn.neighbors import KNeighborsClassifier
knn=KNeighborsClassifier(n_neighbors=5,metric='minkowski',p=2)
knn.fit(xtrain,ytrain)
ypredknn=knn.predict(xtest)
cm=confusion_matrix(ytest,ypredknn,labels=[1,0])
cm[0, 1], cm[1, 0] = cm[1, 0], cm[0, 1]
sns.heatmap(cm, annot=True, fmt='d', xticklabels=['Predicted 1', 'P
plt.show()
```

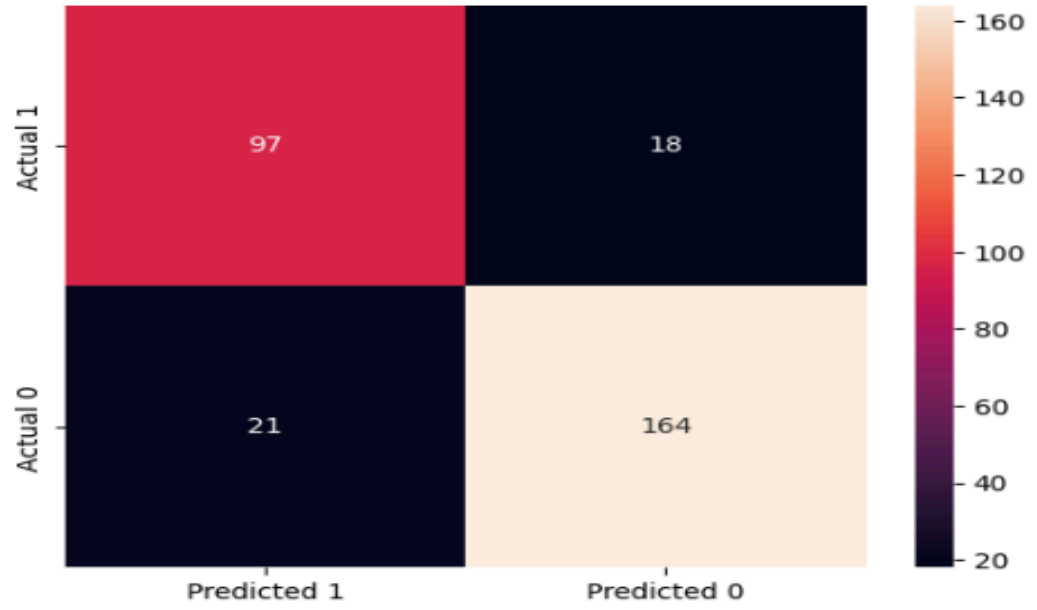

```
print(classification_report(ytest,ypredknn))
```

	precision	recall	f1-score	support
0	0.68	0.70	0.69	182
1	0.52	0.49	0.50	118
accuracy			0.62	300
macro avg	0.60	0.60	0.60	300
weighted avg	0.62	0.62	0.62	300

Εικονα 104: confusion matrix knn

Στην συνέχεια χρησιμοποιήθηκε ο αλγόριθμος του δέντρου απόφασης ο οποίος αποδίδει καλυτέρα σε σχέση με τον αντίστοιχο που χρησιμοποιήθηκε στο behbins και μάλιστα το accuracy του 87% είναι μεγαλύτερο ακόμα και από αυτό της λογιστικής παλινδρόμησης του dataset behbins 82%. Όσον αφορά τις μεταβλητές όπου παίζουν κύριο ρολό στην απόφαση ταξινόμησης του μοντέλου αυτές είναι κύριος οι μεταβλητές ηλικία ετήσιο εισόδημα ο αριθμός προηγουμένων αγορών ο χρόνος στην ιστοσελίδα και η κατοχή η όχι κάρτας μέλους , ενώ το επίπεδο έκπτωσης και η κατηγορία προϊόντος δεν συμβάλουν καθοριστικά στο αποτέλεσμα ταξινόμησης.

```
from sklearn.tree import DecisionTreeClassifier
tree=DecisionTreeClassifier()
tree.fit(xtrain,ytrain)
ypredt=tree.predict(xtest)
cm=confusion_matrix(ytest,ypredt,labels=[1,0])
cm[0, 1], cm[1, 0] = cm[1, 0], cm[0, 1]
sns.heatmap(cm, annot=True, fmt='d', xticklabels=['Predicted 1', 'P
plt.show()
```




```
print(classification_report(ytest,ypredt))
```

	precision	recall	f1-score	support
0	0.89	0.90	0.89	182
1	0.84	0.82	0.83	118
accuracy			0.87	300
macro avg	0.86	0.86	0.86	300
weighted avg	0.87	0.87	0.87	300

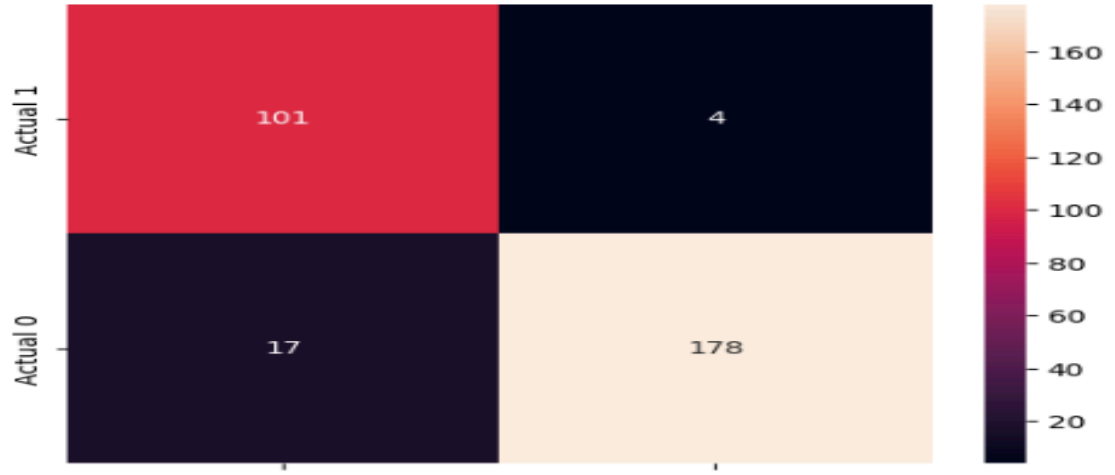
Εικονα 105: confusion matrix logistic decision tree

```
feature_importances = tree.feature_importances_
for i, feature_name in enumerate(x.columns):
  print(f"{feature_name}: {feature_importances[i]}")
```

```
Age: 0.1559750633279394
Gender: 0.006082793579273444
AnnualIncome: 0.16413282833481763
NumberOfPurchases: 0.14514667633265635
TimeSpentOnWebsite: 0.15878424553847156
LoyaltyProgram: 0.10168298183857008
ProductCategory_Clothing: 0.0022716605650990314
ProductCategory_Home Goods: 0.007378941846881062
ProductCategory_Sports: 0.0031978370826582 55
ProductCategory_Beauty: 0.0037253087661423 36
DiscountsAvailed_medium: 0.026743296903123 99
DiscountsAvailed_small: 0.058809283641016 14
DiscountsAvailed_very_big: 0.028350461072607358
DiscountsAvailed_very_small: 0.05542675282818053
DiscountsAvailed_zero: 0.0822918683425629
```

Και τέλος χρησιμοποιείται το μοντέλο του τυχαίου δάσους με 200 δέντρα απόφασης το οποίο αποδίδει καλυτέρα από όλα τα μοντέλα όπου έχουμε εξετάσει μέχρι τώρα συνολικά. Όπως μπορούμε να δούμε ο αλγόριθμος έκανε μόνο τέσσερεις λάθος ταξινομήσεις false positives δηλαδή ο πελάτης να μην αγόρασε στην πραγματικότητα ενώ ο αλγόριθμος τον ταξινόμησε ότι αγόρασε. Το accuracy του μοντέλου είναι ίσο με 93%.

```
from sklearn.ensemble import RandomForestClassifier
rf=RandomForestClassifier(n_estimators=200)
rf.fit(xtrain,ytrain)
ypredrf=rf.predict(xtest)
cm=confusion_matrix(ytest,ypredrf,labels=[1,0])
cm[0, 1], cm[1, 0] = cm[1, 0], cm[0, 1]
sns.heatmap(cm, annot=True, fmt='d', xticklabels=['Predicted 1', 'P
plt.show()
```




```
print(classification_report(ytest,ypredrf))
```

	precision	recall	f1-score	support
0	0.91	0.98	0.94	182
1	0.96	0.86	0.91	118
accuracy			0.93	300
macro avg	0.94	0.92	0.93	300
weighted avg	0.93	0.93	0.93	300

Εικονα 106: confusion matrix random forest

Το συμπέρασμα όλων των παραπάνω είναι ότι εάν πρόκειται τις αριθμητικές μεταβλητές να τις διαχωρίσει ο αναλυτής σε κατηγορίες με την χρήση ψευδομεταβλητων το καλύτερο μοντέλο για τα συγκείμενα δεδομένα είναι η λογιστική παλινδρόμηση , εάν πάλι θέλει να τα κρατήσει στην αρχική τους μορφή υπάρχουν μοντέλα όπως τα δέντρα απόφασης και τα τυχαία δάση που επίσης δίνουν πολύ αποτελεσματικές ταξινομήσεις.

5. ΜΕΔΟΔΟΙ ΑΝΑΛΥΣΗΣ ΧΡΟΝΟΛΟΓΙΚΩΝ ΣΕΙΡΩΝ

Σε αυτό το κεφάλαιο θα μελετήσουμε τις μεθόδους ανάλυσης των χρονολογικών σειρών. Αρχικά θα αναφέρουμε τον ορισμό των χρονολογικών σειρών ,στην συνέχεια θα παρουσιάσουμε τα βασικά χαρακτηριστικά τους και έχοντας ολοκληρώσει τα παραπάνω θα προσπαθήσουμε να προβλέψουμε την μελλοντική τιμή ενός κινητού τηλεφώνου καθώς όπως είδαμε και στο πρώτο κεφάλαιο ένας από τους βασικότερους προσδιοριστικούς παράγοντες της ζήτησης είναι η τιμή του προϊόντος, χρησιμοποιώντας από πολύ απλές μεθόδους ανάλυσης χρονοσειρων όπως ο απλός κινητός μέσος ορός μέχρι την χρήση νευρωνικων δικτυών όπως τα lstm (long/short term memory).

Αρχικά θα πρέπει να αναφέρουμε ότι η ανάλυση χρονολογικών σειρών ανήκει στην κατηγορία των στατιστικών και μαθηματικών μεθόδων για την δημιουργεί προβλέψεων για διάφορες μεταβλητές στο μέλλον όπως τις τιμές μέτοχων της πωλήσεις προϊόντων κλπ. και πιο συγκεκριμένα ανήκει στην κατηγορία των μη αιτιατών μεθόδων σε αντίθεση με πχ την γραμμική παλινδρόμηση οπού είναι ένα μοντέλο μηχανικής μάθησης οπού ανήκει στην κατηγορία των αιτιατών μοντέλων καθώς υπάρχει η εξαρτημένη μεταβλητή και οι ερμηνευτές μεταβλητές που μέσο αυτών προσπαθούμε να αναλύσουμε την μεταβλητότητα της εξαρτημένης για παράδειγμα για τον προσδιορισμό των μισθών των υπάλληλων μιας εταιρείας (εξαρτημένη μεταβλητή) μπορούν να χρησιμοποιηθούν ερμηνευτικές μεταβλητές όπως η προϋπηρεσία μετρημένη σε έτη η κατοχή ή όχι κάποιου πτυχίου κλπ. σε αντίθεση με τα μη αιτιατά μοντέλα οπού για τον προσδιορισμό της επομένης τιμής χρησιμοποιούνται οι προηγούμενες τιμές τις ιδίας της μεταβλητής , για παράδειγμα για τον προσδιορισμό της τιμής κλεισίματος της μετοχής της επομένης ημέρας χρησιμοποιούνται οι τιμές κλεισίματος των προηγουμένων ημέρων προφανώς και σε αυτά τα μοντέλα μπορούν να χρησιμοποιηθούν εξωγενείς μεταβλητές για να βοηθήσουν το μοντέλο να οδηγηθεί σε καλύτερες προβλέψεις αλλά σε αυτή την εργασία για να μείνει όσο τον δυνατόν πιο απλή και κατανοητή δεν θα χρησιμοποιηθούν εξωγενής μεταβλητές για τον προσδιορισμό της τιμής του κινητού τηλεφώνου παρότι θα αναφέρουμε κάποιες οι οποίες θα μπορούσαν να είχαν χρησιμοποιηθεί.

Επίσης πριν ξεκινήσουμε την ανάλυση των χρονολογικών σειρών θα πρέπει να αναφερθούμε στον δείκτη (index) της μεταβλητής οπού θέλουμε να μοντελοποιησουμε γενικά ο συνολικός χρόνος σε μια χρονοσειρα μετρημένη σε έτη ημέρες λεπτά η ακόμα και δευτερόλεπτα συμβολίζεται με T κεφάλαιο ενώ εάν θέλουμε να αναφερθούμε σε μια μόνο

παρατήρηση χρησιμοποιούμε ως δείκτη το μικρό τ και πιο συγκείμενα εάν δεν υπάρχει ακριβής τιμή στον δείκτη το μικρό τ συμβολίζει (αναφέρεται) στην πιο πρόσφατη (τελευταία) τιμή της χρονοσειρας. Για να γίνει πιο κατανοητό έχει δημιουργηθεί η παρακάτω χρονοσειρα με μόνο τέσσερεις παρατηρήσεις των μηνιαίων πωλήσεων κινητών τηλέφωνων ενός καταστήματος λιανικών πωλήσεων. Με βάση αυτά που είπαμε πιο πάνω εάν ο αναλυτής θέλει να αναφερθεί στην τελευταία τιμή της χρονοσειρας τότε το σύμβολο είναι αυτό Y_{τ} και ανφερεται στις πωλησεις του απρηλιου που ηταν ισες με 200 εάν τωρα θελει να αναφερθει στις πωλησει του μαρτιου τοτε χρισισιμοποιει την παρακατω εκφραση $Y_{\tau-1}$ το ιδιο ισχυει και για τον Φεβρουάριο $Y_{\tau-2}$ αν από την άλλη πλευρα θελει να αναφερθει σε προβλεψεις δηλαδή έστω ότι θέλει να προβλέψει τις πωλήσεις του Μάιου η έκφραση που θα χρησιμοποιήσει είναι η παρακάτω $Y_{\tau+1}$ και το ιδιο ισχυει για οποιαδηποτε άλλη προβλεψη πχ του ιουλιου του 2025 θα συμβολιστει με $Y_{\tau+3}$ καθως είναι τρεις μηνες μετα την τελυταια τιμη της χρονοσειρας.

Χρονος (Τ)	Μηνιαίες πωλήσεις κινητών τηλέφωνων
31/1/2025	650
31/2/2025	400
31/3/2025	350
31/4/2025	200

Επίσης επειδή σε αυτό το κεφάλαιο θα χρησιμοποιήσουμε μοντέλα όπως το αυτοπαληνδρομο καλό είναι να εξηγήσουμε ότι το εκάστοτε $Y_{\tau-1}$ όταν εκφραζεται ως ερμηνευτικη μεταβλητη της εξαρτιμενης μεταβλητής είναι στην πραγματικότητα η ιδιά η εξαρτημένη μεταβλητή αλλά εκφρασμένη με μια χρονική καθυστέρηση (lag).

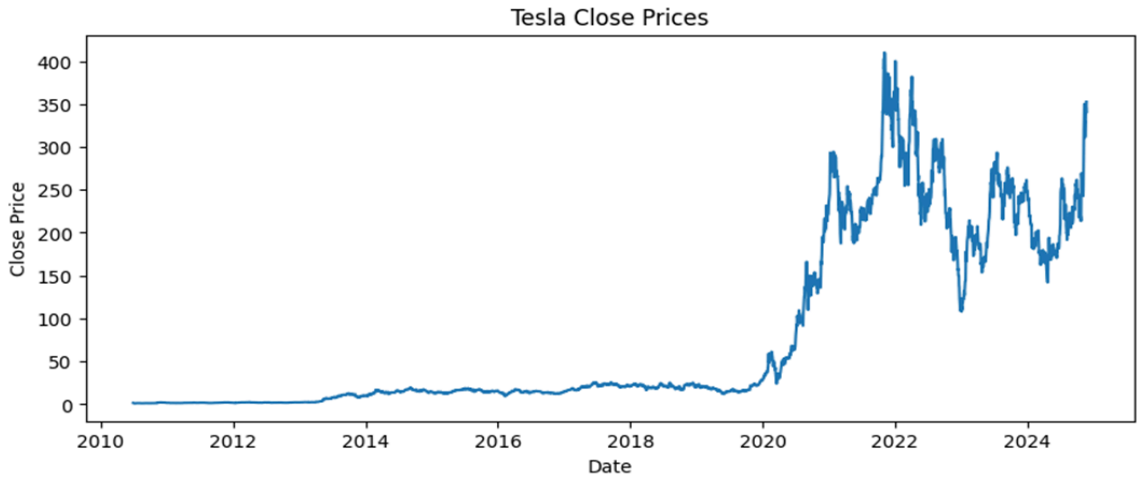
$$AR(P)\, Yt = d0 + d1 * Y_{\tau-1} + d2 * Y_{\tau-2} + \ldots + dnY_{T-P} + \varepsilon\tau$$

για παράδειγμα η lagged εκδοχή της χρονοσειρας των πωλήσεων κινητών τηλεφώνων παρουσιάζεται στον παρακάτω πίνακα.

Χρονος(Τ)	Μηνιαίες πωλήσεις κινητών τηλέφωνων(Yt)	$Y_{\tau-1}$	$Y_{\tau-2}$

31/1/2025	650	-	-
31/2/2025	400	650	-
31/3/2025	350	400	650
31/4/2025	200	350	400

Έχοντας αναλύσει τα παραπάνω θα αναφερθούμε στο τι είναι μια χρονοσειρα. Μια χρονοσειρα αποτελείται από ένα σύνολο δεδομένων τα οποία έχουν συλλεχθεί σε διαδοχικές χρονικές στιγμές κατά την πάροδο του χρόνου. Οι χρονοσειρες χωρίζονται σε δυο κατηγορίες η πρώτη είναι οι συνέχεις οπού τα δεδομένα συλλέγονται συνέχεια κατά την πάροδο του χρόνου όπως για παράδειγμα η κατανάλωση ηλεκτρικής ενέργειας με έναν έξυπνο δείκτη μέτρησης του ή για παράδειγμα η θερμοκρασία ενός δωμάτιου ή ενός έξυπνου ψυγείου , ή ακόμα και ο καρδιακός παλμός ενός αθλητή που φοράει ένα έξυπνο ρολόι στο χέρι του. Ενώ στην δεύτερη κατηγορία ανήκουν οι διακριτές χρονοσειρες οπού τα δεδομένα συλλέγονται ανά τακτά προκαθορισμένα χρονικά διαστήματα που ορίζει ο εκάστοτε αναλυτής για παράδειγμα η καταγραφή της μεταβολής του πληθυσμού της Ελλάδας καταγράφεται ανά δέκα έτη ,οι πωλήσεις ενός κινητού τηλεφώνου μπορούν να συλλέγονται ανά μηνά , οι τιμές κλεισίματος μιας μετοχής συλλέγονται ανά ημέρα.



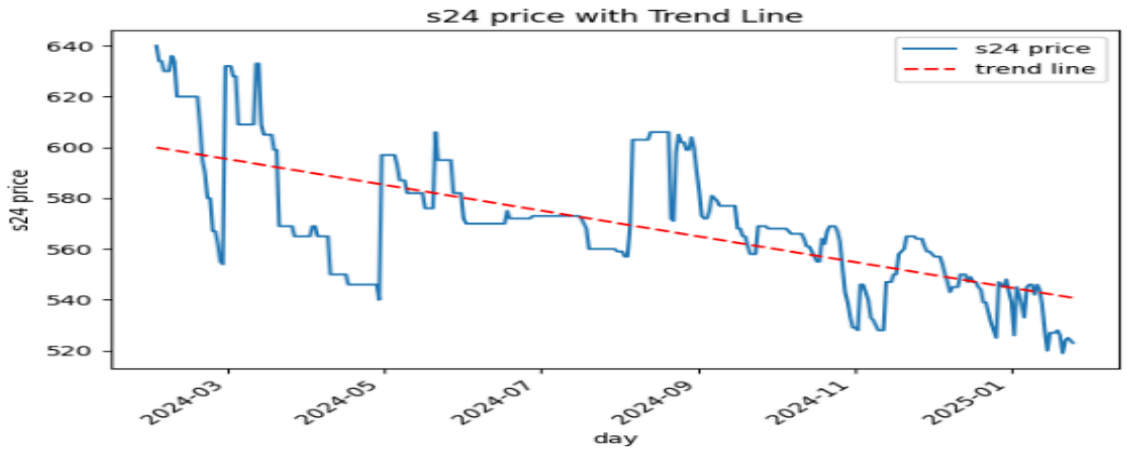
Εικονα 107: time series of tesla stock close prices

5.1 ΧΑΡΑΚΤΙΡΙΣΤΙΚΑ ΧΡΟΝΟΛΟΓΙΚΩΝ ΣΕΙΡΩΝ

5.1.1 ΤΑΣΗ

Ένα από τα σημαντικότερα χαρακτηριστικά των χρονολογικών σειρών είναι η τάση των δεδομένων δηλαδή η μακροπρόθεσμη κλίση / κατεύθυνση της χρονολογικής σειράς.

Ένας άλλος τρόπος με τον οποίο μπορούμε να εκφράσουμε την τάση είναι όταν για ένα μεγάλο χρονικό διάστημα οι τιμές της μεταβλητής έχουν μεγάλη απόσταση οπό την μέση τιμή της χρονοσειρας δηλαδή όταν οι τιμές είναι μεγαλύτερες από τον μέσο ορό τότε υπάρχει ανοδική τάση όταν οι τιμές είναι για μεγάλο χρονικό διάστημα μικρότερες από τον μέσο ορό λεμέ ότι η υπάρχει πτωτική τάση. Προφανώς υπάρχουν πόλοι παράγοντες που επηρεάζουν την τάση ενός προϊόντος όπως οικονομικοί παράγοντες για παράδειγμα όταν το εισόδημα των καταναλωτών μειώνεται τότε μειώνονται και οι δαπάνες τους για διαφορά αγαθά , ακόμα υπάρχουν και περιβαλλοντικοί παράγοντες για παράδειγμα στην Ελλάδα πλέον λόγο της κλιματικής αλλαγής κατά την διάρκεια του χειμώνα οι θερμοκρασίες δεν είναι τόσο χαμηλές όσο τα προηγούμενα έτη με αποτέλεσμα η ζήτηση για πετρέλαιο θέρμανσης να είναι μειωμένη, επίσης υπάρχουν και τεχνολογικοί σε συνδυασμό με πολίτικους και περιβαλλοντικούς παράγοντες για παράδειγμα η συνολική τάση των χωρών της ευρωπαϊκής ένωσης είναι η μείωση των εκπομπών αέριων θερμοκηπίου στην ατμόσφαιρα από τα αυτοκίνητα με κινητήρες εσωτερικής καύσης αυτό έχει ως αποτέλεσμα οι πωλήσεις αυτοκίνητων με βενζινοκινητήρας και πετρελαιοκινητήρες να μειωθούν και αντίθετος οι πωλήσεις των ηλεκτρικών και hybrid οχημάτων να αυξηθούν. Παρακάτω βλέπουμε την πτωτική πορεία της χρονοσειρας της τιμής του Samsung galaxy s24 οπού θα αναλύσουμε στις επόμενες παραγράφους.

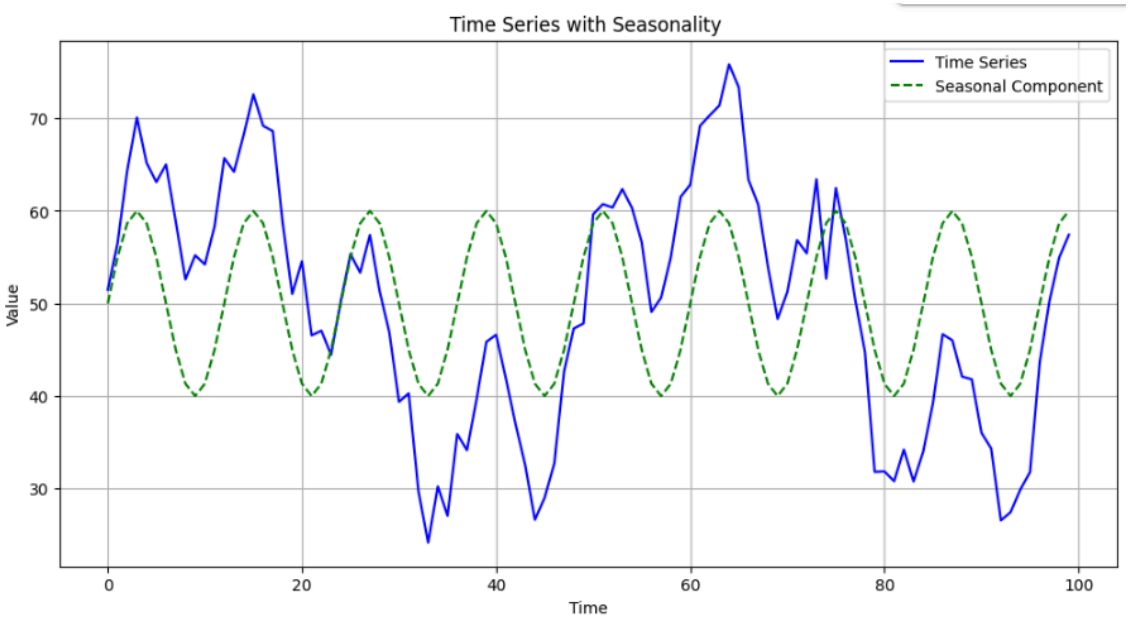


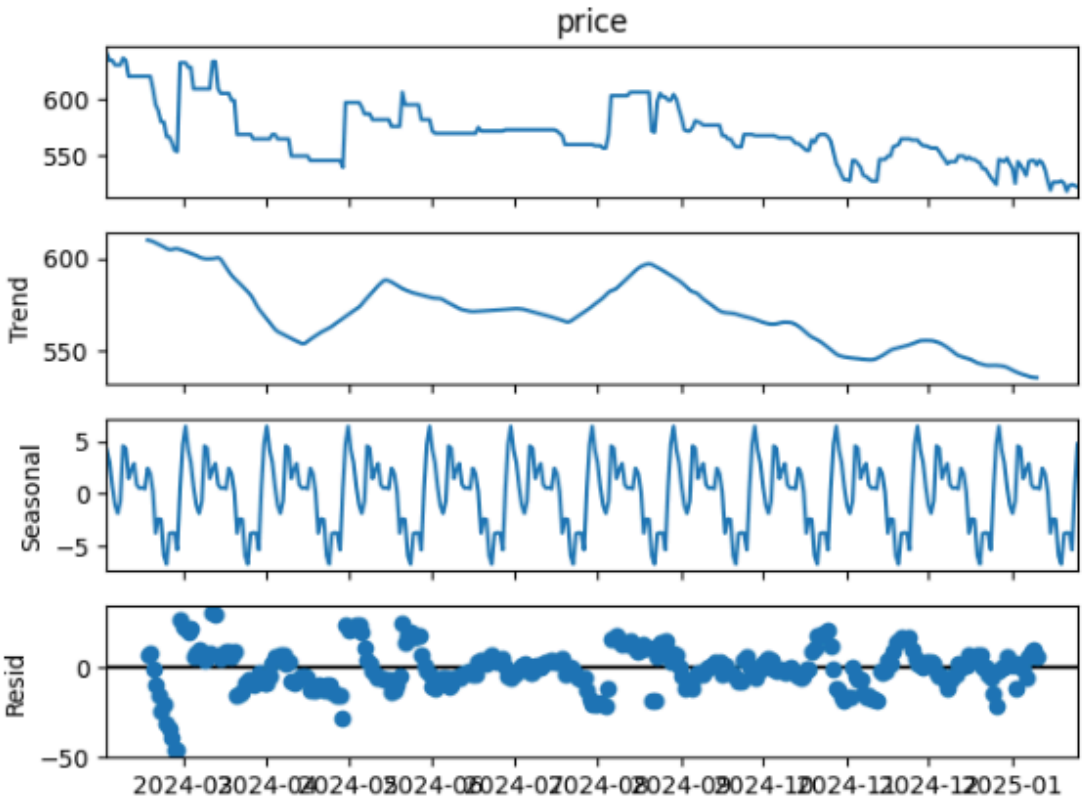
Εικονα 108: τάση τιμής s24

5.1.2 ΕΠΟΧΙΚΟΤΗΤΑ ΚΑΙ ΚΥΚΛΙΚΟΤΗΤΑ

Ο ορός εποχικότητα στις χρονοσειρες αναφέρεται σε βραχυχρόνιες γνωστές εκ των προτέρων από τους αναλυτές και επαναλαμβανόμενες αυξήσεις ή μειώσεις της μεταβλητής της χρονοσειρας. Για παράδειγμα οι πωλήσεις ηλεκτρικών συσκευών αυξάνονται κατά

κύριο λόγο πριν και κατά την περίοδο των Χριστουγέννων λόγο των γιορτών αλλά και των εκπτώσεων στις τιμές των προϊόντων πχ black Friday cyber Monday κλπ. ή για παράδειγμα επηρεάζονται από άλλους κοινωνικούς και θρησκευτικούς παράγοντες για παράδειγμα η πωλήσεις τροφίμων από σόγια έχουν αυξημένη ζήτηση στην Ελλάδα κατά την περίοδο της νηστείας πριν από το Πάσχα και ειδικά κατά την διάρκεια της μεγάλης εβδομάδας , επίσης επηρεάζονται από τις καιρικές συνθήκες για παράδειγμα η ζήτηση παγωτών και οι ενοικιάσεις καταλυμάτων σε ξενοδοχειακές μονάδες των νησιών του αιγαίου αυξάνονται κατά την περίοδο του καλοκαιριού , επίσης εποχικότητα μπορεί να δημιουργηθεί και από οικονομικούς παράγοντες για παράδειγμα κατά τα Χριστούγεννα οι εργαζόμενοι στον ιδιωτικό τομέα λαμβάνουν από τους εργοδότες τους έναν επιπλέον μισθό αυτό το επιπλέον εισόδημα έχει ως αποτέλεσμα η ζήτηση για προϊόντα να αυξάνεται. Αρά από τα παραπάνω παραδείγματα γίνεται κατανοητό ότι η εποχικότητα είναι εύκολο να προβλεφθεί και οι επιχειρήσεις ανάλογος το κλάδο είναι εύκολο να πάρουν τις κατάλληλες αποφάσεις ώστε να αντιμετωπίσουν την αυξημένη ή μειωμένη ζήτηση πχ με την διακρατική μεγαλύτερης ποσότητας αποθεμάτων και την πρόσληψη εποχικών εργαζομένων στην περίπτωση της αύξησης της ζήτησης και το αντίθετο στην περίπτωση μείωσης της ζήτησης προφανώς δεν μπορεί να είναι γνωστό εκ τον προτέρων το ύψος της αύξησης ή της μείωσης της ζήτησης από την μια περίοδο στην άλλη για παράδειγμα σιγουρά η ζήτηση για αγορά νέας τηλεόρασης τα Χριστούγεννα λόγο των εκπτώσεων ,του γιορτινού κλήματος και του αυξημένου εισοδήματος θα είναι μεγαλύτερη σε σχέση με την ζήτηση τον μηνά Φεβρουάριο αλλά δεν γνωρίζουμε εκ τον προτέρων ποσό μεγαλύτερη ή μικρότερη θα είναι η ζήτηση για τηλεοράσεις τα επόμενα Χριστούγεννα καθώς υπάρχουν και άλλοι παράγοντες που μπορεί να επηρεάζουν την ζήτηση. Παρακάτω παρατίθεται ένα παράδειγμα εποχικότητας σε τυχαία δεδομένα ενώ στην συνέχεια με την χρήση του seasonal decompose βλέπουμε αναλυτικά το trend και το seasonality της χρονοσειρας με τις τιμές του Samsung galaxy s24

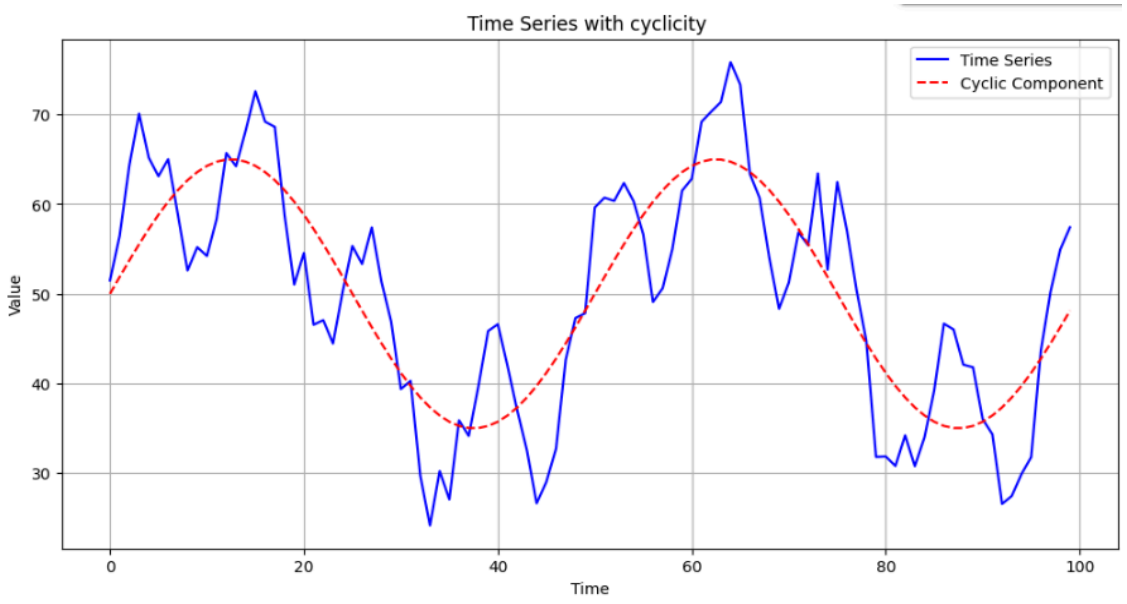




Εικονα 109: παράδειγμα εποχικότητα

Σε αντίθεση με την εποχικότητα οπού γνωρίζουμε εκ των προτέρων την διάρκεια και την μεταβολή της ζήτησης (αύξηση ή μείωση) το φαινόμενο της κυκλικότητας αναφέρεται σε καταστάσεις οπού η ζήτηση αυξάνεται ή μειώνεται μακροπρόθεσμα χωρίς να μπορεί ο αναλυτής να την προσδιορίσει εκ των προτέρων ούτε ως προς το χρονικό διάστημα ούτε

ως προς την διακύμανση αυτής της μεταβολής. Τις περισσότερες φορές η κυκλικότητα προκύπτει από αλλαγές στην παγκόσμια οικονομία ένα τέτοιο παράδειγμα είναι η χρηματοοικονομική κρίση του 2008 οπού σε άλλες χώρες όπως τις ηνωμένες πολιτείες της Αμερικής η κρίση διήρκησε λίγα μόνο χρονιά σε αντίθεση με την Ελλάδα οπού χαριστικέ τουλάχιστον μια δεκαετία για να την ξεπεράσει και ακόμα το ΑΕΠ της δεν έχει φτάσει σε επίπεδα προ κρίσης καθώς όπως μπορούμε να αντιληφθούμε μια κρίση οδηγεί σε μείωση της ζήτησης για προϊόντα , αυτή η μείωση της κατανάλωσης έχει ως αποτέλεσμα την μείωση των επενδύσεων των επιχειρήσεων σε νέα εργοστάσια κατασκευής προϊόντων και μάλιστα πολλές φορές οδηγεί σε κλείσιμο των ήδη υπαρχόντων εργοστασίων το παραπάνω με την σειρά του αυξάνει την ανεργία το αποτέλεσμα της αύξησης της ανεργίας είναι η μείωση του εισοδήματος και η περετέρω μείωση της ζήτησης και αρά η οικονομία οδηγείται σε ένα φαινόμενο που στην μακροοικονομική θεωρία ονομάζεται στασιμοπληθωρισμός δηλαδή στην οικονομία επικρατούν υψηλές τιμές και το ΑΕΠ μειώνεται. Προφανώς αλλά τέτοιου είδους παράγοντες που οδηγούν σε μακροπρόθεσμη μείωση της ζήτησης και δεν μπορούν να προβλεφθούν ούτε για την διάρκεια ούτε για το μέγεθος της μεταβολής που θα προκαλέσουν στην ζήτηση ήταν επίσης ο covid και η καραντίνα που επιβλήθηκε το 2022 , επίσης ο πόλεμος μεταξύ Ουκρανίας και Ρωσίας ανήκει στην ιδιά κατηγορία καθώς έκανε δύσκολη την μεταφορά των προϊόντων με αποτέλεσμα τα κόστη μεταφοράς να αυξηθούν και κατά επέκταση αυτή η αύξηση να μεταφερθεί στην τελική τιμή των προϊόντων. Ένα ακόμα πιο πρόσφατο γεγονός που τα αποτελέσματα(συνέπειες) του θα φανούν μακροπρόθεσμα είναι η επιβολή δασμών στα εξαγόμενα προϊόντα του Καναδά, του Μεξικού της κινας και των χωρών της ευρωπαϊκής ένωσης από τις Ηνωμένες πολιτείας.



Εικονα 110: παράδειγμα κυκλικότητας

5.1.3 ΤΥΧΑΙΑ ΔΙΑΔΡΟΜΗ

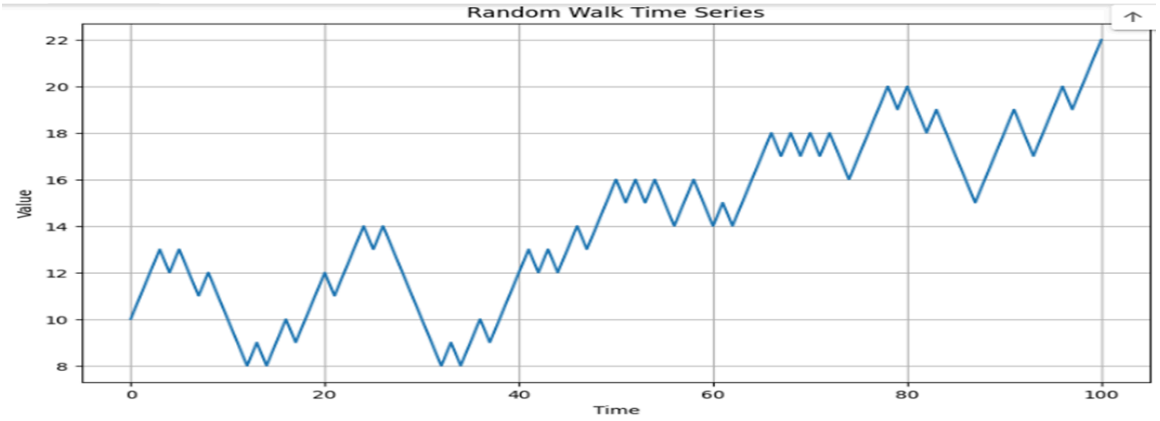
Η τυχαία διαδρομή σε μια χρονοσειρά αναφέρεται σε μια διαδικασία όπου η τιμή της χρονοσειράς μεταβάλλεται τυχαία, χωρίς να ακολουθεί κάποιο προβλέψιμο μοτίβο ή τάση. Κάθε νέα τιμή στην τυχαία διαδρομή εξαρτάται από την προηγούμενη τιμή αλλά η μεταβολή αυτή είναι τυχαία και δεν έχει κάποιο συγκεκριμένο μοτίβο που να επαναλαμβάνεται ανά κάποια συγκεκριμένη χρονική περίοδο όπως για παράδειγμα η εποχικότητα τα Χριστούγεννα. ο ποιο απλός τρόπος για να φτιάξουμε ένα dataset που είναι τυχαία διαδρομή είναι να ξεκινήσουμε από ένα σημείο χ0 και στην συνέχεια να προσθέσουμε σε αυτό ένα σφάλμα που παίρνει την τιμή -1 και 1 με πιθανότητα να συμβεί το καθένα από τα δυο ίση με 50% δηλαδή σαν να ρίχνουμε ένα νόμισμα. Τα χαρακτηριστικά της τυχαίας διαδρομής είναι ότι δεν έχει σταθερή τάση δηλαδή σε κάποιες περιόδους αυξάνεται και σε άλλες μειώνετε επίσης δεν έχει σταθερή διακύμανση και τέλος υπάρχει ισχυρή αυτοσυσχετιση με την προηγουμένη τιμή της χρονοσειρας(yt-1). Ο λόγος που αναφερόμαστε στο φαινόμενο της τυχαίας διαδρομής είναι καθώς αν μια χρονοσειρα μοιάζει με τυχαία διαδρομή αυτό σημαίνει ότι οι προβλέψεις οποιοδήποτε μοντέλου δεν θα είναι αποτελεσματικές και ότι η καλύτερη πρόβλεψη για της μελλοντικές τιμές είναι η τελευταία γνωστή τιμή της χρονοσειρας. Για παράδειγμα πόλοι οικονομολόγοι θεωρούν ότι οι τιμές κλεισίματος των μέτοχων στο χρηματιστήριο είναι χρονοσειρες random walk δηλαδή λόγο της υψηλής διακύμανσης και των πολλών

παραγόντων που επηρεάζουν τις τιμές τους η μονή μεταβλητή που είναι κατάλληλη για να προβλέψει την επόμενη τιμή της μετοχής είναι η τελευταία διαθέσιμη τιμή της.

$$x0 = \text{μια τυχαια τιμη}$$

$$\chi1 = \chi0 + \varepsilon1 \text{ οπου } \varepsilon\{-1, +1\}$$

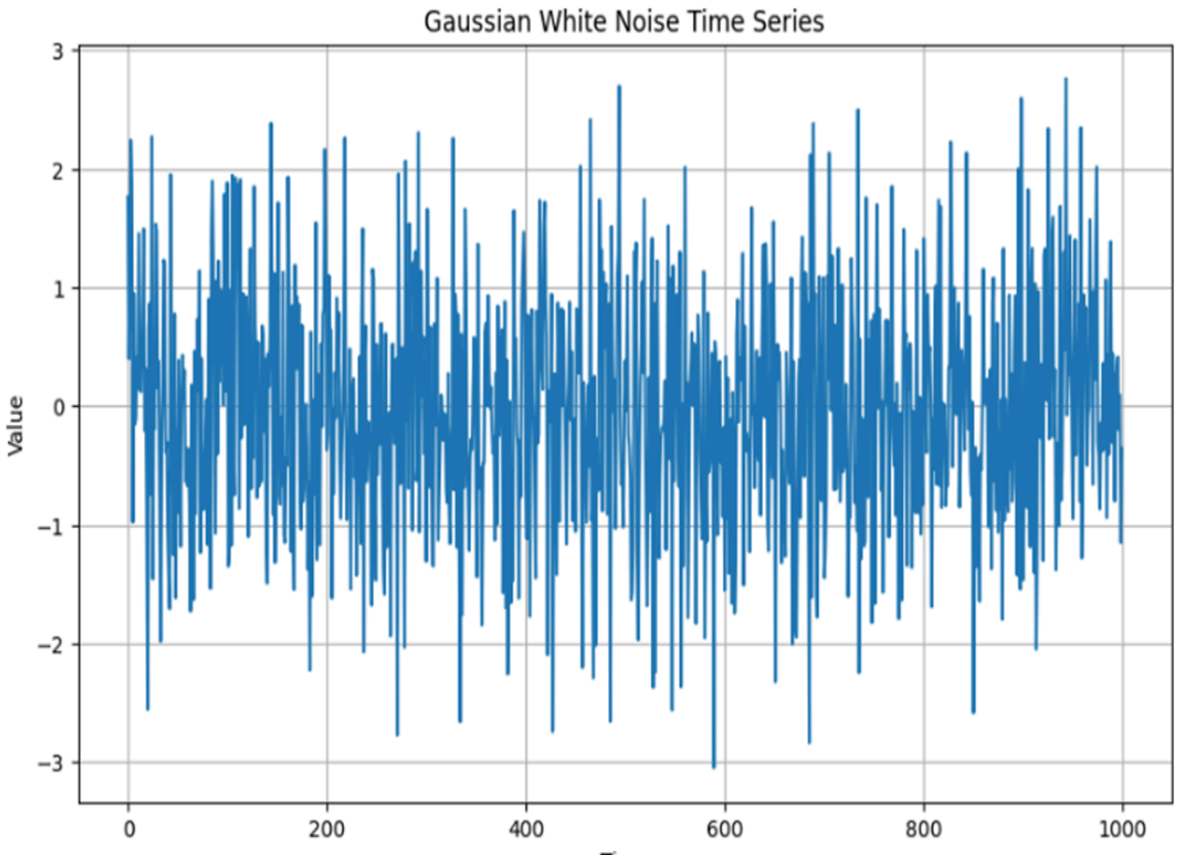
$$\chi2 = \chi1 + \varepsilon2$$



Εικονα 111: random walk time serie

5.1.4 ΛΕΥΚΟΣ ΘΩΡΥΒΟΣ

Ο λόγος για τον οποίο αναφερόμαστε στον λευκό θόρυβο είναι επειδή χρησιμοποιείται ως το σφάλμα (e) σε πολλά μοντέλα χρονοσειρών. Ο λευκός θόρυβος αναφέρεται σε μια χρονοσειρά δεδομένων που δεν ακολουθεί κανένα σαφές μοτίβο ή τάση, με αποτέλεσμα οι προβλέψεις που προκύπτουν από αυτά τα δεδομένα να μην έχουν καμία στατιστική σημασία. Για παράδειγμα, αν κάποιος προσπαθήσει να δημιουργήσει ένα σύνολο δεδομένων από ρίψεις ενός ζαριού και να προβλέψει τον αριθμό που θα προκύψει στην επόμενη ρίψη, η πρόβλεψη δεν θα έχει καμία αξία, καθώς η πιθανότητα για κάθε αριθμό είναι ίση με (1/6), και η προηγούμενη ρίψη δεν επηρεάζει την επόμενη.Πιο συγκεκριμένα, λέμε ότι μια χρονοσειρά είναι λευκός θόρυβος όταν η μέση τιμή και η διακύμανση των δεδομένων παραμένουν σταθερές, ενώ δεν υπάρχει αυτοσυσχέτιση (δηλαδή η συσχέτιση μεταξύ των τιμών της χρονοσειράς σε διαδοχικές χρονικές στιγμές είναι μηδενική(corr(yt,yt-1)=0). Αυτό σημαίνει ότι δεν υπάρχει στατιστική σχέση ή εξάρτηση ανάμεσα στις τιμές της χρονοσειράς σε διαφορετικές χρονικές στιγμές.



Εικονα 112: white noise time series

Όπως βλέπουμε στο παραπάνω διάγραμμα η χρονοσειρα δεν έχει τάση (καθώς τα δεδομένα έχουν μέση τιμή το μηδέν)και οι αυξομειώσεις (volatility/variance) είναι ισόποσες(καθώς τα δεδομένα έχουν διακύμανση ίση με ένα). Και όσο αφορά την αυτοσυσχετιση την οποία θα υπολογίσουμε αναλυτικά στην επόμενη ενότητα βλέπουμε ότι δεν υπάρχει για τον λευκό θόρυβο σε κανένα lag(καθυστέρηση).

5.1.5 ΑΥΤΟΣΥΣΧΕΤΙΣΗ ΚΑΙ ΜΕΡΙΚΗ ΑΥΤΟΣΥΣΧΕΤΙΣΗ

Όπως αναλύσαμε και στο κεφάλαιο 2, η συσχέτιση μεταξύ δύο μεταβλητών μετρά τον βαθμό στον οποίο οι τιμές τους κινούνται μαζί (προς την ιδιά ή την αντίθετη κατεύθυνση). Για παράδειγμα, η σχέση μεταξύ του μισθού ενός εργαζομένου και των ετών προϋπηρεσίας του μπορεί να εκφραστεί με τον συντελεστή συσχέτισης, ο οποίος κυμαίνεται από -1 έως 1. Όταν η τιμή του είναι κοντά στο 1, σημαίνει ότι υπάρχει ισχυρή θετική συσχέτιση δηλαδή, καθώς αυξάνονται τα έτη προϋπηρεσίας, αυξάνεται και ο μισθός. Αντίθετα, τιμές κοντά στο -1 υποδεικνύουν αρνητική συσχέτιση, δηλαδή οι δύο μεταβλητές κινούνται αντίθετα. Όταν η τιμή είναι 0, τότε οι δύο μεταβλητές θεωρούνται ανεξάρτητες, χωρίς καμία στατιστική σχέση μεταξύ τους. Η ίδια αρχή εφαρμόζεται και

στις χρονοσειρές, αλλά αντί να εξετάζουμε τη σχέση μεταξύ δύο διαφορετικών μεταβλητών, μελετάμε τη σχέση μιας χρονοσειράς με τις προηγούμενες τιμές της. Αυτή η μέτρηση ονομάζεται αυτοσυσχέτιση (autocorrelation) και μας βοηθά να κατανοήσουμε αν και πώς οι προηγούμενες τιμές μιας χρονοσειράς επηρεάζουν τις τρέχουσες τιμές της. Για παράδειγμα, σε μια χρονοσειρά που αφορά τις μηνιαίες πωλήσεις ενός προϊόντος, η αυτοσυσχέτιση μπορεί να μας δείξει αν οι πωλήσεις του προηγούμενου μήνα σχετίζονται με τις πωλήσεις του τρέχοντος μήνα. Ο υπολογισμός της αυτοσυσχέτισης γίνεται συγκρίνοντας την αρχική χρονοσειρά με προηγούμενες εκδόσεις της, όπου κάθε μια (lag) αντιστοιχεί σε μια χρονική καθυστέρηση. Αν για ένα συγκεκριμένο lag παρατηρείται υψηλός συντελεστής αυτοσυσχέτισης, σημαίνει ότι η χρονοσειρά εμφανίζει μοτίβα που επαναλαμβάνονται με συγκεκριμένη χρονική καθυστέρηση. Μέσο της αυτοσυσχετισης μπορεί ο αναλυτής να αναγνωρίσει επαναλαμβανόμενα μοτίβα στις χρονοσειρες όπως για παράδειγμα εποχικά φαινόμενα όπως την αύξηση της ζήτησης τα Χριστούγεννα. Επίσης μέσο της αυτοσησχετισης καθορίζεται ο αριθμός των χρονικών καθυστερήσεων που πρέπει ο αναλυτής να συμπεριλάβει στα μοντέλα του κινητού μέσου ορού (ma) και κατά επέκταση στα μοντέλα Arma και Arima καθώς μέσα από αυτό προσδιορίζεται ο συντελεστής q των παραπάνω μοντέλων. Ο τύπος με τον οποίο μπορεί να υπολογιστεί η αυτοσυσχετιση μεταξύ των προηγουμένων εκδόσεων μιας χρονοσειρας είναι ο παρακάτω:

$$\rho_{\kappa} = \frac{cov\,(Y_{t'}, Y_{t-k})}{s_{Yt}{}^{*}\, s_{Yt-k}}$$

Στην συνέχεια ακολουθεί ένα παράδειγμα υπολογισμού της αυτοσυσχετισης στο excel με την χρήση της συνάρτησης corell.

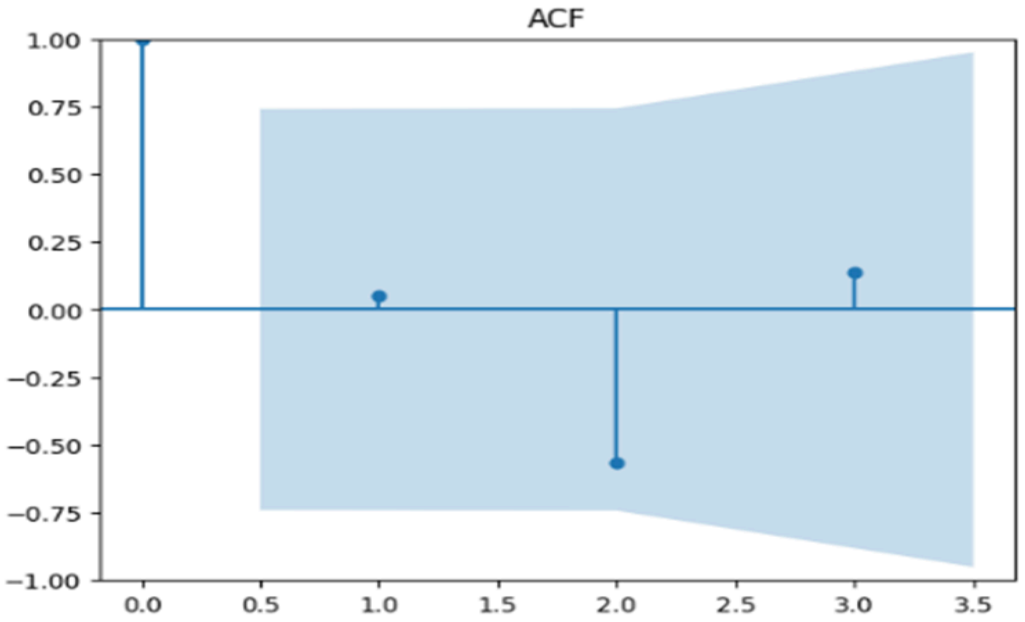
t	yt	yt-1	yt+2	yt-3
1	5			
2	15	5		
3	25	15	5	
4	17	25	15	5
5	12	17	25	15
6	21	12	17	25
7	24	21	12	17
		24	21	12
			24	21
				24
correlation		-0,02884	-0,5884	0,198861

Εικονα 112: υπολογισμός αυτοσυσχετισης

Προφανώς οι περισσότερες χρονοσειρες δεν αποτελούνται μόνο από 7 τιμές αλλά από μεγάλο πλήθος δεδομένων και ταυτόχρονα η αυτοσυσχετιση δεν υπολογίζεται μόνο για τρεις καθυστερήσεις αλλά για πολλές περισσότερες για αυτό το λόγο είναι προτιμότερο ο υπολογισμός των παραπάνω να πραγματοποιείται μέσω της χρήσης της python και πιο συγκεκριμένα με την δημιουργία ενός acf collorgram.

```
import pandas as pd
import matplotlib.pyplot as plt
from statsmodels.graphics.tsaplots import plot_acf
data = {'values': [5, 15, 25, 17, 12, 21, 24]}
df = pd.DataFrame(data, index=[1, 2, 3, 4, 5, 6, 7])
plt.figure(figsize=(10, 6))
plot_acf(df['values'], lags=3, alpha=0.05)
plt.title("ACF")
plt.show()
```

```
<Figure size 1000x600 with 0 Axes>
```



Εικονα 113: acf plot

Για να προσδιορίσουμε αν ένας συντελεστής αυτοσυσχέτισης είναι στατιστικά σημαντικός, χρησιμοποιούμε το t-test. Σύμφωνα με τον Bartlett, οι συντελεστές αυτοσυσχέτισης ακολουθούν κανονική κατανομή με μέσο όρο 0 και διακύμανση 1/N, όπου N είναι το μέγεθος του δείγματος της χρονοσειράς. Ο στατιστικός έλεγχος βασίζεται στις εξής υποθέσεις: Μηδενική υπόθεση (H_0): Ο συντελεστής αυτοσυσχέτισης είναι ίσος με μηδέν, δηλαδή δεν υπάρχει συσχέτιση μεταξύ των χρονικών σημείων της σειράς. Εναλλακτική υπόθεση (H_1): Ο συντελεστής αυτοσυσχέτισης είναι διαφορετικός από το μηδέν, γεγονός που υποδηλώνει ότι υπάρχει κάποια στατιστικά σημαντική σχέση μεταξύ των διαδοχικών παρατηρήσεων. Για επίπεδο σημαντικότητας $\alpha = 0,05$ (που αντιστοιχεί σε 95% επίπεδο εμπιστοσύνης) και για μεγάλο δείγμα ($N > 50$), το t-statistic προσεγγίζει την τιμή 2, καθώς πραγματοποιούμε δίπλευρο έλεγχο. Έτσι, το διάστημα εμπιστοσύνης για τον συντελεστή αυτοσυσχέτισης $\hat{\rho}$ διαμορφώνεται ως εξής.

$$\hat{\rho} \pm 2 * 1/\sqrt{N}$$

Αρά αν ένας συντελεστής αυτοσυσχετισης είναι έξω από το διάστημα εμπιστοσύνης τότε είναι στατιστικά σημαντικός. Προφανώς στο παραπάνω παράδειγμα οι παρατηρήσεις είναι μόνο 7 οπότε και το τ στατιστικό είναι πολύ μεγάλο και αρά όλοι οι συντελεστές αυτοσυσχετισης είναι στατιστικά ασήμαντοι.

Αναφορικά τη μερική αυτοσυσχετιση(pacf) αυτή χρησιμοποιείται για να βρεθεί ο συντελεστής p στα ar, Arma,Arima μοντέλα η διαφορά μεταξύ του συντελεστή μερικής αυτοσυσχετισης σε σύγκριση με τον συντελεστή αυτοσυσχετισης είναι ότι ο πρώτος μετράει αποκλειστικά και μόνο την απευθείας συσχέτιση του Υτ με κάποια καθυστέρηση δηλαδή δεν λαμβάνει υπόψη του την επίδραση των ενδιάμεσων χρονικών σημείων. Υπάρχουν δυο τρόποι υπολογισμού των συντελεστών συσχέτισης του pacf ο πρώτος είναι μέσο της επίλυσης των yule-walker εξισώσεων και ο δεύτερος είναι μέσο της χρήσης της αυτοπαλινδρομισης. Συγκεκριμένα στην αυτοπαλινδρομιση δεν χριζόμαστε την σταθερά (slope/intercept) και αντί να συμβολίζουμε τους συντελεστές β1,β2 κτλ. όπως στην γραμμική παλινδρόμηση τους συμβολίζουμε με φ1,φ2 κλπ.

$$\Upsilon_{\tau} = \varphi 1 * \Upsilon_{\tau-1} + \varphi 2 * \Upsilon_{\tau-2} + \varepsilon\tau$$

Ο τρόπος υπολογισμού των παραπάνω στην python γίνεται είτε μέσο της γραμμικής παλινδρόμησης χωρίς slope ή με της εξισώσεις yule-walker ,βέβαια ο ποιο απλός τρόπος είναι μέσο του pacf collorgram οπού θα δούμε στην συνέχεια για την χρονοσειρα της τιμής του Samsung galaxy s24.

5.1.6 ΣΤΑΣΙΜΟΤΗΤΑ

Το επόμενο χαρακτηριστικό των χρονολογικών σειρών που θα αναλύσουμε είναι η στασιμότητα. Ουσιαστικά η στασιμότητα εκφράζει το γεγονός ότι η κατανομή που ακολουθούν οι τυχαίες μεταβλητές της χρονοσειρας ανήκουν στην ιδιά κατανομή σε όλη την διάρκεια της χρονοσειρας. Ο ορισμός που αναφέρθηκε πιο πάνω αντιστοιχεί στην αυστηρή στασιμότητα για παράδειγμα έστω μια χρονοσειρα με 2000 παρατηρήσεις έστω ότι περνούμε δυο υποσύνολα αυτής με 200 παρατηρήσεις το καθένα έστω από την τιμή Υ(300) έως την παρατήρηση Υ(500) το πρώτο και το δεύτερο από την παρατήρηση

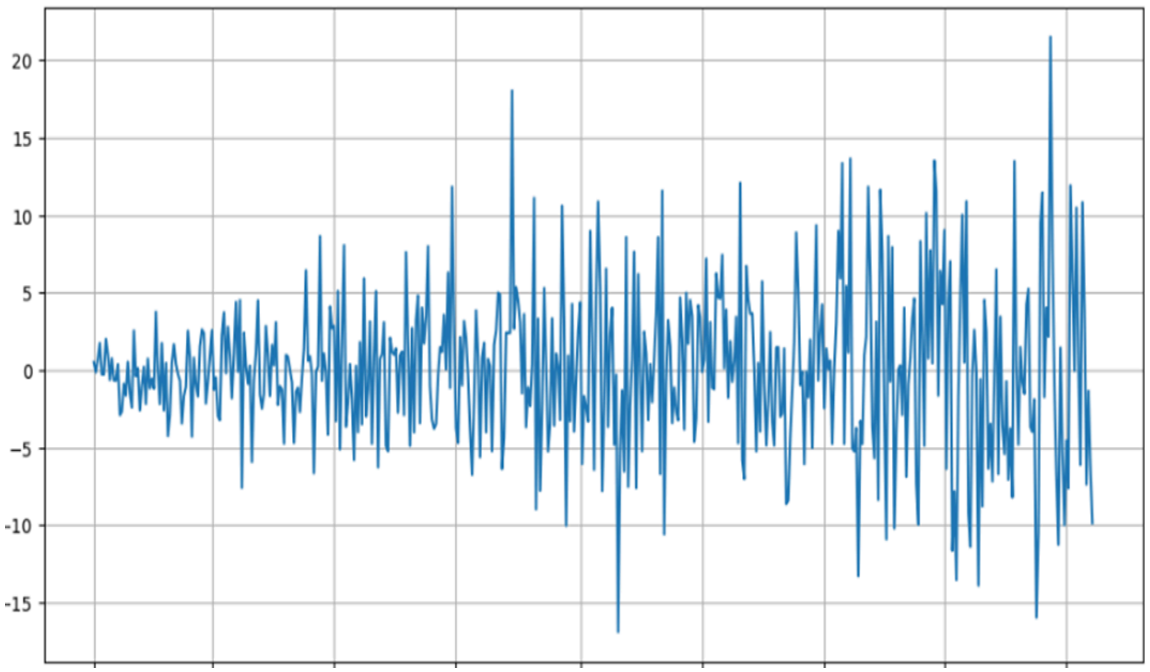
Υ(1200) έως την παρατήρηση Υ(1400) τότε λεμέ ότι η χρονοσειρα είναι αυστηρά στάσιμη όταν η κατανομή(πχ student t ,binomial ,normal κλπ.) του πρώτου υποσυνόλου είναι ιδιά με την κατανομή του δευτέρου υποσυνόλου. Η δεύτερη μορφή στασιμότητας είναι η ασθενώς στάσιμη οπού εκεί για να θεωρηθεί μια χρονοσειρα ως ασθενώς στάσιμη τότε πρέπει να μην μεταβάλετε στην πάροδο του χρόνου ούτε η μέση τιμή των παρατηρήσεων ούτε η διακύμανση τους και ταυτόχρονα η αυτόσυσχέτιση μεταξύ δυο τυχαίων παρατηρήσεων που απέχουν μεταξύ τους τ περιόδους να είναι ίση με την αυτοσυσχετιση δυο άλλων παρατηρήσεων που και αυτές απέχουν τ περιόδους μεταξύ τους. Για παράδειγμα έστω χρονοσειρα με 100 παρατηρήσεις για να θεωρήσουμε ότι είναι ασθενώς στάσιμη θα πρέπει πρώτων η μέση τιμή των παρατηρήσεων να είναι σταθερή το ίδιο να ισχύει και για την διακύμανση τους και τρίτων θα πρέπει η αυτοσυσχέτιση μεταξύ δύο χρονικών στιγμών που απέχουν μεταξύ τους τ περιόδους εξαρτάται μόνο από το **τ** και όχι από την αρχική χρονική στιγμή για παράδειγμα έστω επιλεγμένα στην τύχη Υ που απέχουν μεταξύ τους τ=2 περιόδους τότε για να θεωρηθεί ασθενώς στάσιμη η χρονοσειρα θα πρέπει να ισχύει το COV(Y20,Y23)=COV(Y30,13)=COV(Y40,Y43) αν αυτό ισχύει τότε λεμέ ότι η χρονοσειρα είναι ασθενώς στάσιμη.

$$\mu\Upsilon = \mu\Upsilon(\tau + T) \text{ για ολλες τις περιοδους}$$

$$var(Yt) = E[Yt - E(Yt)]^2 = \sigma\text{^}2_{\Upsilon}$$

$$COV\left(YT, Y_{t+K}\right) = COV\left(Y_{t+m}, Y_{t+m+k}\right) = Y_k \text{ για ολες τις περιοδους τ και ολα τα } la$$

Βέβαια υπάρχει και πιο απλώς τρόπος για να αναγνωρίσει ο αναλυτής αν είναι ασθενώς στάσιμη ή όχι απλός παρατηρώντας την και ελέγχοντας την μέσο του dickey-fuller τεστ. Όταν αναφερόμαστε στην παρατήρηση της χρονοσειρας εννοούμε ότι ο αναλυτής βλέποντας την χρονοσειρα να παρατηρήσει αν υπάρχει τάση στα δεδομένα ή και αν αυξάνεται η μειώνεται η διακύμανση των δεδομένων. Ένα παράδειγμα αυξανομένης διακύμανσης σε μια χρονοσειρα παρατίθεται στην παρακάτω εικόνα.



Εικονα 114: χρονοσειρα με αυξανομένη διακύμανση

Προφανώς ο βέλτιστος τρόπος για να επιβεβαιώσει ο αναλυτής την ύπαρξη ή μη στασιμότητας είναι με το στατιστικό τεστ που ανέπτυξαν οι ερευνητές David dickey και Wayne fuller το οποίο εξετάζει για το αν μια χρονοσειρα έχει μοναδιαια ριζά (αν είναι ασθενώς στάσιμη).Κατά την εκτέλεση αυτού του τεστ στην python δίνονται στον αναλυτή ανάλογα το μέγεθος του δείγματος διάφορες κριτικές τιμές από την κατανομή του student t σε διαφορά επίπεδα σημαντικότητας όπως 90%,95%,99%) μαζί με αυτά δίνονται στον αναλυτή και το adf score (είναι το αντίστοιχο τ στατιστικό) ώστε να μπορέσει να κάνει το έλεγχο υποθέσεων .αναλυτικότερα οι υποθέσεις του ελέγχου είναι οι παρακάτω H0: η χρονοσειρα δεν είναι ασθενώς στάσιμη και H1: η χρονοσειρα είναι ασθενώς στάσιμη, προφανώς στόχος του αναλυτή είναι να αποκρύψει την μηδενική απόφαση. ο στατιστικός έλεγχος που προκύπτει είναι εάν το adf score είναι μικρότερο από τα διαφορά t critical values (έστω ότι είναι μικρότερο του τ στατιστικού σε επίπεδο σημαντικότητας 99%) τότε λεμέ ότι σε επίπεδο 99% είμαστε σίγουροι ότι η χρονοσειρα είναι στάσιμη. Προφανώς ο έλεγχος μπορεί να γίνει και με το p-value με την αντίστοιχη τιμή του alpha. Ας εξετάσουμε την στασιμότητα της χρονοσειρας με τις τιμές του κινητού τηλεφώνου που θα αναλύσουμε διεξοδικά στην επόμενη ενότητα. Έχοντας εισάγει τα δεδομένα χρησιμοποιώντας την βιβλιοθήκη stats models και συγκεκριμένα την adfuller συνάρτηση για να εξετάσουμε την στασιμότητα της.

```
result = adfuller(dfar['price'])
print('ADF Statistic: %f' % result[0])
print('p-value: %f' % result[1])
print('Critical Values:')
for key, value in result[4].items():
    print('\t%s: %.3f' % (key, value))
if result[1] <= 0.05:
    print("Reject the null hypothesis. The time series is stationary.")
else:
    print("Fail to reject the null hypothesis. The time series is non-stationary.")
```

```
ADF Statistic: -3.243182
p-value: 0.017617
Critical Values:
        1%: -3.449
        5%: -2.870
        10%: -2.571
Reject the null hypothesis. The time series is stationary.
```

Εικονα 115: adf test

Όπως παρατηρούμε το adf statistic που προκύπτει(-3,24) είναι μικρότερο από το τ critical value(-2,87) με α=0,05 αρά μπορούμε να απορρίψουμε την μηδενική απόφαση και να οδηγηθούμε στο συμπέρασμα ότι η χρονοσειρα είναι στάσιμη σε επίπεδο σημαντικότητα 95%. Βέβαια δεν ισχύει το ίδιο αν το α που επιλεγεί είναι ίσο με 0,01. Στην περίπτωση όμως που τα δεδομένα δεν ήταν στασιμά καλό θα ήταν ο αναλυτής αν ήθελε να δημιουργήσει προβλέψεις χρησιμοποιώντας μοντέλα όπως το Arima να τα μετατρέψει σε στασιμά, αν όμως στόχος του ήταν να τα μοντελοποιησει με μοντέλα μηχανικής μάθησης όπως το svr δεν απαιτείται κάποια μετατροπή καθώς αυτά το μοντέλα αποδίδουν καλές προβλέψεις ακόμα και χωρίς η χρονοσειρα να είναι στάσιμη. Παρόλα αυτά θα δείξουμε δυο μεθόδους με τις οποίες ο αναλυτής μπορεί να μετατρέψει μια μη στάσιμη χρονοσειρα σε στάσιμη. Ο πρώτη μέθοδος είναι η λογαριθμική των δεδομένων., και η δεύτερη μέθοδος είναι παίρνοντας την πρώτη διαφορά των δεδομένων δηλαδή Υτ – Υτ-1. Παρακάτω παρατίθεται ο κώδικας στην python για την μετατροπή.

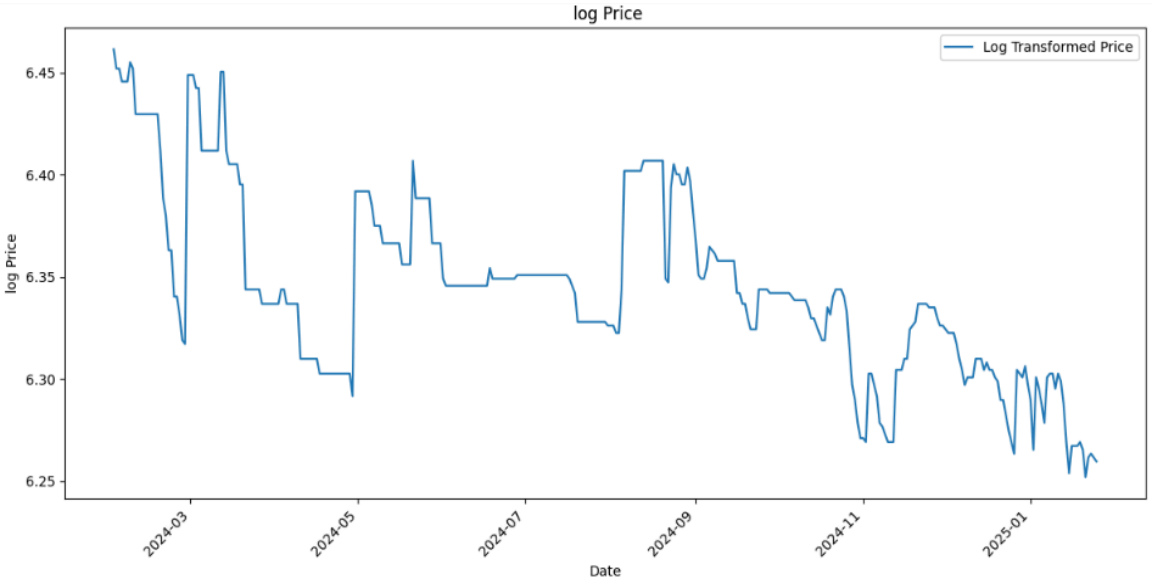
```
dfs24['log_price'] = np.log(dfs24['price'])
dfs24['price_diff'] = dfs24['price'].diff()
```

(το dfs24 είναι αντίγραφο του dfar)

Οι νέες στήλες που δημιουργηθήκαν είναι οι παρακάτω.

date	price	log_price	price_diff
2024-02-02	640	6.461468	NaN
2024-02-03	634	6.452049	-6.0
2024-02-04	634	6.452049	0.0
2024-02-05	630	6.445720	-4.0
2024-02-06	630	6.445720	0.0
...	...	...	...
2025-01-21	519	6.251904	-7.0
2025-01-22	524	6.261492	5.0
2025-01-23	525	6.263398	1.0
2025-01-24	524	6.261492	-1.0
2025-01-25	523	6.259581	-1.0

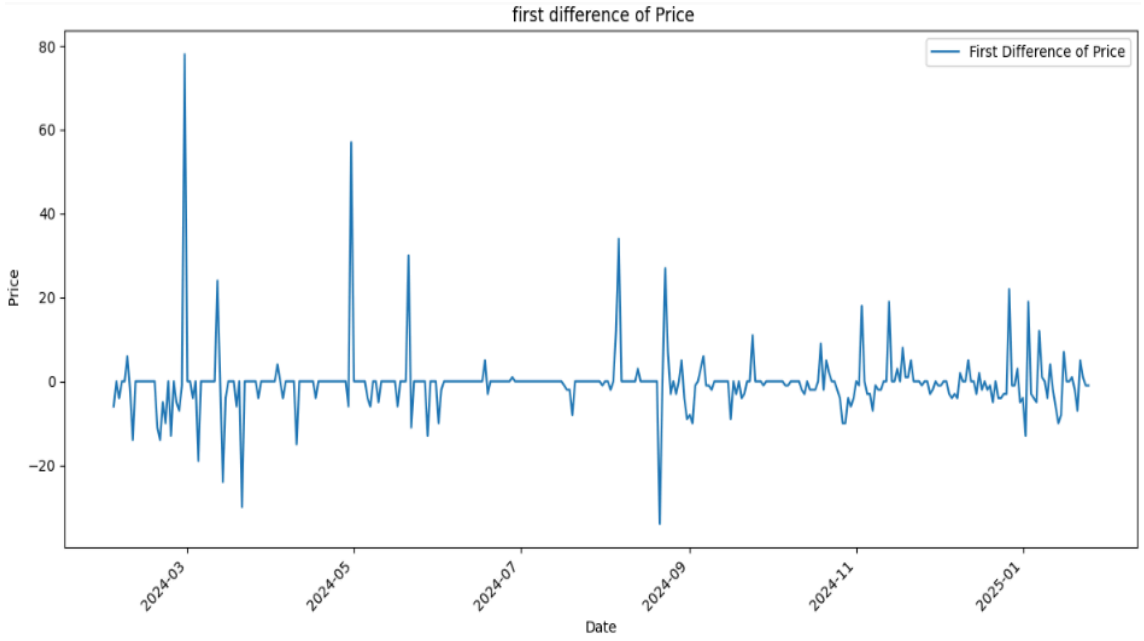
Το διάγραμμα της λογαριθμικής τιμής του κινητού παρουσιάζεται στο παρακάτω plot.



Στην συνέχεια δοκιμάσαμε τον έλεγχο στασιμότητα με το dickey fuller test στην λογαριθμισμενη τιμή και όπως παρατηρούμε παρόλο που είναι στάσιμη με α=0,05 το p-value έχει μεγαλώσει και το ίδιο έχει γίνει και με το adf statistic δηλαδή η μετατροπή έκανε τα δεδομένα λιγότερο στασιμά.

```
ADF Statistic: -3.150106
p-value: 0.023061
Critical Values:
        1%: -3.449
        5%: -2.870
        10%: -2.571
Reject the null hypothesis. The time series is stationary.
```

Στην συνέχεια δοκιμάζουμε το dickey fuller test στα δεδομένα της πρώτης διαφορας. Μπορούμε να παρατηρήσουμε βλέποντας μόνο το διάγραμμα ότι ο μέσος ορός της χρονοσειρας είναι σταθερός σχεδόν ίσος με το μηδέν επίσης η διακύμανση έχει και αυτή εξομαλυνθεί στο μεγαλύτερο μέρος της χρονοσειρας , αρά η χρονοσειρα είναι στάσιμη, βέβαια καλό είναι να το επιβεβαιώσουμε με το dickey fuller test.



Όντως ακόμα και σε α=0,01 η χρονοσειρα είναι στάσιμη το p-value είναι ίσο με μηδέν ενώ το adf statistic είναι πολύ μικρότερο από όλα τα t critical values.

```
ADF Statistic: -18.403097
p-value: 0.000000
Critical Values:
        1%: -3.449
        5%: -2.870
        10%: -2.571
Reject the null hypothesis. The time series is stationary.
```

Εικονα 116: adf test στην πρώτη διαφορά

Παρόλα αυτά στην επόμενη ενότητα θα μοντελοποιησουμε την τιμή στην αρχική της μορφή και η ιδιά μέθοδος θα εφαρμοστεί και στο κεφάλαιο 9 με τις πωλήσεις της εταιρείας πλαίσιο.

5.2 ΕΦΑΡΜΟΓΗ ΜΟΝΤΕΛΩΝ ΧΡΟΝΟΛΟΓΙΚΩΝ ΣΕΙΡΩΝ ΣΤΗΝ ΠΡΟΒΛΕΨΗ ΤΗΣ ΤΙΜΗΣ ΤΟΥ SAMSUNG GALAXY S24

Σε αυτή την ενότητα θα εξετάσουμε θεωρητικά αλλά και πρακτικά διάφορες μεθόδους ανάλυσης χρονολογικών σειρών ξεκινώντας από πολύ απλές όπως το κινήτο μέσο ορό , θα συνεχίσουμε με ποιο συνθέτες όπως την αυτοπαλινδρομιση , και τέλος θα ολοκληρώσουμε την ενότητα με τεχνικές μηχανικής μάθησης όπως το support vector regression και νευρωνικων δικτυών όπως το μοντέλο lstm. Η χρονοσειρα που θα μοντελοποιησουμε αποτελείται από 359 παρατηρήσεις των χαμηλότερων τιμών στην ελληνική αγορά του κινητού τηλεφώνου Samsung s24 στην έκδοση με τα 128 gb μνήμης. Τα δεδομένα αποκτήθηκαν χειροκίνητα μέσο της πλατφόρμας σύγκρισης τιμών best price ο σύνδεσμος παρατίθεται παρακάτω.

https://www.bestprice.gr/item/2159327068/samsung-galaxy-s24-128gb.html

Προφανώς οι τεχνικές που θα περιγράψουμε στην συνέχεια μπορούν να χρησιμοποιηθούν στην πρόβλεψη και άλλων δεδομένων όπως για παράδειγμα τιμές μέτοχων όπως της tesla , της apple, της google κλπ. , επίσης μπορούν να χρησιμοποιηθούν και για μεταβλητές όπως τις πωλήσεις ενός καταστήματος ή ενός συγκεκριμένου προϊόντος και την διαχείριση αποθεμάτων. Επίσης είναι πολύ αποτελεσματικές και σε άλλους κλάδους της οικονομίας πέραν από τον χρηματοοικονομικό όπως για παράδειγμα τον τουριστικό προβλέποντας τους ημερήσιους πελάτες ενός ξενοδοχείου ή τον ημερήσιο αριθμό των επιβατών ενός πλοίου προς ένα νησί των Κυκλάδων και με αυτό τον τρόπο μπορούν να παρθούν αποφάσεις για την αγορά ενός νέου πλοίου για να μπορεί να εξυπηρετηθούν επιπλέον τουρίστες ή να μεταφερθεί ένα πλοίο από μια γραμμή χαμηλής ζήτησης σε μια άλλη υψηλής. Όπως είπαμε και στην αρχή αυτού του κεφαλαίου για οποιαδήποτε μεταβλητή μπορεί ο αναλυτής να συλλέξει διαδοχικές τιμές σε ένα προκαθορισμένο χρονικό διάστημα μπορούν να μοντελοποιηθουν με τεχνικές πρόβλεψης χρονομέτρων ακόμα και το DNA μέσα στα κύτταρα είναι μια χρονοσειρα (συνεχής).

Έχοντας αναφερθεί στα παραπάνω εισάγουμε τα δεδομένα στην python σε ένα pandas dataframe.Όπως είπαμε η χρονοσειρα αποτελείται από 359 παρατηρήσεις των ελάχιστων τιμών από την 2/2/2024 έως την 25/1/2025.

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
```

```
s24=pd.read_excel('s24_128gb_price.xlsx')
```

```
s24
```

	date	price
0	2024-02-02	640
1	2024-02-03	634
2	2024-02-04	634
3	2024-02-05	630
4	2024-02-06	630
...	...	...
354	2025-01-21	519
355	2025-01-22	524
356	2025-01-23	525
357	2025-01-24	524
358	2025-01-25	523

359 rows × 2 columns

Στην συνέχεια θέτουμε ως δείκτη την στήλη με τις ημερομηνίες

```
s24.set_index('date', inplace=True)
```

```
s24
```

date	price
2024-02-02	640
2024-02-03	634
2024-02-04	634
2024-02-05	630
2024-02-06	630
...	...
2025-01-21	519
2025-01-22	524
2025-01-23	525
2025-01-24	524
2025-01-25	523

359 rows × 1 columns

Εν συνέχεια δημιουργούμε έναν πίνακα με κάποια στοιχεία περιγραφικής στατιστικής για παράδειγμα βλέπουμε ότι η μέση τιμή της χρονοσειρας είναι 570 ευρώ η τυπική απόκλιση ίση με 25,8 η ελάχιστη τιμή για όλη αυτή την χρονική περίοδο ίση με 519 και η μέγιστη ίση με 640.

```
s24.describe()
```

	price
count	359.000000
mean	570.370474
std	25.793536
min	519.000000
25%	550.000000
50%	569.000000
75%	582.000000
max	640.000000

Εικονα 117: περιγραφικά στατιστικά της χρονοσειρας

Επίσης πολύ σημαντική είναι η επόμενη γραμμή κώδικα οπού θέτουμε στην python ότι ο δείκτης της χρονοσειραw είναι σε ημερήσια βάση (frequency=daily) ,καθώς είναι πολύ σημαντικό για την δημιουργία των προβλέψεων και μελλοντικών εκτός του dataset ημερομηνιών που θα τοποθετηθούν οι προβλέψεις.

```
s24.index=pd.date_range(start=s24.index[0],periods=len(s24),freq='D')
```

5.2.1 ΚΙΝΗΤΟΣ ΜΕΣΟΣ ΟΡΟΣ

Ο κινητός μέσος ορός είναι μια απλή τεχνική εξομάλυνσης των δεδομένων που χρησιμοποιείται κατά κύριο λόγο σε χρονοσειρες οπού δεν υπάρχουν έντονα φαινόμενα εποχικότητας , καθώς ο βασικός σκοπός του είναι να ανάδειξη της γενικότερης τάσης της χρονοσειρας και να μειώσει την μεταβλητότητα της. Ο τύπος εφαρμογής του κινητού μέσου ορού είναι ο παρακάτω:

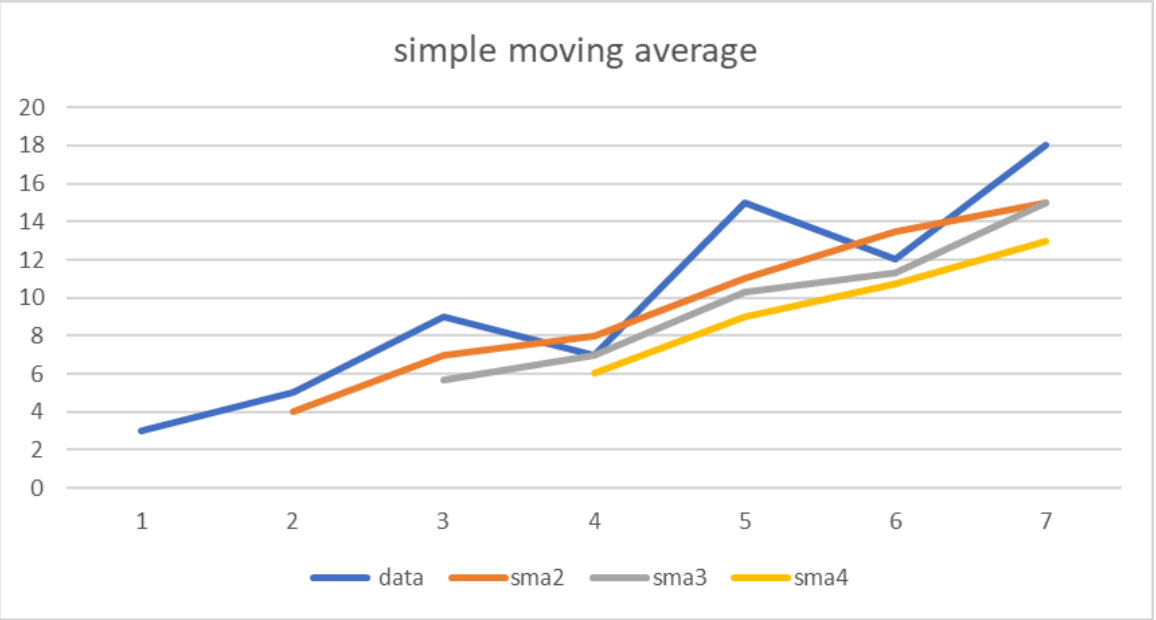
$$MA_t = \frac{1}{k}\sum_{i=0}^{k-1} Y_{t-i}$$

Οπού Υτ είναι η τιμή της χρονοσειρας στον χρόνο τ, κ είναι το μέγεθος του παραθύρου δηλαδή ο αριθμός των παρατηρήσεων από τις οποίες υπολογίζεται ο μέσος ορός ΜΑτ. Όταν το κ είναι ένας μεγάλος αριθμός δηλαδή ο μέσος ορός υπολογίζεται από πολλές τιμές τότε η καμπύλη του κινητού μέσου ορού είναι πιο εξομαλυμένη smooth το θετικό σε αυτή την περίπτωση είναι ότι ο αναλυτής μπορεί να αντιληφθεί καλυτέρα την τάση της χρονοσειρας από την άλλη πλευρά όμως οι προβλέψεις απέχουν πολύ από τις πραγματικές τιμές και στην περίπτωση οπού υπάρξει μια έντονη αλλαγή στην κλίση της χρονοσειρας ο κινητός μέσος ορός με μεγάλο κ δεν μπορεί να αποκριθεί σε αυτήν την αλλαγή γρηγορά . Θα μπορούσε κάνεις να θεωρήσει ότι ένα μεγάλο κ οδηγεί σε underfitting των πραγματικών δεδομένων. Από την άλλη πλευρά ένα μικρό κ για παράδειγμα κ=2 ανταποκρίνεται πολύ γρηγορά στις αλλαγές της κλίσης της χρονοσειρας αλλά επηρεάζεται πολύ από την διακύμανση με αποτέλεσμα να μην κάνει πραγματικές προβλέψεις των τιμών και απλά να ακολουθεί τον θόρυβο(διακύμανση) δηλαδή θα μπορούσαμε να

θεωρήσουμε ότι επιλέγοντας μικρό κ το μοντέλο κάνει overfitting στα πραγματικά δεδομένα.

Ένα απλό παράδειγμα υπολογισμού του κινητού μέσου ορού παρουσιάζεται παρακάτω.

data	3	5	9	7	15	12	18
sma2		4	7	8	11	13,5	15
sma3			5,666667	7	10,33333	11,33333	15
sma4				6	9	10,75	13



Εικονα 117: απλό παράδειγμα sma

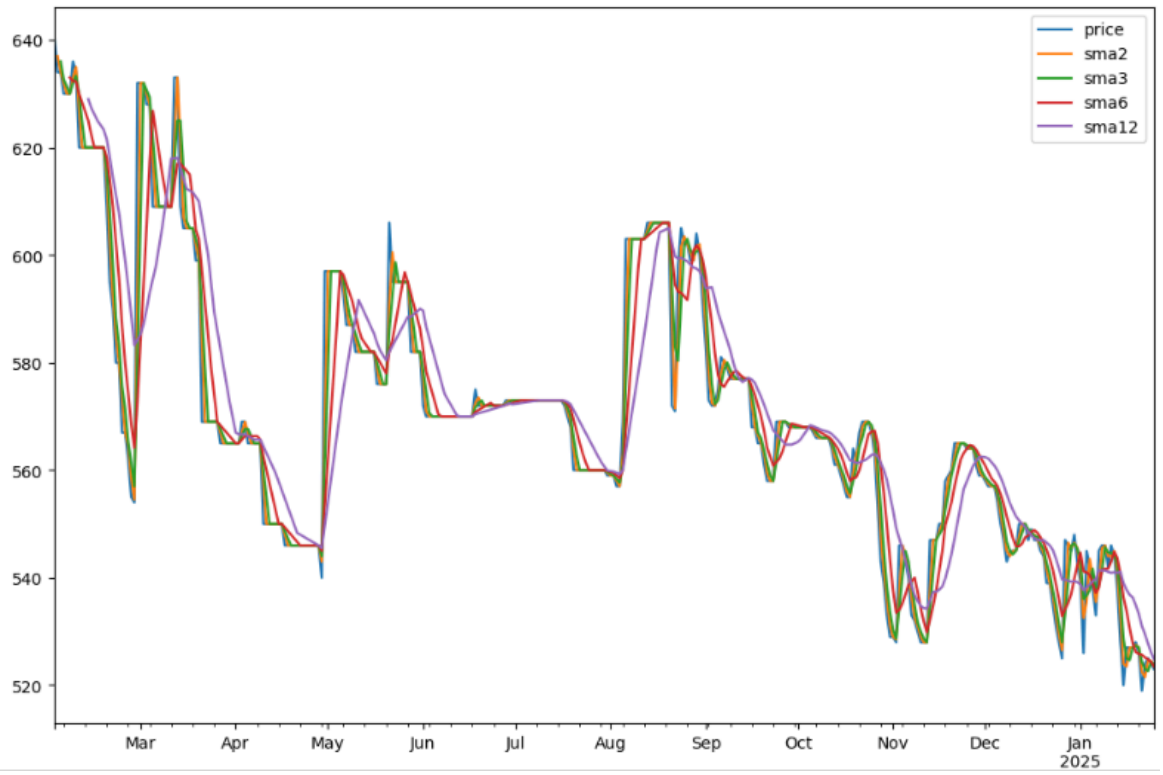
Για την εκτέλεση του μοντέλου του κινητού μέσου ορού στην python στην χρονοσειρα με τις τιμές του κινητού τηλεφώνου χρησιμοποιείται ο παρακάτω κώδικας (μέσα στο rolling επιλέγεται ο αριθμός κ δηλαδή το μέγεθος του window)

```
df['sma2']=df['price'].rolling(2).mean()
df['sma3']=df['price'].rolling(3).mean()
df['sma6']=df['price'].rolling(6).mean()
df['sma12']=df['price'].rolling(12).mean()
df[['price','sma2','sma3','sma6','sma12']].plot(figsize=(12,8))
plt.show()
```

παρατηρούμε ότι καθώς το κ αυξάνεται τόσο πιο ομαλή γίνεται η καμπύλη του κινητού μέσου ορού που προκύπτει , ταυτόχρονα όσο πιο μικρό είναι το κ τόσο πιο μικρό είναι το mse που προκύπτει μεταξύ των προβλέψεων και των πραγματικών τιμών.

```
from sklearn.metrics import mean_squared_error
mse2=mean_squared_error(df['price'].iloc[2:],df['sma2'].iloc[2:])
mse3=mean_squared_error(df['price'].iloc[3:],df['sma3'].iloc[3:])
mse6=mean_squared_error(df['price'].iloc[6:],df['sma6'].iloc[6:])
mse12=mean_squared_error(df['price'].iloc[12:],df['sma12'].iloc[12:])
print('mse2:',mse2)
print('mse3:',mse3)
print('mse6:',mse6)
print('mse12:',mse12)
```

```
mse2: 15.087535014005603
mse3: 34.319600499375746
mse6: 88.926581680831
mse12: 169.50016010246551
```



Εικονα 118: sma στην python για την τιμή του s24

Επιπλεον θα πρεπει να αναφερουμε ότι αυτό το μοντελο δεν μπορει να κανει προβλεψεις για την τιμη του κινητου τηλεφωνου στο μελλον, επισης παρατιρουμε οτι το μοντελο δεν μπορει να κανει προβλεψεις για τις πρωτες τιμες της χρονοσειρας αναλογα τον αριθμο των παρατιρισεων που εχουν επιλεχθει για τον υπολογισμο του μεσου ορου(κ).

5.2.2 WEIGHTED MOVING AVERAGE

Ο σταθμικός μέσος ορός είναι μια παρόμοια μέθοδος ανάλυσης χρονολογικών σειρών με τον κινήτο μέσο ορό αλλά σε αντίθεση με αυτόν η κάθε τιμή της χρονοσειρας είναι πολλαπλασιασμένη με ένα βάρος τα βαρύ ανάλογα με τις τιμές οπού θέλουμε να υπολογίζεται ο μέσος ορός θα πρέπει να αθροίζουν με το 1. Ο τύπος υπολογισμού του είναι ο παρακάτω.

wmat= $w1yt + w2yt-1+. . . +wkyt-k+1$

οπού wmat είναι ο σταθμισμένος μέσος ορός της περιόδου τ

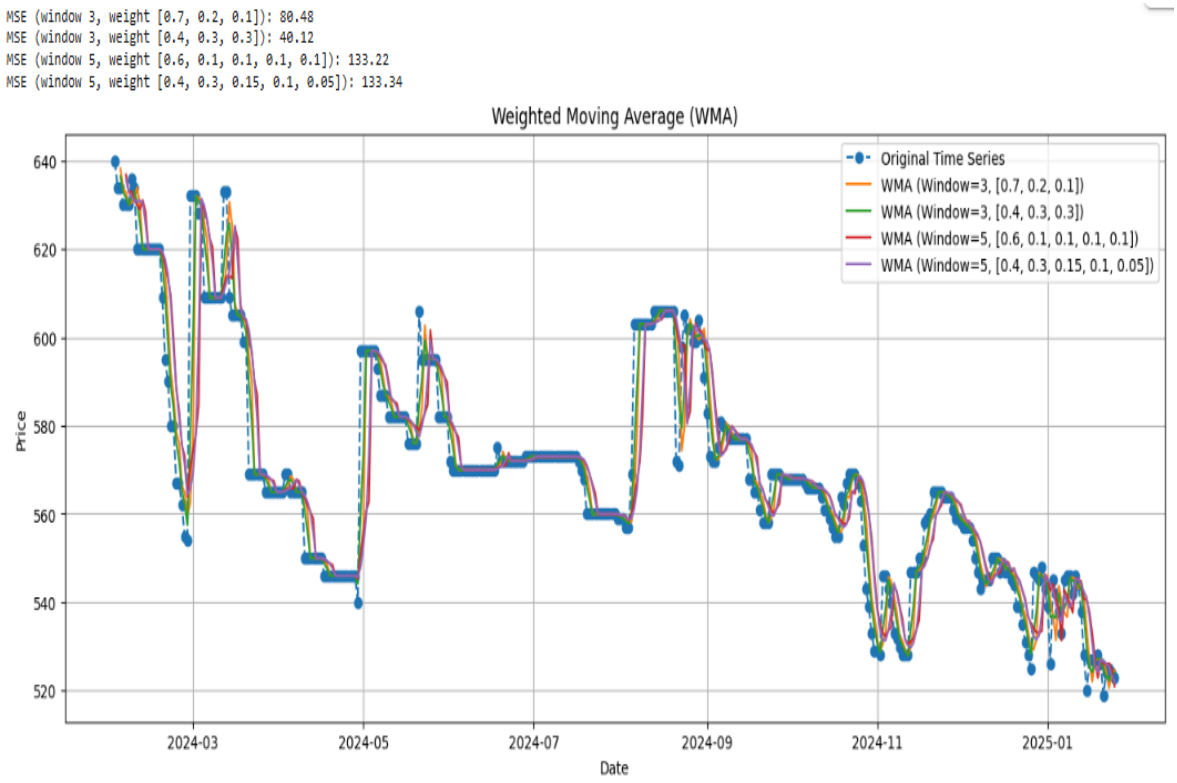
τα w είναι τα βάρη που πολλαπλασιάζονται με την εκάστοτε παρατήρηση yt

και το κ είναι ο αριθμός των παρατηρήσεων που χρησιμοποιούνται για τον υπολογισμό του.

Το πλεονέκτημα του ενάτη του κινητού μέσου ορού είναι ότι στην περίπτωση οπού έχει τεθεί ένα μεγάλο βάρος στην πιο πρόσφατη παρατήρηση ο μέσος ορός πέφτει πολύ κοντά στα πραγματικά δεδομένα δηλαδή το μοντέλο ανιχνεύει γρηγορότερα τις αλλαγές στην χρονοσειρα προφανώς αυτό έχει ως αποτέλεσμα μικρό mse αλλά παρατηρείται έντονα το φαινόμενο του overfitting. Αντίθετος αν τοποθετηθεί ένα μικρό βάρος στις πιο πρόσφατες τιμές της χρονοσειρας τότε η καμπύλη των προβλέψεων είναι πιο smooth δηλαδή εστιάζει κύριος στην τάση της χρονοσειρας και όχι στις έντονες αλλαγές της. Παρακάτω ακολουθεί ο κώδικας εκτέλεσης του στην python . Συγκεκριμένα έχουν υπολογιστεί τέσσερείς κινητοί μεσοί οροί ο πρώτος είναι με k ίσο με 3 παρατηρήσεις και τα βάρη είναι 0,7 0,2 και 0,1 στην πρώτη εκδοχή της ενώ στην δεύτερη τα βάρη είναι 0,4 0,3 και 0,3 ενώ για κ ίσο με 5 έχει επιλεγεί στην πρώτη εκδοχή του τα βάρη 0,6 0,1 0,1 0,1 και 0,1 ενώ στην δεύτερη εκδοχή του τα βάρη είναι πιο μοιρασμένα και στις προηγούμενες τιμές της χρονοσειρας

και είναι ίσα με 0,4 0,3 0,15 0,1 και 0,1 προφανώς η μεγάλη τιμή των βαρών τοποθετείται στην πιο πρόσφατη τιμή.

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.metrics import mean_squared_error
def weighted_moving_average(series, weights):
    window = len(weights)
    wma = np.convolve(series, weights[::-1], mode='valid')
    return pd.Series(wma, index=series.index[window - 1:])
weights_3_1 = np.array([0.7, 0.2, 0.1])
weights_3_2 = np.array([0.4, 0.3, 0.3])
weights_5_1 = np.array([0.6, 0.1, 0.1, 0.1, 0.1])
weights_5_2 = np.array([0.4, 0.3, 0.15, 0.1, 0.05])
wma_3_1 = weighted_moving_average(df['price'], weights_3_1)
wma_3_2 = weighted_moving_average(df['price'], weights_3_2)
wma_5_1 = weighted_moving_average(df['price'], weights_5_1)
wma_5_2 = weighted_moving_average(df['price'], weights_5_2)
mse_3_1 = mean_squared_error(df['price'][wma_3_1.index], wma_3_1)
mse_3_2 = mean_squared_error(df['price'][wma_3_2.index], wma_3_2)
mse_5_1 = mean_squared_error(df['price'][wma_5_1.index], wma_5_1)
mse_5_2 = mean_squared_error(df['price'][wma_5_2.index], wma_5_2)
print(f"MSE (window 3, weight [0.7, 0.2, 0.1]): {mse_3_1:.2f}")
print(f"MSE (window 3, weight [0.4, 0.3, 0.3]): {mse_3_2:.2f}")
print(f"MSE (window 5, weight [0.6, 0.1, 0.1, 0.1, 0.1]): {mse_5_1:.2f}")
print(f"MSE (window 5, weight [0.4, 0.3, 0.15, 0.1, 0.05]): {mse_5_2:.2f}")
plt.figure(figsize=(16, 6))
plt.plot(df.index, df['price'], label='Original Time Series', marker='o', linestyle='dashed')
plt.plot(wma_3_1.index, wma_3_1, label='WMA (Window=3, [0.7, 0.2, 0.1])', linestyle='solid')
plt.plot(wma_3_2.index, wma_3_2, label='WMA (Window=3, [0.4, 0.3, 0.3])', linestyle='solid')
plt.plot(wma_5_1.index, wma_5_1, label='WMA (Window=5, [0.6, 0.1, 0.1, 0.1, 0.1])', linestyle='solid')
plt.plot(wma_5_2.index, wma_5_2, label='WMA (Window=5, [0.4, 0.3, 0.15, 0.1, 0.05])', linestyle='solid')
plt.xlabel('Date')
plt.ylabel('Price')
plt.title('Weighted Moving Average (WMA)')
plt.legend()
plt.grid()
plt.show()
```



Εικονα 119: weighted moving average στην python για την τιμή του s24

Στο παραπάνω διάγραμμα παρατηρείται αυτό που αναφέρθηκε στην αρχή όσο πιο μεγάλο είναι το βάρος που τοποθετείται στην πιο πρόσφατη τιμή τόσο το mse γίνεται μικρότερο, επίσης παρατηρούμε ότι αυξάνοντας τον αριθμό κ επειδή ο μέσος ορός υπολογίζεται από περισσότερες πιο παλιές τιμές δημιουργείται lag (καθυστέρηση) στην χρονοσειρα. Αρά το συμπέρασμα είναι ότι εάν η χρονοσειρα αλλάζει(μεταβάλλεται) συνεχώς όπως πχ η τιμή μιας μετοχής καλό είναι να χρησιμοποιηθεί μικρό κ και μεγάλο βάρος αν η χρονοσειρα είναι σταθερή με μικρή διακύμανση το αντίστροφο. Τέλος και αυτή η μέθοδος δεν μπορεί να κάνει προβλέψεις για μελλοντικές τιμές.

5.2.3 EXPONENTIAL SMOOTHING

Η μέθοδος της εκθετικής εξομάλισης είναι παρόμοια με εκείνη της μεθόδου του σταθμισμένου μέσου ορού με την διαφορά ότι τα βάρη ακολουθούν εκθετική κατανομή, δίνοντας με αυτό τον τρόπο περισσότερη βαρύτητα στις πιο πρόσφατες τιμές. Το βάρος σε αυτήν την μέθοδο συμβολίζεται με α και παίρνει τιμές από το 0 έως το 1. Όσο πιο μεγάλο νούμερο επιλεγεί για το α τόσο πιο κοντά θα είναι οι προβλέψεις στην πραγματική τιμή και αυτό όπως έχουμε πει πολλές φορές ήδη οδηγεί σε overfitting καθώς το μοντέλο ακολουθεί και τον θόρυβο των δεδομένων , από την άλλη πλευρά αν επιλεγεί ένα α κοντά στο μηδέν

τότε οι προβλέψεις του μοντέλου δεν ακολουθούν άμεσα τις αλλαγές των πραγματικών δεδομένων. Παρακάτω ακολουθεί ο τύπος αυτού του μοντέλου.

$$expt = a * Yt + (1 - a)\ Yt\text{-}1$$

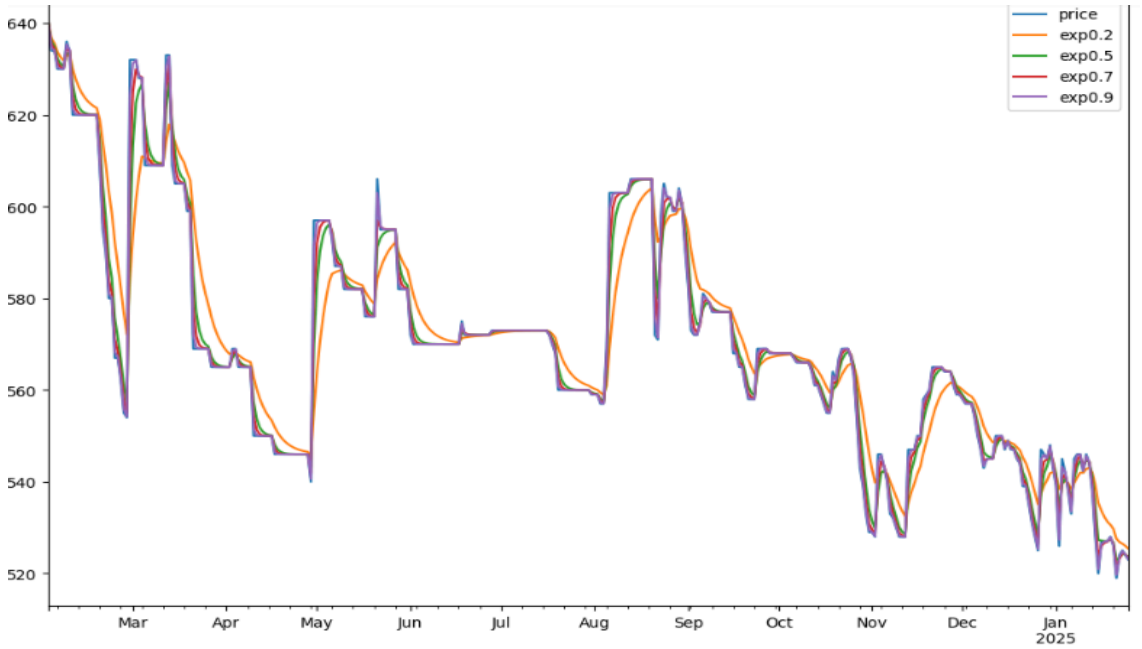
expt είναι η εκθετικά εξομαλυμένη τιμή στον χρόνο τ

Υτ είναι η παρατήρηση στον χρόνο τ (δηλαδή η πιο πρόσφατη τιμή)

Και Υτ-1 είναι η προηγουμένη τιμή από την πιο πρόσφατη , σε αυτό το μοντέλο μπορούν να χρησιμοποιηθούν για τον υπολογισμό του περισσότερες από δυο παρατηρήσεις.

Στην συνέχεια ακολουθεί η εκτέλεση της απλής εκθετικής εξομάλυνσης στην python με διάφορες τιμές του α όπως το 0,2 το 0,5 το 0,7 και το 0,9.

```
df['exp0.2']=df['price'].ewm(alpha=0.2).mean()
df['exp0.5']=df['price'].ewm(alpha=0.5).mean()
df['exp0.7']=df['price'].ewm(alpha=0.7).mean()
df['exp0.9']=df['price'].ewm(alpha=0.9).mean()
df[['price','exp0.2','exp0.5','exp0.7','exp0.9']].plot(figsize=(12,8))
plt.show()
```



Εικονα 120: sma στην python για την τιμή του s24

```
msewma02=mean_squared_error(df['price'].iloc[1:],df['exp0.2'].iloc[1:])
msewma05=mean_squared_error(df['price'].iloc[1:],df['exp0.5'].iloc[1:])
msewma07=mean_squared_error(df['price'].iloc[1:],df['exp0.7'].iloc[1:])
msewma09=mean_squared_error(df['price'].iloc[1:],df['exp0.9'].iloc[1:])
print('expwma02:',msewma02)
print('expwma05:',msewma05)
print('expwma07:',msewma07)
print('expwma09:',msewma09)
```

```
expwma02: 85.94591657572192
expwma05: 19.58543731669871
expwma07: 5.964772258738684
expwma09: 0.6110589582922684
```

Και όπως παρατηρούμε καθώς το α αυξάνεται δηλαδή δίνεται μεγαλύτερο βάρος στην πιο πρόσφατη παρατήρηση το mse μειώνεται και ταυτόχρονα η καμπύλη τείνει να εφάπτεται στα πραγματικά δεδομένα (το mse με α=0,9 είναι σχεδόν ίσο με μηδέν επίσης ο λόγος που χρησιμοποιείται το iloc είναι επειδή η πρώτη εγγραφή των προβλέψεων δεν υπάρχει (είναι nan αρά για να υπολογιστεί το mse πρέπει να αφαιρεθεί από το dataset). Τέλος και αυτό το μοντέλο δεν μπορεί να προβλέψει τιμές εκτός dataset.

5.2.4 HOLT ΚΑΙ HOLT WINTERS

Το πρώτο μοντέλο το οποίο μπορεί να κάνει προβλέψεις για μελλοντικές τιμές πέραν των υπαρχόντων τιμών του dataset είναι το holt μοντέλο. Σε αντίθεση με τη εκθετική εξομάλυνση που έχει μόνο έναν συντελεστή εξομάλυνσης το α η συγκεκριμένη μέθοδος έχει δυο συντελεστές την τιμή α για την εξομάλυνση των παρατηρήσεων της χρονοσειρας (επίπεδο / στάθμη) και την τιμή β για την εξομάλυνση της τάσης της.

Η εξίσωση για τον υπολογισμό του επιπέδου είναι η παρακάτω

$$Lt = \alpha Yt + (1 - a)\,(L_{\tau-1} + T_{\tau-1})$$

Οπού το Lt είναι η στάθμη δηλαδή μια τιμή γύρω από την οποία κυμαίνεται η χρονοσειρα δηλαδή σαν τον μέσο ορό της χρονοσειρας

Το Τt-1 είναι η τάση της χρονοσειρας δηλαδή η κατεύθυνση της αλλά για την προηγουμένη τιμή της χρονοσειρας (η πορεία προς τα πάνω ή προς τα κάτω)

Το α είναι ο συντελεστής εξομάλυνσης της στάθμης και παίρνει τιμές από 0 έως το 1 . ο αναλυτής μπορεί να επιλέξει χειροκίνητα το συντελεστή α αλλά στην python χωρίς να γίνει επιλογή από τον χρήστη επιλέγεται αυτόματα η βέλτιστη τιμή του ,το ίδιο ισχύει και για τον συντελεστή β οπού θα δούμε στην συνέχεια .

Η εξίσωση για την τάση είναι η παρακάτω

$$Tt = \beta\ (\mathrm{L}t - L_{\tau-1}) + (1 - \beta)\ T_{\tau-1}$$

Tt είναι η τάση της χρονοσειρας την χρονική στιγμή τ (την πιο πρόσφατη κατά τον υπολογισμό του μοντέλου) και β είναι ο συντελεστής της τάσης οπού παίρνει τιμές από το 0 έως το 1 όσο ποιο κοντά στο 1 είναι τόσο πιο γρηγορά προσαρμόζεται (ανταποκρίνεται) το μοντέλο στις αλλαγές της τάσης. Και η τελική εξίσωση για τον υπολογισμό του holt μοντέλου είναι η παρακάτω

$$\mathrm{Y}_{\tau+h} = St + h*T\tau$$

το h είναι ο αριθμός των μελλοντικών περιόδων για πρόβλεψη

και το Υτ είναι η τιμή της χρονοσειρας την χρονική περίοδο τ

Έχοντας αναφέρει την θεωρία θα εφαρμόσουμε το παραπάνω στην χρονοσειρα με τις τιμές του τηλεφώνου. Αρχικά εισάγουμε από την βιβλιοθήκη stats.models το exponential smoothing και το holt. Στην συνέχεια χωρίζουμε τα δεδομένα σε train και test set στην συγκεκριμένη χρονοσειρα όλες οι παρατηρήσεις εκτός από τις τελευταίες 30 θα χρησιμοποιηθούν για την εκπαίδευση του μοντέλου ενώ οι τελευταίες 30 για να δοκιμάσουμε την αποδοτικότητα του μοντέλου (test set) σε δεδομένα που δεν έχει <<δει>>. Το initialization_method = estimated επιτρέπει στο μοντέλο να επιλέξει αυτόματα τις αρχικές τιμές για την στάθμη και την τάση.

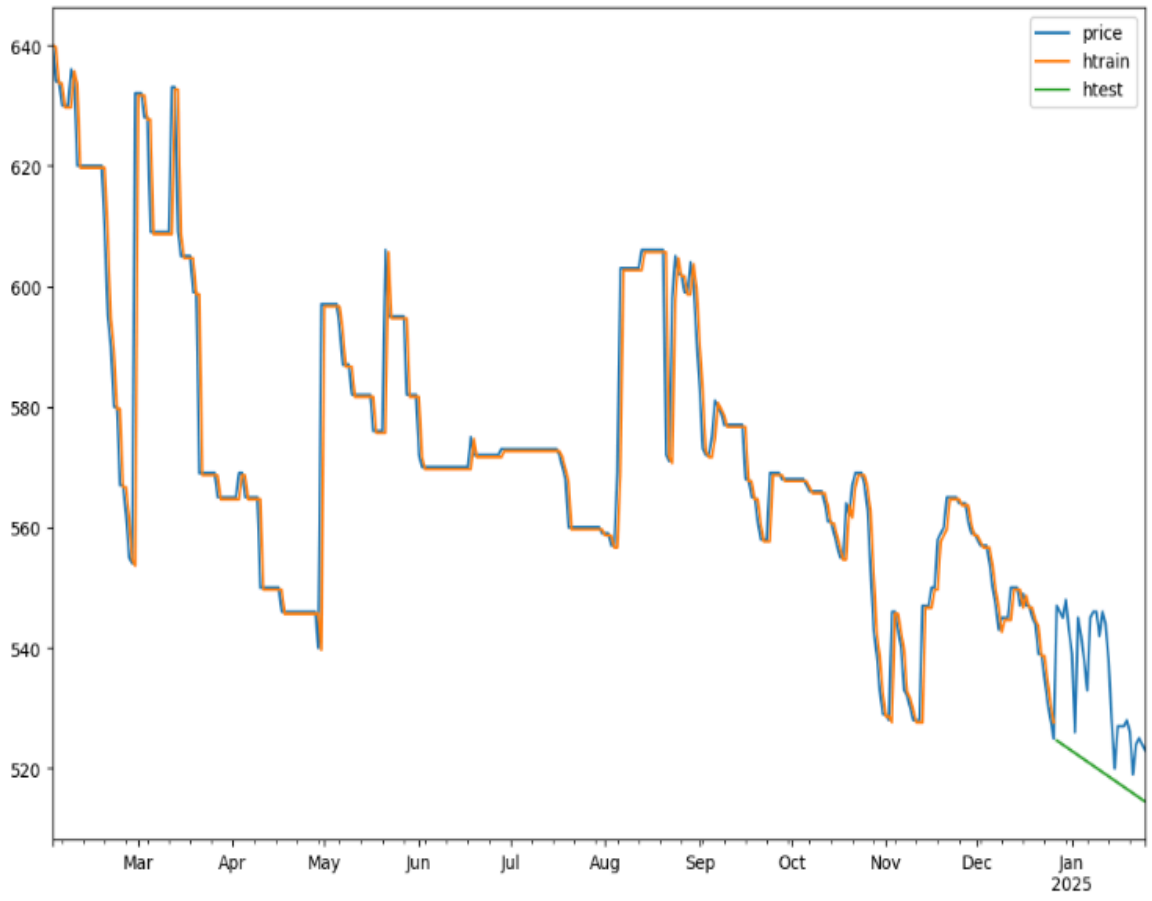
```
from statsmodels.tsa.holtwinters import ExponentialSmoothing
```

```
ntest=30
train=df.iloc[:-ntest]
test=df.iloc[-ntest:]
train_idx=df.index<=train.index[-1]
test_idx=df.index>train.index[-1]
```

```
df=s24.copy()
```

```
from statsmodels.tsa.holtwinters import Holt
h=Holt(train['price'],initialization_method='estimated')
h_fit=h.fit()
df.loc[train_idx,'htrain']=h_fit.predict(start=train.index[0],end=train.index[-1])
df.loc[test_idx,'htest']=h_fit.predict(start=test.index[0],end=test.index[-1])
```

```
df[['price','htrain','htest']].plot(figsize=(12,8))
plt.show()
```



Εικονα 121: εφαρμογή μοντέλου holt για την τιμή του s24

```
msehtrain=mean_squared_error(df['price'].loc[train_idx].iloc[1:],df['htrain'].loc[train_idx].iloc[1:])
msehtest=mean_squared_error(df['price'].loc[test_idx].iloc[1:],df['htest'].loc[test_idx].iloc[1:])
print('msehtrain:',msehtrain)
print('msehtest:',msehtest)
```

```
msehtrain: 60.709436146618806
msehtest: 295.1813437670745
```

Βλέπουμε ότι οι προβλέψεις του μοντέλου όσον αφορά τις γνωστές τιμές (του train set) είναι πολύ κοντά μεταξύ τους με αποτέλεσμα το mse να είναι πολύ μικρό. Αλλά στην συνέχεια επειδή όπως είπαμε και στη θεωρία το μοντέλο αυτό προσαρμόζεται μόνο στην τάση οι προβλέψεις για τις τιμές του test set είναι απλά ακολουθούν μια σταθερή καθοδική πορεία. Στην συνέχεια προβλέπουμε και 30 τιμές εκτός του dataset (στο forecast έχει τοποθετηθεί ο αριθμός 60 καθώς είναι οι πρώτες 30 για το test set και οι επόμενες 30 ως πρόβλεψη μετρά το τέλος των ημερομηνιών).Για να μπορέσουμε αυτές να τις προβάλουμε σε ένα διάγραμμα πρέπει πρώτα να τις τοποθετήσουμε στο dataframe αλλά για να γίνει αυτό θα πρέπει να επεκτείνουμε το dataframe ώστε η στήλη με τον δείκτη να συμπεριλάβει και τις επόμενες ημερομηνίες των προβλέψεων.

```
forecasth=h_fit.forecast(60)
forecasth
```

	0
2024-12-27	524.649454
2024-12-28	524.298909
2024-12-29	523.948363
2024-12-30	523.597817
2024-12-31	523.247271
2025-01-01	522.896725
2025-01-02	522.546180
2025-01-03	522.195634
2025-01-04	521.845088
2025-01-05	521.494542

Εικονα 121 : προβλέψεις μοντέλου holt

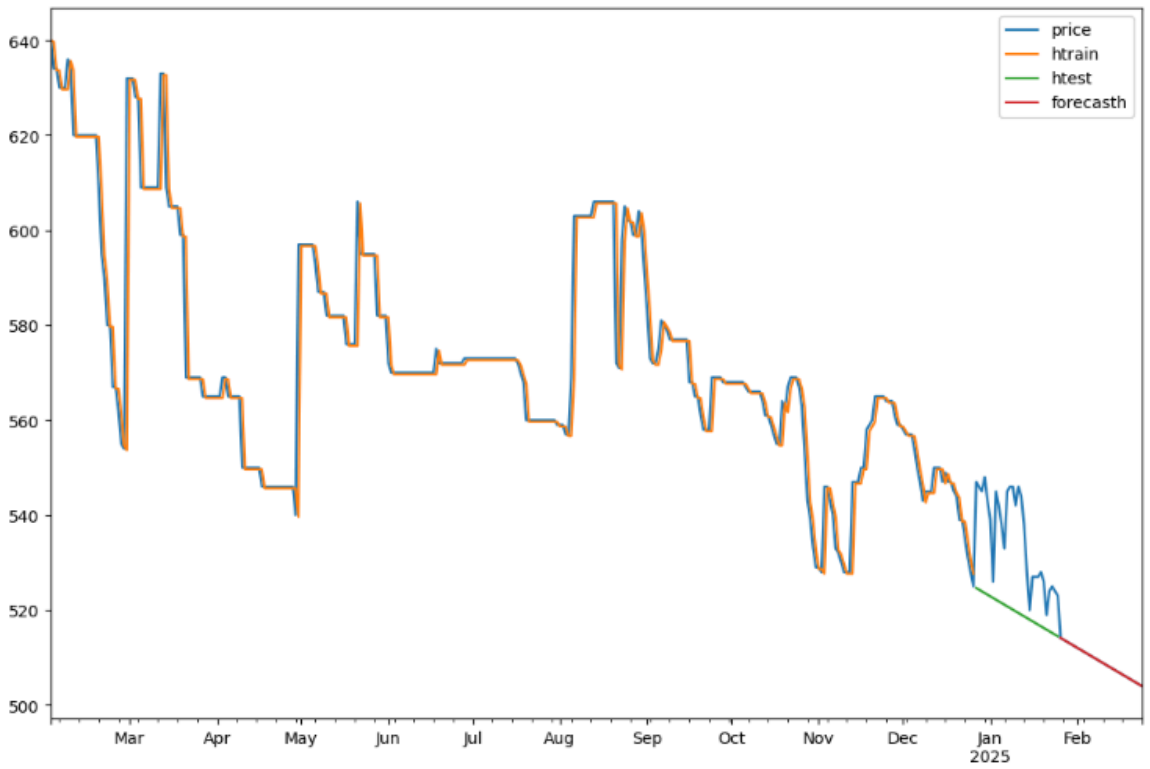
```
forecasth = forecasth.tail(30)
```

```
future_dates = pd.date_range(start=df.index[-1] + pd.Timedelta(days=1), periods=30, freq='D')
future_df = pd.DataFrame(index=future_dates, columns=df.columns)
future_df['price'] = forecasth.values
df = pd.concat([df, future_df])
```

```
<ipython-input-85-6e57797be0de>:4: FutureWarning: The behavior of DataFrame concatenation with emp
  df = pd.concat([df, future_df])
```

```
df['forecasth'] = None
df.loc[forecasth.index, 'forecasth'] = forecasth.values
```

```
df[['price','htrain','htest','forecasth']].plot(figsize=(12,8))
plt.show()
```



Όπως είδαμε παραπάνω το μοντέλο holt δημιουργεί προβλέψεις που απλά ακολουθούν της τάση της χρονοσειρας χωρίς να μπορεί να μοντελοποιησει την εποχικότητα της. Αυτό το πρόβλημα λύνεται μέσο του μοντέλου με τρεις συντελεστές εξομάλυνσης το holt winter το οποίο περιλαμβάνει το συντελεστή επιπέδου α το συντελεστή τάσης β και τον συντελεστή εποχικότητας γ. Οι εξισώσεις υπολογισμού του μοντέλου διαφέρουν ανάλογα αν θεωρήσουμε ότι η τάση είναι προσθετική δηλαδή αυξάνεται η μειώνεται σταθερά στον χρόνο δηλαδή έστω ότι η τιμή του κινητού μειώνεται σταθερά καθημερινά κατά 1 ευρώ τότε η εξίσωση της τάσης που προκύπτει είναι ιδιά με αυτή του μοντέλου holt.

$$Tt = \beta\ (\mathrm{L}t - L_{\tau-1}) + (1 - \beta)\ T_{\tau-1}$$

Η δεύτερη περίπτωση είναι όταν η τάση είναι πολλαπλασιαστική δηλαδή αυξάνεται ή μειώνεται εκθετικά σε αυτή την περίπτωση η εξίσωση υπολογισμού της τάσης είναι η παρακάτω.

$$T_{\tau} = \beta\left(\frac{L_t}{L_t}\right) + (1 - \beta) * T_{\tau-1}$$

Αναφορικά με την εποχικότητα και αυτή έχει δυο εξισώσεις ανάλογα αν θεωρήσει ο αναλυτής ότι αυτή εμφανίζεται προσθετικά ή πολλαπλασιαστικά στην περίπτωση οπού είναι προσθετική δηλαδή η εποχικότητα έχει σταθερό εύρος (οι διακυμάνσεις της είναι σταθερές) παραδείγματος χάρη οι πωλήσεις τα Χριστούγεννα είναι επιπλέον 400 κινητά τηλεφωνά Samsung στο χ κατάστημα τότε αυτή η αύξηση θα είναι σταθερή και ίση με 400 και για όλα τα επόμενα Χριστούγεννα , η εξίσωση αυτού του είδους της εποχικότητας είναι η παρακάτω

St=γ(Yt−Lt) +(1−γ) St−m

Οπού St είναι ο παράγοντας που επηρεάζει την εποχικότητα

Μ είναι η περίοδος της εποχικότητας για παράδειγμα 12 για ετήσια δεδομένα

Και γ ο συντελεστής εξομάλυνσης της εποχικότητας (δηλαδή καθορίζει ποσό γρηγορά προσαρμόζεται το μοντέλο στην εποχικότητα και παίρνει τιμές από το 0 έως το 1).

Στην περίπτωση οπού η εποχικότητα είναι πολλαπλασιαστική δηλαδή στο παράδειγμα με τα Χριστούγεννα οι πωλήσεις κατά την διάρκεια τους για το 2020 αυξήθηκαν σε σχέση με τους προηγουμένους μήνες κατά 100 τεμάχια, το 2021 κατά 200 τεμάχια το 2023 κατά 400 τεμάχια κλπ. τότε η εξίσωση που πρέπει να χρησιμοποιηθεί είναι η παρακάτω

$$s_{\tau} = \gamma\left(\frac{\Upsilon_t}{\Upsilon_t}\right) + (1 - \gamma) * s_{\tau-m}$$

και ο τύπος του μοντέλου holt winter είναι

$$\Upsilon_{\tau+h} = \left(L_t + h * Tt\right) * s_{\tau+h-m}$$

Τώρα αναφορικά με το m δηλαδή την περίοδο οπού εμφανίζεται η εποχικότητα για την τιμή του κινητού τηλεφώνου μπορούν να επιλέγουν διάφορες τιμές καθώς τα δεδομένα είναι ημερήσια όπως για παράδειγμα ο αριθμός 7 δηλαδή ότι η εποχικότητα είναι

εβδομαδιαία ή για παράδειγμα το 14 δηλαδή ανά δυο εβδομάδες ή ακόμα και το 30 για μηνιαία εποχικότητα. Στην python στο παράδειγμα με προσθετική τάση και προσθετική εποχικότητα έχει επιλεγεί m ίσο με 30 ενώ στην δεύτερη εφαρμογή του μοντέλου με προσθετική τάση και πολλαπλασιαστική εποχικότητα το m είναι ίσο με 15.

Για την εφαρμογή του μοντέλου holt winter στην python χρησιμοποιείται η formula exponential smoothing.

```
hw=ExponentialSmoothing(train['price'],trend='add',seasonal='add',seasonal_periods=30,initialization_method='estimated')
hw_fit=hw.fit()
df.loc[train_idx,'hwtrain']=hw_fit.predict(start=train.index[0],end=train.index[-1])
df.loc[test_idx,'hwtest']=hw_fit.predict(start=test.index[0],end=test.index[-1])
```

```
msehwtrain=mean_squared_error(df['price'].loc[train_idx].iloc[1:],df['hwtrain'].loc[train_idx].iloc[1:])
msehwtest=mean_squared_error(df['price'].loc[test_idx].iloc[1:],df['hwtest'].loc[test_idx].iloc[1:])
print('msehwtrain:',msehwtrain)
print('msehwtest:',msehwtest)
```

```
msehwtrain: 55.42816600733153
msehwtest: 511.96106377425315
```

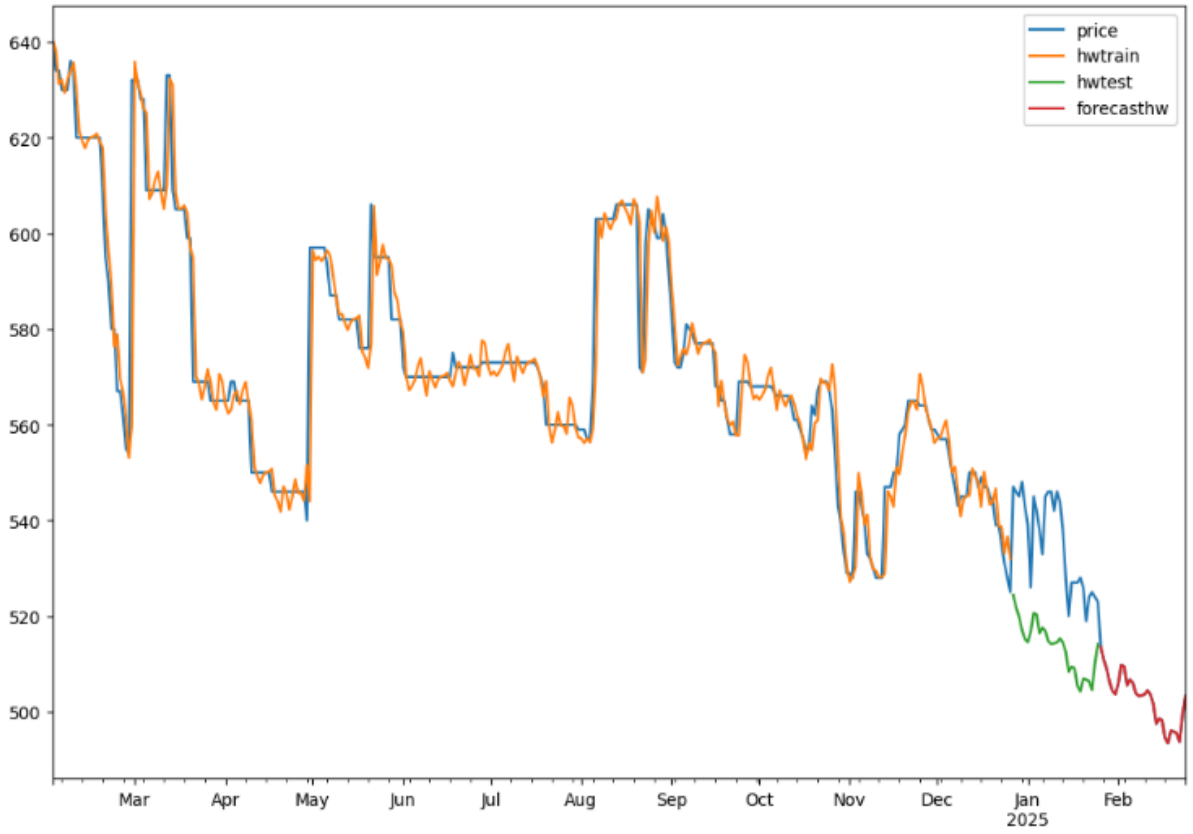
```
forecasthw=hw_fit.forecast(60)
forecasthw
```

Όπως βλέπουμε το mse στα test set είναι διπλάσιο από το αντίστοιχο του απλού holt , στην συνέχεια επεκτείνουμε το column με το index του dataframe ώστε να τοποθετηθούν οι προβλέψεις. Και τέλος δημιουργούμε το διάγραμμα με τις αντίστοιχες προβλέψεις.

```
future_dates = pd.date_range(start=df.index[-1] + pd.Timedelta(days=1), periods=30, freq='D')
future_df = pd.DataFrame(index=future_dates, columns=df.columns)
future_df['price'] = forecasthw.values
df = pd.concat([df, future_df])
```

```
<ipython-input-26-8b871e9935a2>:4: FutureWarning: The behavior of DataFrame concatenation with e
  df = pd.concat([df, future_df])
```

```
df['forecasthw'] = None
df.loc[forecasthw.index, 'forecasthw'] = forecasthw.values
```



Εικονα 122: εφαρμογη Holt winters

Όπως βλέπουμε και σε αυτό το μοντέλο η πρόβλεψη για την μελλοντική τιμή του κινητού είναι φαίνεται να ακολουθεί πτωτική πορεία που είναι αποτέλεσμα του συντελεστή της τάσης ενώ ο συντελεστής της εποχικότητας είναι αυτός που δημιουργεί το ανεβοκατέβασμα(διακύμανση) στις προβλέψεις.

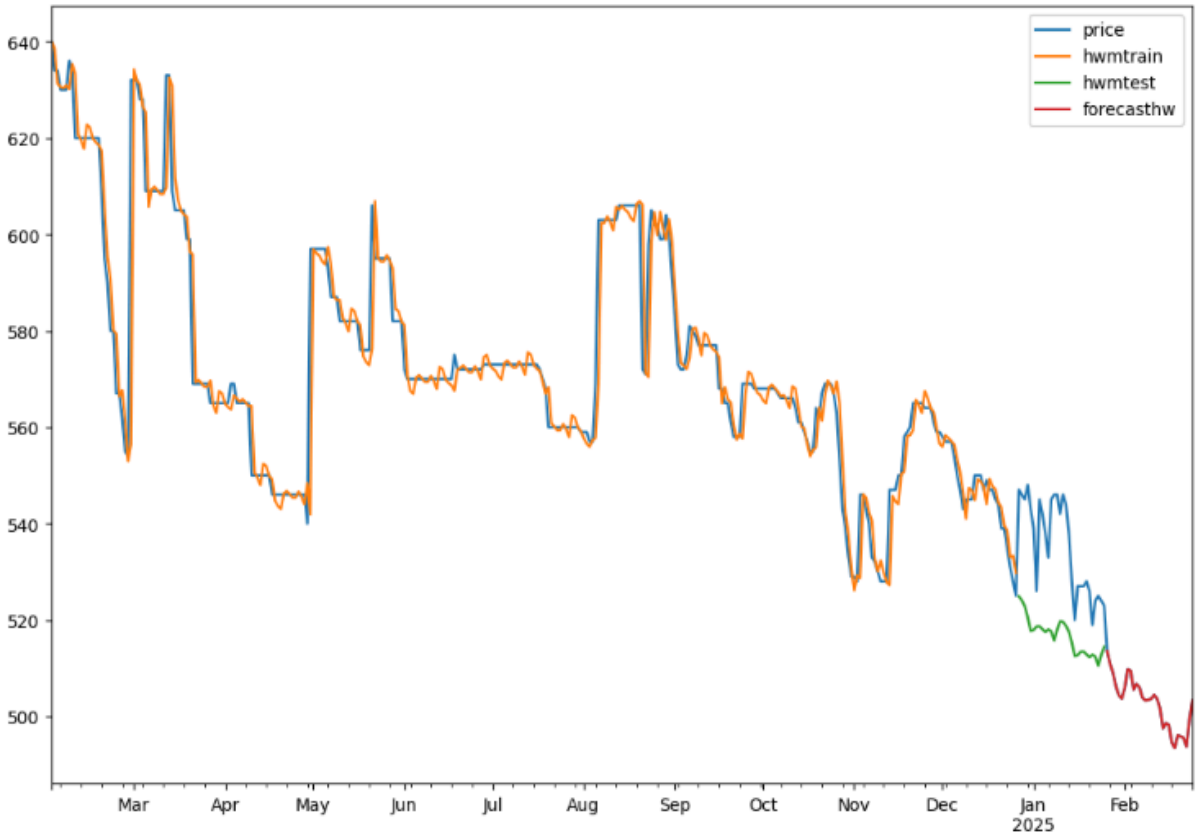
Στην συνέχεια ακολουθεί η εφαρμογή του μοντέλου αλλά με πολλαπλασιαστική εποχικότητα και m=15.

```
hwm=ExponentialSmoothing(train['price'],trend='add',seasonal='mul',seasonal_periods=15,initialization_method='estimated')
hwm_fit=hwm.fit()
df.loc[train_idx,'hwmtrain']=hwm_fit.predict(start=train.index[0],end=train.index[-1])
df.loc[test_idx,'hwmtest']=hwm_fit.predict(start=test.index[0],end=test.index[-1])
```

```
msehwtrain=mean_squared_error(df['price'].loc[train_idx].iloc[1:],df['hwmtrain'].loc[train_idx].iloc[1:])
msehwtest=mean_squared_error(df['price'].loc[test_idx].iloc[1:],df['hwmtest'].loc[test_idx].iloc[1:])
print('msehwtrain:',msehwtrain)
print('msehwtest:',msehwtest)
```

```
msehwtrain: 58.50119175344269
msehwtest: 388.3091862573488
```

Παρατηρούμε ότι το mse στο test set είναι αρκετά μικρότερο από το αντίστοιχο του holt winters με προσθετική εποχικότητα αλλά και πάλι είναι μεγαλύτερο από το mse του απλού holt μοντέλου. Επίσης τα διαγράμματα και οι προβλέψεις των δυο τελευταίων μοντέλων είναι σχεδόν παροιμίες. Το συμπέρασμα στο οποίο οδηγούμαστε μέσο αυτού του μοντέλου τείνει να πλησιάζει αρκετά την πραγματικότητα όσον αφορά τις προβλέψεις για το βραχυχρόνιο μέλλον οπού η τιμή του κινητού μειώνεται όταν κυκλοφορεί το καινούργιο μοντέλο μετα από ένα έτος από την κυκλοφορία του προηγουμένου.



Εικονα 122 μέρος 2: εφαρμογή Holt winters με πολλαπλσιατικη εποχικότητα και μ=15

5.2.5 ΑΥΤΟΠΑΛΙΝΔΡΟΜΑ ΜΟΝΤΕΛΑ AR (AYTOREGRESSIVE MODELS)

Έχοντας αναφερθεί στα απλά μοντέλα ανάλυσης χρονολογικών σειρών συνεχίζουμε με πιο σύνθετα στατιστικά μοντέλα όπως τα αυτοπαλινδρομα μοντέλα. Τα αυτοπαλινδομα μοντέλα όπως αναφέρει και η ονομασία τους χρησιμοποιούν τις προηγούμενες τιμές της χρονοσειρας ως ερμηνευτικές μεταβλητές της εξαρτημένης μεταβλητής η οποία είναι η αρχική χρονοσειρα.

$$AR(P): Yt = \varphi 0 + \varphi 1 * Y_{\tau-1} + \varphi 2 * Y_{\tau-2} + \dots Y_{T-P} + \varepsilon\tau$$

Οπού Yt είναι η εξαρτημένη μεταβλητή (η αρχική χρονοσειρα)

Φ0 είναι η σταθερά ή αλλιώς ο μέσος ορός της χρονοσειρας

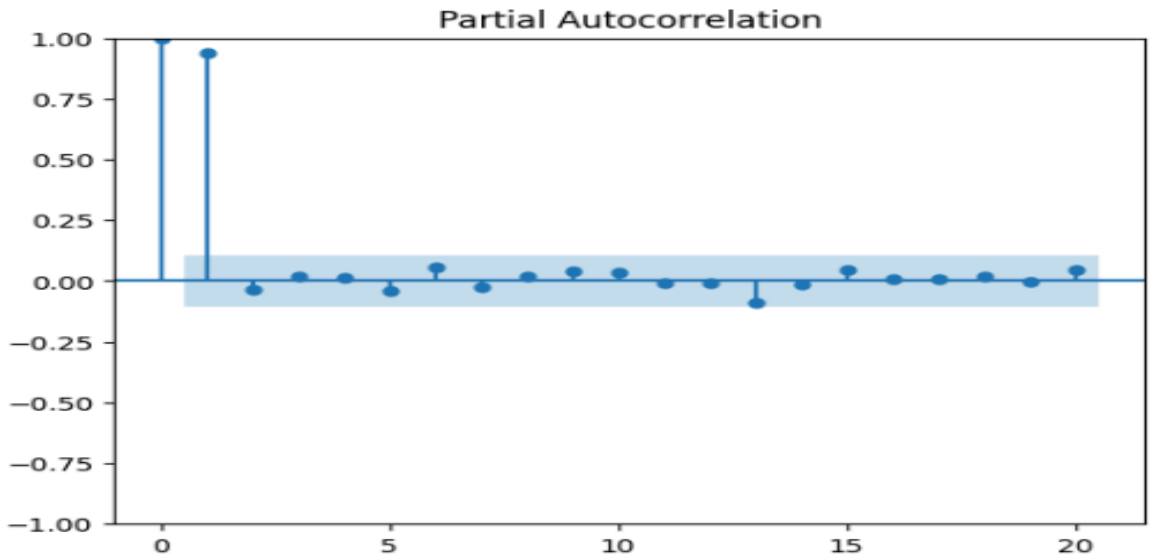
Φ1,φ2,…..φν είναι οι συντελεστές του μοντέλου και

Yτ-1, Yτ-2,….Yτ-p είναι οι προηγούμενες τιμές της χρονοσειρας (οι καθυστερήσεις/lags)

η τιμή p είναι ο αριθμός των καθυστερήσεων και τέλος ετ είναι το σφάλμα (λευκός θόρυβος)

Πριν ξεκινήσουμε την ανάλυση θα πρέπει να αναφέρουμε ότι για την εφαρμογή του παραπάνω μοντέλου απαιτείται η χρονοσειρα να είναι στάσιμη και όπως έχουμε είδη δει με α=0,05 με βάση το dickey fuller test τα δεδομένα είναι στασιμά. Αναφορικά με τον αριθμό p των καθυστερήσεων αυτός μπορεί να βρεθεί είτε αυτόματα με συναιρέσεις στην python όπως το auto Arima αλλά και χειροκίνητα με το pacf collogram (διάγραμμα μερικής αυτοσυσχετησης).

```
plot_pacf(dfar['price'],lags=20)
plt.show()
```



Εικονα 123: διάγραμμα pacf

Όπως παρατηρούμε ο αριθμός των lag που πρέπει να ληφθούν υπόψη είναι ίσος με ένα δηλαδή ότι η τιμή του κινητού τηλεφώνου εξαρτάται κύριος από την τιμή της προηγουμένης ημέρας , και από άλλους εξωγενείς παράγοντες, βέβαια επειδή ο συντελεστής μερικής αυτοσυσχετισης στην 13 καθυστέρηση(13 τιμές πίσω από την πιο πρόσφατη) τείνει να είναι στατιστικά σημαντικός θα θέσουμε p=13. Γενικά είναι καλύτερο το μοντέλο να είναι όσο το δυνατόν πιο απλό δηλαδή με όσο το δυνατόν λιγότερες ερμηνευτικές μεταβλητές.

Ο τρόπος με τον οποίο εκτελείται το μοντέλο στην python είναι μέσο της βιβλιοθήκης statsmodels και πιο συγκεκριμένα μέσο της συνάρτησης Arima. Η συγκεκριμένη συνάρτηση έχει διάφορες παραμέτρους αλλά η πιο σημαντική είναι αυτή του order στην πρώτη θέση μπαίνει τιμή για το p(δηλαδή για το αυτοπαλινδρομο μοντέλο) στην δεύτερη θέση η τιμή για το d δηλαδή για το πόσες φορές θέλουμε να διαφοροποιηθεί η χρονοσειρα (θα το αναλύσουμε παραπάνω στην παραγραφώ για το Arima μοντέλο) και τέλος στην τρίτη θέση μπαίνει η τιμή για το q δηλαδή του συντελεστή του κινητού μέσου ορού(ma). Αρά εφόσον θέλουμε να εφαρμόσουμε το ar(13) επιλέγουμε ως order το (13,0,0) και το trend=’c’ σημαίνει ότι θέλουμε στο μοντέλο να προσδεθεί και ο συντελεστής Φο δηλαδή η σταθερά.

```
ar13=ARIMA(train['price'],order=(13,0,0),trend='c')
ar13_fit=ar13.fit()
print(ar13_fit.summary())
```

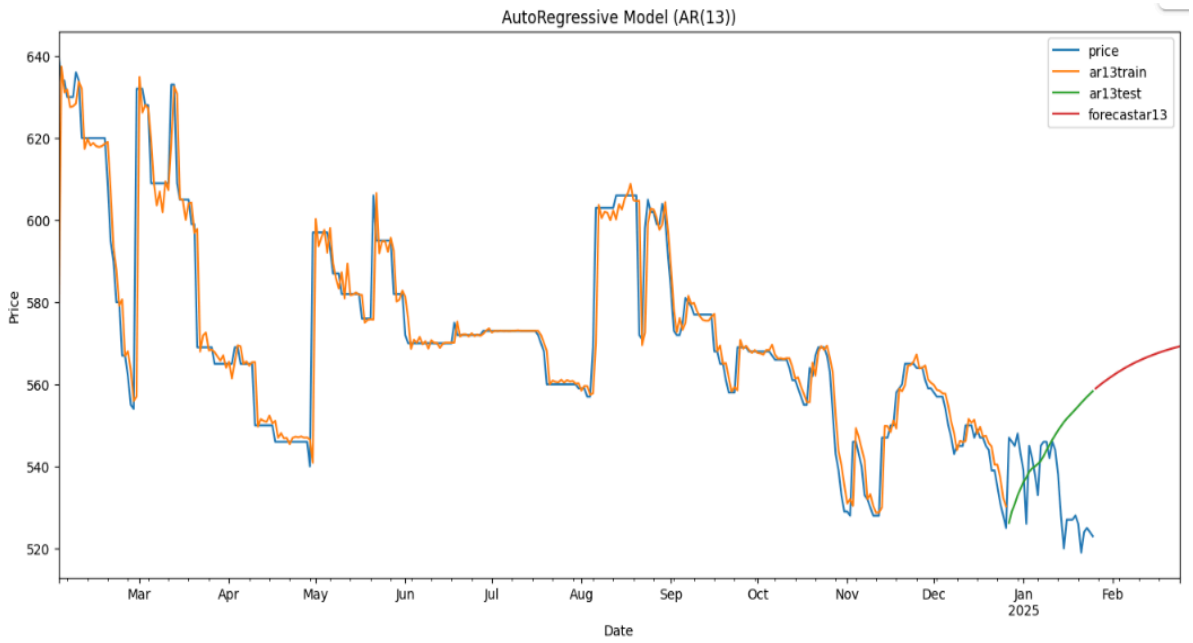
```
                               SARIMAX Results
==============================================================================
Dep. Variable:                  price   No. Observations:                  329
Model:                ARIMA(13, 0, 0)   Log Likelihood               -1135.471
Date:                Thu, 20 Mar 2025   AIC                           2300.942
Time:                        11:32:20   BIC                           2357.883
Sample:                    02-02-2024   HQIC                          2323.657
                         - 12-26-2024
Covariance Type:                  opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const        573.5806     16.155     35.505      0.000     541.917     605.244
ar.L1          1.0307      0.079     13.043      0.000       0.876       1.186
ar.L2         -0.1127      0.097     -1.158      0.247      -0.303       0.078
ar.L3          0.0405      0.174      0.233      0.816      -0.301       0.381
ar.L4          0.0228      0.236      0.097      0.923      -0.439       0.485
ar.L5         -0.0868      0.160     -0.544      0.586      -0.399       0.226
ar.L6          0.0972      0.145      0.673      0.501      -0.186       0.380
ar.L7         -0.0752      0.155     -0.487      0.627      -0.378       0.228
ar.L8          0.0330      0.170      0.194      0.846      -0.301       0.367
ar.L9         -0.0457      0.165     -0.277      0.782      -0.370       0.278
ar.L10         0.0733      0.120      0.613      0.540      -0.161       0.308
ar.L11        -0.0116      0.148     -0.078      0.938      -0.302       0.279
ar.L12         0.1228      0.132      0.929      0.353      -0.136       0.382
ar.L13        -0.1264      0.048     -2.644      0.008      -0.220      -0.033
sigma2        57.7325      2.687     21.484      0.000      52.466      62.999
==============================================================================
```

Εικονα 123 : ar (13)

(Επίσης να αναφέρουμε ότι και σε αυτά τα μοντέλα το train και το test set έχουν διαχωριστεί με τον ίδιο τρόπο όπως και στο holt και στο holt winter μοντέλο.)

Όπως μπορούμε να παρατηρήσουμε με βάση τον πίνακα summary του μοντέλου μόνο η σταθερά ,η πρώτη και δέκατη τρίτη καθυστέρηση είναι στατιστικά σημαντικές ερμηνευτικές μεταβλητές. Όσον αφορά το mse για το train σετ αυτό είναι αρκετά μικρό αλλά το ίδιο δεν ισχύει για το test set καθώς όπως θα δούμε και στο διάγραμμα οι προβλέψεις του μοντέλου για το test set αλλά και αυτές για τις μελλοντικές τιμές του κινητού είναι εκθετικά αυξανόμενες , κάτι το οποίο δεν ισχύει με βάση την συνολική τάση της χρονοσειρας και το γεγονός ότι οι προβλέψεις είναι βραχυπρόθεσμες , καθώς με βάση τους παράγοντες που αναλύσαμε στο πρώτο κεφάλαιο όσο τα καταστήματα που πωλούν το κινήτο λιγοστεύουν και η παραγωγή του σταματάει η τιμή του προϊόντος αυξάνεται.

```
msear13train: 56.94271592362662
msear13test: 407.39785824675585
```



Εικονα 124: εφαρφογη ar(13)

5.2.6 ΜΟΝΤΕΛΑ ΚΙΝΗΤΩΝ ΜΕΣΩΝ ΟΡΩΝ MA (MOVING AVERAGE)

Τα μοντέλα κινητού μέσου ορού που θα αναλύσουμε σε αυτήν την παράγραφο δεν έχουν κάποια σχέση με εκείνα που είδαμε στην ενότητα 7.2.1 καθώς τα πρώτα κάνουν τις εκάστοτε προβλέψεις με βάση τα προηγούμενα σφάλματα δηλαδή τις λάθος προβλέψεις που έκανε το μοντέλο σε προηγούμενο στάδιο. Ο τύπος του μοντέλου παρατίθεται παρακάτω.

$$MA(Q)\, Yt = \mu + \varepsilon t + \theta 1\varepsilon_{\tau-1} + \theta 2\varepsilon_{\tau-2} + \cdots + \theta q\varepsilon_{\tau-q}$$

Οπού το Υτ είναι η τιμή της χρονοσειρας στον χρόνο τ

Το μ είναι η σταθερά ή αλλιώς ο μέσος όρος

Θ1,θ2,…θq είναι οι συντελεστείς του μοντέλο και

Ετ-1 , ετ-2 , ετ-q είναι τα σφάλματά των προηγούμενων περίοδών και τέλος

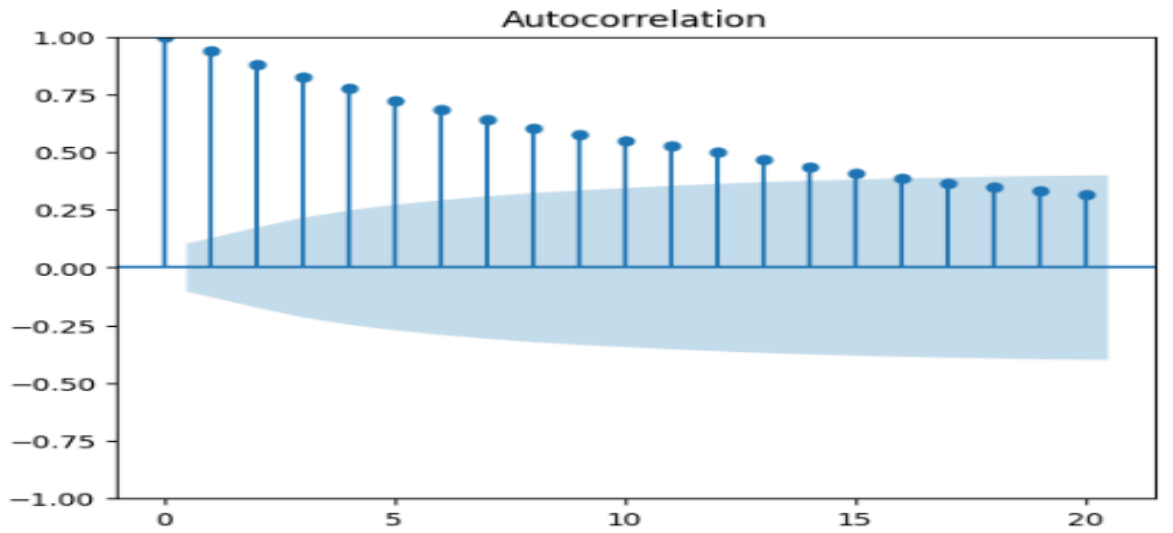
Το q είναι ο αριθμός των καθυστερήσεων των σφαλμάτων ο οποίος μπορεί να βρεθεί για κάθε χρονοσειρα μέσο του διαγράμματος αυτοσυσχετισης acf .

Στην συνέχεια ακολουθεί ένα παράδειγμα για τον υπολογισμό του ετ σε ένα απλό παράδειγμα με q=2.

Χρόνος (t)	Πραγματική Τιμή (Yt)	Πρόβλεψη (ŷt)	Σφάλμα (et)	Σφάλμα et−1	Σφάλμα et−2
1	500	400.00	100.00	0.00	0.00
2	450	400.00	50.00	100.00	0.00
3	400	460.00	-60.00	50.00	100.00
4	420	379.00	41.00	-60.00	50.00
5	380	406.60	-26.60	41.00	-60.00
6	390	396.34	-6.34	-26.60	41.00
7	410	388.22	21.78	-6.34	-26.60
8	370	411.17	-41.17	21.78	-6.34
9	360	381.83	-21.83	-41.17	21.78
10	340	374.55	-34.55	-21.83	-41.17

Αναφορικά με τον συντελεστή q για την χρονοσειρα του κινητού τηλεφώνου αυτός παίρνει την τιμή 15 με βάση το acf διάγραμμα. Και ο τρόπος εφαρμογής του είναι μέσο της συνάρτησης Arima με order(0,0,15)

```
plot_acf(dfar['price'],lags=20)
plt.show()
```



Εικονα 125: acf plot

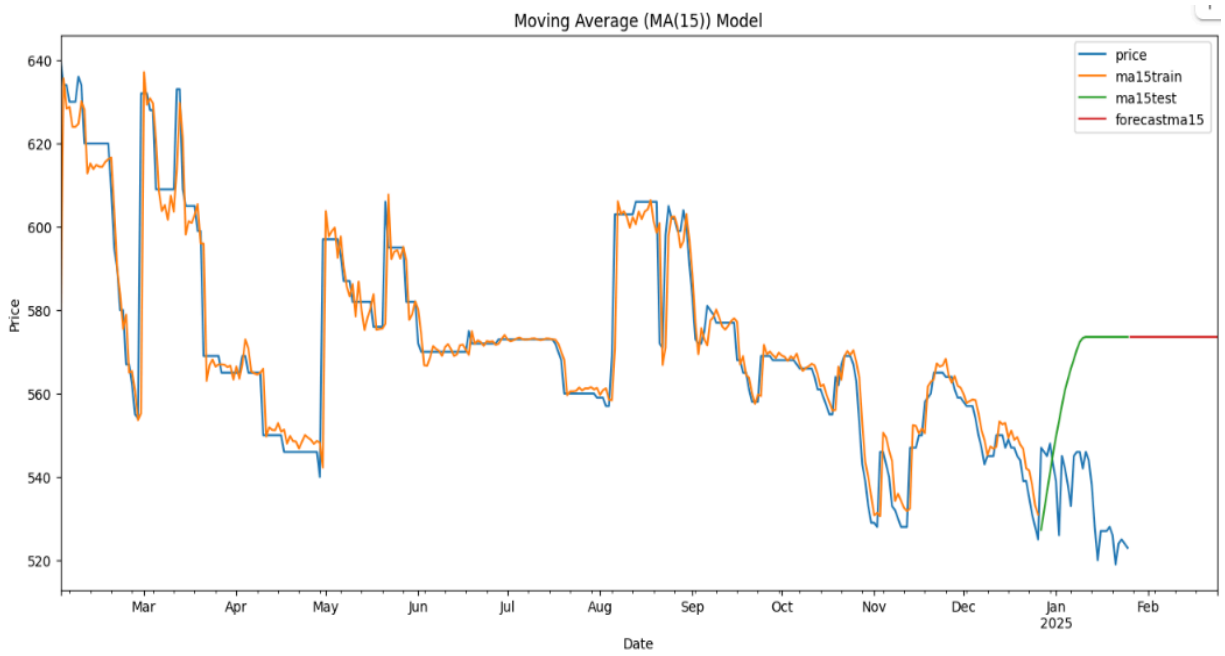
```
ma15=ARIMA(train['price'],order=(0,0,15),trend='c')
ma15_fit=ma15.fit()
print(ma15_fit.summary())
```

```
                               SARIMAX Results
==============================================================================
Dep. Variable:                  price   No. Observations:                  329
Model:                 ARIMA(0, 0, 15)   Log Likelihood               -1141.806
Date:                Thu, 20 Mar 2025   AIC                           2317.612
Time:                        11:32:35   BIC                           2382.145
Sample:                    02-02-2024   HQIC                          2343.356
                         - 12-26-2024
Covariance Type:                  opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const        573.5769      6.463     88.745      0.000     560.909     586.245
ma.L1          1.0676      0.064     16.624      0.000       0.942       1.193
ma.L2          1.0330      0.097     10.626      0.000       0.842       1.224
ma.L3          1.0107      0.143      7.089      0.000       0.731       1.290
ma.L4          0.9942      0.144      6.881      0.000       0.711       1.277
ma.L5          0.8645      0.151      5.712      0.000       0.568       1.161
ma.L6          0.8119      0.160      5.072      0.000       0.498       1.126
ma.L7          0.7254      0.164      4.432      0.000       0.405       1.046
ma.L8          0.6433      0.170      3.780      0.000       0.310       0.977
ma.L9          0.5152      0.173      2.976      0.003       0.176       0.855
ma.L10         0.4491      0.172      2.603      0.009       0.111       0.787
ma.L11         0.3556      0.167      2.133      0.033       0.029       0.682
ma.L12         0.3882      0.166      2.339      0.019       0.063       0.714
ma.L13         0.3211      0.120      2.676      0.007       0.086       0.556
ma.L14         0.1644      0.099      1.666      0.096      -0.029       0.358
ma.L15         0.0417      0.097      0.431      0.667      -0.148       0.231
```

Εικονα 126: ma(15)

Όλες οι ερμηνευτικές μεταβλητές εκτός από την 15 είναι στατιστικά σημαντικές σε επίπεδο σημαντικότητας 90% με βάση το p-value. Παρόλα αυτά οι προβλέψεις του μοντέλο αναφορικά με το test set δεν είναι πολύ καλές καθώς το μοντέλο δεν αντιλαμβάνεται ούτε την τάση ούτε την εποχικότητα των δεδομένων με αποτέλεσμα το mse να είναι μεγαλύτερο από εκείνο του ar μοντέλο, ενώ οι προβλέψεις για το μέλλον είναι απλά μια ευθεία γραμμή στο ύψος του μέσου ορού της χρονοσειρας. Αυτό υποδεικνύει ότι ίσως έπρεπε να είχαν χρησιμοποιηθεί λιγότερα σφάλματά του παρελθόντος δηλαδή το q θα έπρεπε να ήταν μικρότερο, παρόλα αυτά θα συνεχίσουμε την ανάλυση μας με αυτή την επιλογή παρότι το μοντέλο κάνει overfit τα δεδομένα του train set.

```
msema15train: 58.704002459684666
msema15test: 1301.6078178523103
```



Εικονα 126: εφαρφογη ma(15)

5.2.7 ARMA (P, Q) AUTOREGRESSIVE / MOVING AVERAGE MODEL

Το επόμενο μοντέλο είναι το Arma το οποίο είναι συνδυασμός του ar και του ma μοντέλου , δηλαδή σε αυτό το μοντέλο το Υτ εξαρτάται και από τις προηγούμενες τιμές της χρονοσειρα μέσο των ερμηνευτικών μεταβλητών του αυτοπαλινδρομου μοντέλου και ταυτόχρονα από τα σφάλματά των προηγουμένων περίοδών τ μέσο των ερμηνευτικών μεταβλητών του κινητού μέσου ορού. Ο τύπος του είναι ο παρακάτω.

$$ARMA(P,Q)\,Yt = \varphi 1 * \Upsilon_{\tau-1} + \varphi 2 * \Upsilon_{\tau-2} + \ldots + \Upsilon_{T-P} + \theta 1 * \varepsilon_{\tau-1} + \theta 2 * \varepsilon_{\tau-2} + \cdots + \theta q$$

Το Υτ είναι η τιμή της χρονοσειρας στον χρόνο τ

Φ1, φ2…φν είναι οι συντελεστείς συσχέτισης του αυτοπαλινδρομου μοντέλου

Θ1, θ2…θν είναι οι συντελεστές συσχέτισης των σφαλμάτων του κινητού μέσου

Ετ είναι το σφάλμα και όπως έχουμε είδη αναφέρει το q είναι ο αριθμός των καθυστερήσεων των σφαλμάτων ενώ το p είναι ο αριθμός των καθυστερήσεων των προηγουμένων τιμών της χρονοσειρας.

Ο τρόπος εκτέλεσης του στην python είναι μέσο του Arima αυτή την φορά όμως επιλέγεται στο order το (13 το p του ar μοντέλου ,0, 15 το q του ma μοντέλου).

```
arma1315=ARIMA(train['price'],order=(13,0,15),trend='c')
arma1315_fit=arma1315.fit()
print(arma1315_fit.summary())
```

SARIMAX Results

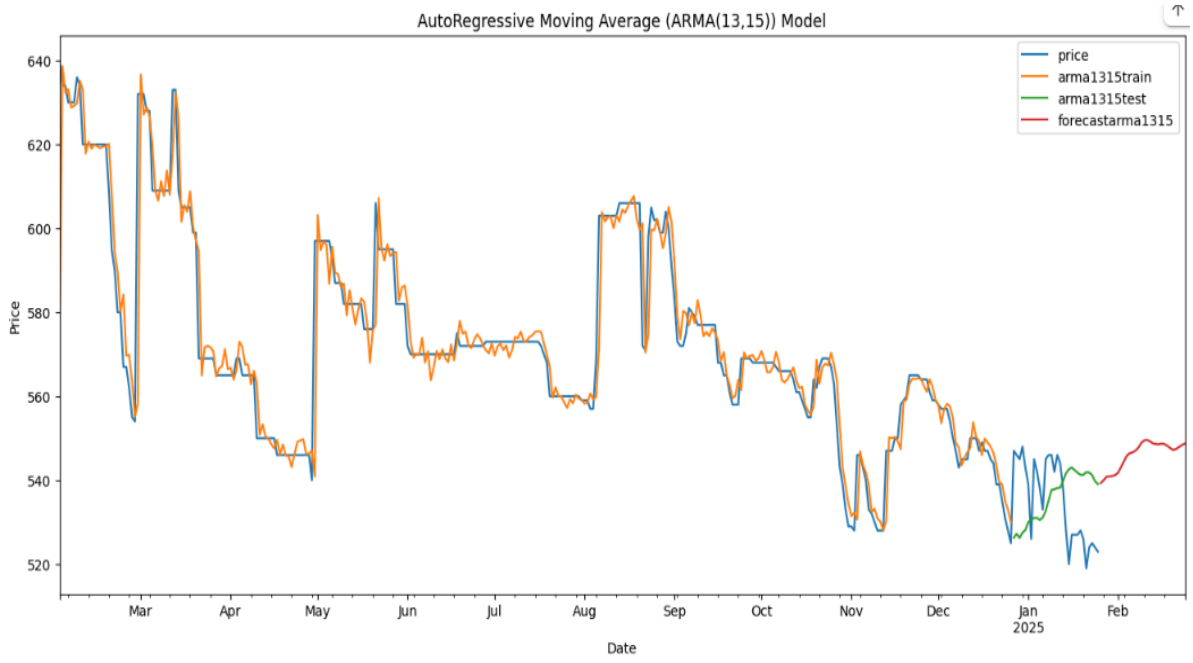
Dep. Variable:	price	No. Observations:	329
Model:	ARIMA(13, 0, 15)	Log Likelihood	-1129.644
Date:	Thu, 20 Mar 2025	AIC	2319.288
Time:	11:33:20	BIC	2433.170
Sample:	02-02-2024 - 12-26-2024	HQIC	2364.719
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
const	573.5847	41.788	13.726	0.000	491.682	655.487
ar.L1	0.7825	0.635	1.233	0.218	-0.462	2.027
ar.L2	0.6842	0.604	1.132	0.258	-0.500	1.869
ar.L3	-0.5316	0.593	-0.897	0.370	-1.693	0.630
ar.L4	-0.2657	0.698	-0.381	0.704	-1.634	1.103
ar.L5	1.0566	0.640	1.651	0.099	-0.198	2.311
ar.L6	-0.4711	0.915	-0.515	0.607	-2.264	1.322
ar.L7	-0.7001	0.656	-1.067	0.286	-1.986	0.586
ar.L8	0.5497	0.708	0.776	0.438	-0.839	1.938
ar.L9	-0.0510	0.627	-0.081	0.935	-1.279	1.177
ar.L10	-0.6326	0.483	-1.309	0.190	-1.579	0.314
ar.L11	0.1298	0.627	0.207	0.836	-1.099	1.358
ar.L12	0.6585	0.407	1.619	0.105	-0.139	1.455
ar.L13	-0.2212	0.469	-0.472	0.637	-1.140	0.698
ma.L1	0.2372	0.622	0.382	0.703	-0.981	1.455
ma.L2	-0.5525	0.436	-1.268	0.205	-1.406	0.301
ma.L3	-0.0032	0.466	-0.007	0.994	-0.917	0.910
ma.L4	0.3471	0.455	0.762	0.446	-0.546	1.240
ma.L5	-0.8436	0.480	-1.759	0.079	-1.784	0.097
ma.L6	-0.3412	0.626	-0.545	0.585	-1.567	0.885
ma.L7	0.5097	0.572	0.891	0.373	-0.612	1.631
ma.L8	-0.0451	0.543	-0.083	0.934	-1.109	1.019
ma.L9	-0.1107	0.470	-0.236	0.814	-1.031	0.810
ma.L10	0.6984	0.383	1.824	0.068	-0.052	1.449
ma.L11	0.5314	0.459	1.158	0.247	-0.368	1.431
ma.L12	-0.1982	0.559	-0.354	0.723	-1.295	0.898
ma.L13	-0.0691	0.244	-0.283	0.777	-0.547	0.409
ma.L14	-0.1503	0.202	-0.745	0.456	-0.546	0.245
ma.L15	-0.1694	0.180	-0.943	0.346	-0.521	0.183
sigma2	54.5631	4.340	12.572	0.000	46.057	63.069

Εικονα 127: arma(13,15)

Όπως μπορούμε να παρατηρήσουμε καμία ερμηνευτική μεταβλητή δεν είναι στατιστικά σημαντική με βάση το p-value. Παρόλα αυτά το μοντέλο αποδίδει το μικρότερο mse στο test set σε σχέση με τα προηγούμενα δυο και αναφορικά με τις προβλέψεις τους δεν είναι τόσο ακραίες αλλά αντίθετος ακολουθούν μια ανοδική τάση και περνούν τιμές από το τα 540 μέχρι τα 560 ευρώ.

```
msearma1315train: 55.918106197635495
msearma1315test: 196.9216978672188
```



Εικονα 128: εφαρφογη arma(13,15)

5.2.8 ARIMA MODEL (P, D, Q) AUTOREGRESSIVE INTEGRATED MOVING AVERAGE ΚΑΙ SARIMA

Η χρήση του μοντέλου Arima ενδείκνυται σε χρονοσειρες οπού δεν είναι στάσιμες καθώς από μόνο του το μοντέλο μετατρέπει τις μη στάσιμες χρονοσειρες σε στάσιμες μέσο της διαφοροποίησης των δεδομένων όπως αναφέρθηκε στις παραπάνω παραγράφους(τελική(Υτ) – αρχική(Υτ-1) παρατήρηση). Αυτό το μοντέλο είναι συνδυασμός των προηγουμένων καθώς περιλαμβάνει τον αριθμό των καθυστερήσεων της χρονοσειρας p από τα ar μοντέλα ,αλλά και τον αριθμό των καθυστερήσεων των σφαλμάτων q από τα ma μοντέλα, και τα συνδυάζει με την διαφοροποίηση των δεδομένων της χρονοσειρας δηλαδή παίρνοντας την d διαφορά τους. Αν χρησιμοποιηθεί d=1 τότε η διαφορά είναι πρώτου βαθμού αν d=2 τότε το μοντέλο χρησιμοποιεί ως δεδομένα την διαφορά της διαφοράς τους. Για την σωστή επιλογή του βαθμού d πρέπει να χρησιμοποιηθεί ταυτόχρονα το dickey fuller test και να επιλεγεί εικονικός ο αριθμός διαφοροποίησης(ολοκλήρωσης) οπού κάνει την χρονοσειρα στάσιμη.

$$ARIMA(P,D,Q)\ Yt = \varphi 1 * \Delta^{d}\Upsilon_{\tau-1} + \varphi 2 * \Delta^{d}\Upsilon_{\tau-2} + \ldots + \Delta^{d}\Upsilon_{T-P} + \theta 1 * \varepsilon_{\tau-1} + \theta 2 * \varepsilon_{\tau-2} +$$

Όπως έχουμε ήδη αναφέρει το φ1,φ2,φν οι συντελεστείς των ερμηνευτικών μεταβλητών του ar, ενώ οι το θ1,θ2,θν είναι οι συντελεστείς του ma τμήματος της συνάρτησης. το Δ^{d}

αναφερεται στην διαφοροποιηση των δεδομενων και το d είναι ο αριθμός των διαφοροποιήσεων. Στην συνέχεια εκτελούμε το μοντέλο με p=13 q=15 και d=1 στην χρονοσειρα των τιμών του κινητού τηλεφώνου μέσο της python.

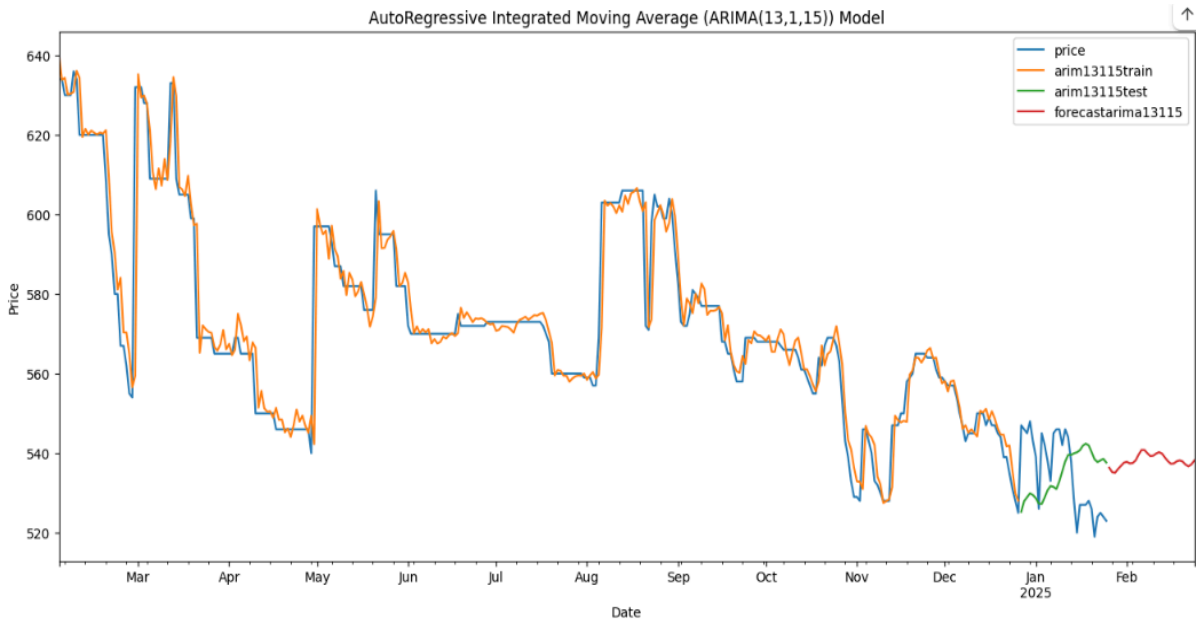
Dep. Variable:	price	No. Observations:	329
Model:	ARIMA(13, 1, 15)	Log Likelihood	-1124.488
Date:	Thu, 20 Mar 2025	AIC	2306.977
Time:	11:34:07	BIC	2416.974
Sample:	02-02-2024 - 12-26-2024	HQIC	2350.862
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
ar.L1	-0.2667	1.020	-0.261	0.794	-2.266	1.733
ar.L2	0.6292	0.767	0.820	0.412	-0.875	2.133
ar.L3	0.1562	0.481	0.325	0.745	-0.787	1.099
ar.L4	-0.3393	0.319	-1.062	0.288	-0.965	0.287
ar.L5	0.6711	0.473	1.419	0.156	-0.256	1.598
ar.L6	0.5471	0.980	0.559	0.576	-1.373	2.467
ar.L7	-0.5603	0.408	-1.372	0.170	-1.360	0.240
ar.L8	-0.2944	0.741	-0.397	0.691	-1.746	1.158
ar.L9	0.1890	0.305	0.620	0.535	-0.408	0.786
ar.L10	-0.4167	0.340	-1.225	0.220	-1.083	0.250
ar.L11	-0.6243	0.456	-1.368	0.171	-1.519	0.270
ar.L12	0.2932	0.457	0.641	0.521	-0.603	1.189
ar.L13	0.4314	0.643	0.671	0.502	-0.829	1.692
ma.L1	0.2566	1.011	0.254	0.800	-1.725	2.239
ma.L2	-0.7053	0.825	-0.855	0.392	-2.322	0.911
ma.L3	-0.2070	0.600	-0.345	0.730	-1.382	0.968
ma.L4	0.3487	0.397	0.879	0.379	-0.429	1.126
ma.L5	-0.7746	0.541	-1.431	0.153	-1.836	0.287
ma.L6	-0.5868	1.118	-0.525	0.600	-2.778	1.604
ma.L7	0.6341	0.558	1.136	0.256	-0.460	1.729
ma.L8	0.3687	0.874	0.422	0.673	-1.344	2.081
ma.L9	-0.2383	0.420	-0.568	0.570	-1.061	0.585
ma.L10	0.5016	0.484	1.036	0.300	-0.448	1.451
ma.L11	0.7038	0.635	1.108	0.268	-0.542	1.949

Εικονα 128: arima(13,1,15)

Όπως βλέπουμε με βάση το p-value δεν υπάρχουν στατικά σημαντικές ερμηνευτικές μεταβλητές. Βέβαια το mse του test set είναι μικρότερο από εκείνο του Arma μοντέλου και οι προβλέψεις για τις μελλοντικές τιμές του κινητού ταιριάζουν αρκετά στις προηγούμενες τιμές της χρονοσειρας καθώς βλέπουμε ότι όταν η τιμή του κινητού πέφτει απότομα σε ένα τοπικό ελάχιστο στην συνέχεια αυξάνεται για το βραχύχρονοί μέλλον. Και αυτό ακριβός έχει προβλέψει αρκετά σωστά το μοντέλο στις τιμές του test set.

```
msearim13115train: 56.185947984787184
msearim13115test: 180.04600733103092
```



Εικονα 129: εφαρφογη arima(13,1,15)

Και τέλος ολοκληρώνουμε την ενότητα των στατιστικών μοντέλων για την ανάλυση των χρονοσειρων με την μέθοδο sarima(p,d,q)*(P,D,Q,m) δηλαδή την seasonal Arima η οποία μοντελοποιει πέραν από την τάση των δεδομένων και την εποχικότητα που υπάρχει σε μια χρονοσειρα. Σε αυτό το μοντέλο υπάρχουν οι μη εποχικοί παράμετροι όπως το p οπού είναι ο αριθμός των αυτοπαλινδρομων ορών το d που είναι ο αριθμός των διαφοροποιήσεων των αρχικών δεδομένων και το q που είναι ο αριθμός των συντελεστών του κινητού μέσου ορού , αλλά και οι αντίστοιχοι εποχικοί παράμετροι όπως το P που είναι ο αριθμός των εποχικών αυτοπαλινδρομων ορών , το D που συμβολίζει των αριθμό των εποχικών διαφοροποιήσεων στην χρονοσειρα , το Q που είναι οι εποχικοί οροί του κινητού μέσου ορού , και τέλος το m που είναι η περίοδος της εποχικότητας πχ 30 για μηνιαία εποχικότητα σε ημερήσια δεδομένα ή 7 για εβδομαδιαία εποχικότητα σε ημερήσια δεδομένα , ή 12 για παράδειγμα για ετήσια εποχικότητα σε μηνιαία δεδομένα κλπ. Η συνάρτηση του εποχικού Arima είναι η παρακάτω.

$$(1-\sum_{i=1}^{p}\phi_i L^i)(1-\sum_{i=1}^{P}\Phi_i L^{im})(1-L)^d(1-L^m)^D Y_t = (1+\sum_{j=1}^{q}\theta_j L^j)(1+\sum_{j=1}^{Q}\Theta_j L^{jm})\epsilon_t$$

Το L είναι ο τελεστής της καθυστέρησης της χρονοσειράς

Φι είναι οι συντελεστείς του μη εποχικού αυτοπαλινδρομου

Με κεφάλαιο φι είναι οι συντελεστείς του εποχικού ar και αντίστοιχα

Θj είναι οι συντελεστές του μη εποχικού κινητού μέσου ορού και με κεφάλαιο θ είναι

Οι εποχικοί συντελεστείς του κινητού μέσου ορού

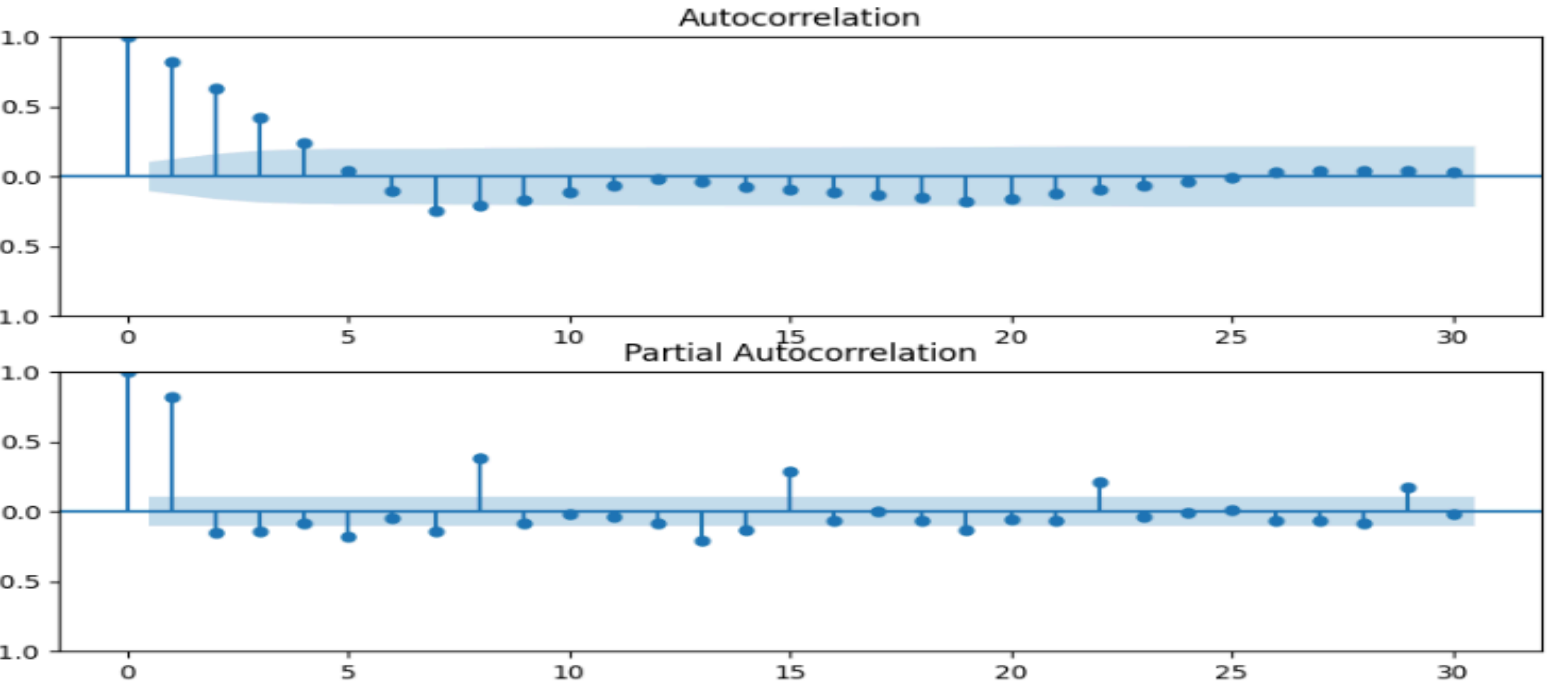
Τα d και D είναι οι διαφοροποιήσεις του μοντέλου και m είναι οι περίοδοι της εποχικότητας.

Στην συνέχεια θα εφαρμόσουμε το μοντέλο sarima στην χρονοσειρα με τις χαμηλότερες τιμές του κινητού τηλεφώνου s24 μέσο της python. Αυτήν την φορά θα κρατήσουμε το μοντέλο πιο απλό δηλαδή θα μειώσουμε τις ερμηνευτικές μεταβλητές(συντελεστές) του κινητού μέσου ορού σε q=6 και του αυτοπαλινδρομου σε p=3 και θα διαφοροποιήσουμε μόνο μια φορά τα αρχικά δεδομένα d=1 καθώς όπως είδαμε στα προηγούμενα μοντέλα καμία ερμηνευτική μεταβλητή δεν ήταν στατιστικά σημαντική. Αναφορικά με τους εποχικούς συντελεστές για να βρεθούν θα πρέπει πρώτα ο αναλυτής να επιλέξει το m δηλαδή την περίοδο της εποχικότητας στο παράδειγμα έχει επιλεγεί ο αριθμός 7 δηλαδή υποθέτουμε εβδομαδιαία εποχικότητα στην συνέχεια θα πρέπει να διαφοροποιήσουμε τα αρχικά δεδομένα 7 φορές όσο και το m. Έχοντας κάνει τα παραπάνω δημιουργούμε το acf και το pacf plot για αυτήν την διαφοροποιημένη επτά φορές χρονοσειρα και με βάση αυτών επιλέγονται οι αντιστοιχεί P και Q συντελεστές εποχικότητας. Στο παράδειγμα 4 συντελεστές αυτοσυσχετισης είναι στατιστικά σημαντικοί αρά το Q είναι ίσο με 4 ενώ αναφορικά με το P παρατηρούμε ότι ανά 7 συντελεστές μερικής αυτοσυσχετισης υπάρχει στατιστική σημαντικότητα μέσα στον τελευταίο μηνά (lags=30) αρά θα θέσουμε ως P=3 (θα μπορούσε να εφαρμοστεί και P=4)

```
df=s24.copy()
```

```
df['seasonal_diff'] = df['price'] - df['price'].shift(7)
```

```
plot_acf(df['seasonal_diff'].dropna(), lags=30, ax=ax[0])
plot_pacf(df['seasonal_diff'].dropna(), lags=30, ax=ax[1])
plt.show()
```




```
from statsmodels.tsa.statespace.sarimax import SARIMAX
sarima13115 = SARIMAX(train['price'], order=(3,1,6), seasonal_order=(3,1,4,7))
sarima13115_fit = sarima13115.fit()
print(sarima13115_fit.summary())
```

Εικονα 129: pacf , acf plot για sarimax

SARIMAX Results

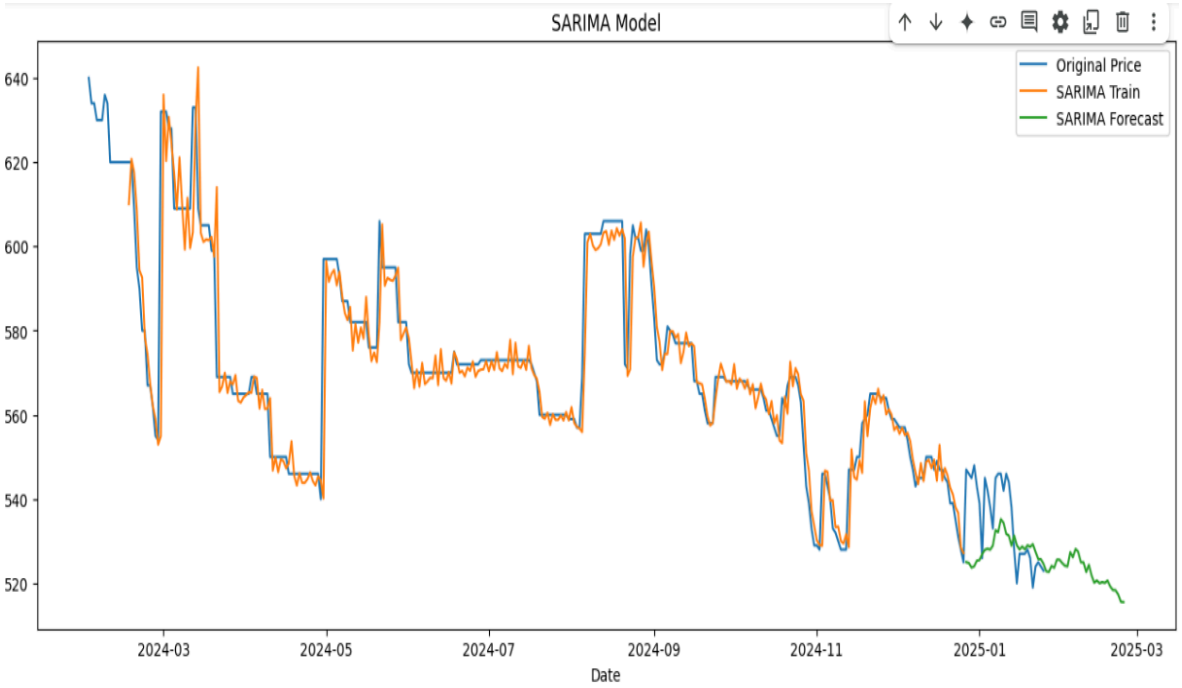
Dep. Variable:	price	No. Observations:	329
Model:	SARIMAX(3, 1, 6)x(3, 1, [1, 2, 3, 4], 7)	Log Likelihood	-1119.093
Date:	Sun, 23 Mar 2025	AIC	2272.185
Time:	16:33:35	BIC	2336.300
Sample:	02-02-2024 - 12-26-2024	HQIC	2297.784
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
ar.L1	-0.1352	45.541	-0.003	0.998	-89.394	89.123
ar.L2	0.7923	1.069	0.741	0.459	-1.303	2.888
ar.L3	0.0856	36.260	0.002	0.998	-70.983	71.154
ma.L1	0.1762	45.533	0.004	0.997	-89.066	89.418
ma.L2	-0.8627	2.919	-0.296	0.768	-6.584	4.858
ma.L3	-0.1815	39.672	-0.005	0.996	-77.937	77.574
ma.L4	0.0370	3.995	0.009	0.993	-7.794	7.868
ma.L5	-0.0554	2.083	-0.027	0.979	-4.138	4.027
ma.L6	-0.0092	2.710	-0.003	0.997	-5.321	5.302
ar.S.L7	-1.1038	0.253	-4.357	0.000	-1.600	-0.607
ar.S.L14	-1.1633	0.169	-6.885	0.000	-1.494	-0.832
ar.S.L21	-0.8545	0.202	-4.223	0.000	-1.251	-0.458
ma.S.L7	0.1056	2.106	0.050	0.960	-4.022	4.233
ma.S.L14	0.0331	2.370	0.014	0.989	-4.611	4.677
ma.S.L21	-0.2322	2.401	-0.097	0.923	-4.938	4.473
ma.S.L28	-0.9004	1.952	-0.461	0.645	-4.726	2.925
sigma2	56.0620	117.529	0.477	0.633	-174.291	286.416

Εικονα 130: sarimax (3,1,6)(3,1,7)

Και όπως παρατηρούμε οι μονές ερμηνευτικές μεταβλητές που είναι στατιστικά σημαντικές είναι οι εποχικοί συντελεστές του ar. Αρά ίσως θα μπορούσε να χρησιμοποιηθεί ένα μοντέλο με ακόμα λιγότερους μη εποχικούς συντελεστές. Παρόλα αυτά το mse στο test σετ είναι το μικρότερο από όλες τις στατιστικές μεθόδους που έχουν χρησιμοποιηθεί και αναφορικά με τις προβλέψεις για την μελλοντική τιμή θα δούμε και στο τέλος του κεφαλαίου ότι είναι πολύ κοντά στις πραγματικές τιμές για τον μηνά Φεβρουάριο του 2025 καθώς το μοντέλο έχει μοντελοποιησει σωστά και την συνολική καθοδική τάση και την εποχικότητα των δεδομένων.

```
msesarimatrain: 382.4359267448722
msesarimatest: 129.82676895994595
```



Εικονα 131: εφαρμογή sarimax (3,1,6)(3,1,7)

Επίσης πριν ολοκληρώσουμε την ενότητα θα πρέπει να αναφέρουμε ότι όλα τα παραπάνω μπορούν να εφαρμοστούν αυτόματα χωρίς ο αναλυτής να πρέπει να ελέγξει την στασιμότητα των δεδομένων του και τα εκάστοτε pacf και acf διαγράμματα για να βρει τους σωστούς συντελεστές του εκάστοτε μοντέλου μέσο της συνάρτησης auto Arima. Βέβαια η παραπάνω συνάρτηση στοχεύει στην ελαχιστοποίηση του aic και του Bic χρησιμοποιώντας όσο το δυνατών λιγότερες ερμηνευτικές μεταβλητές με αποτέλεσμα το μοντέλο μερικές φορές να μην οδηγείται και στις καλύτερες δυνατές προβλέψεις. Επίσης αν ο αναλυτής θέλει να βελτιώσει ακόμα περισσότερο την αποδοτικότητα αυτών των μοντέλων μπορεί να χρησιμοποιήσει και εξωγενές μεταβλητές μέσο του μοντέλου sarimax (το x πηγάζει από την λέξη exogenous) για παράδειγμα μερικές εξωγενείς μεταβλητές για την πρόβλεψη της τιμής του κινητού τηλεφώνου s24 θα μπορούσαν να είναι ο αριθμός των καταστημάτων που πωλούν το κινήτο τηλέφωνο ,το ΑΕΠ της Ελλάδας ,το ποσοστό ανεργίας ,το εισόδημα των καταναλωτών ,ο πληθωρισμός , οι τιμές άλλων ανταγωνιστικών κινητών τηλεφωνων όπως τα iPhone, η τιμή του παλαιοτέρου μοντέλου s23 κινητού τηλεφώνου, τα έξοδα για διαφήμιση του τηλεφώνου στα social media και την τηλεόραση, τα ημερήσια views των trailer του κινητού τηλεφώνου και τα views των βίντεο που προωθούν ή κάνουν review το κινήτο τηλέφωνο στο διαδίκτυο , δασμοί και προβλήματα στην εφοδιαστική αλυσίδα και πολλές άλλες.

5.2.9 LINEAR REGRESSION MACHINE LEARNING METHOD

Στα επόμενα κεφάλαια θα προβλέψουμε την τιμή του κινητού τηλεφώνου με μοντέλα μηχανικής μάθησης , ξεκινώντας με την γραμμική παλινδρόμηση. Έχουμε ήδη αναφέρει στο κεφάλαιο 4 τον τρόπο με τον οποίο λειτουργούν οι παρακάτω μέθοδοι οπότε σε αυτές τις υποενοτητες θα εστιάσουμε κύριος στον τρόπο εφαρμογής των μοντέλων στην python.

Αρχικά δημιουργούμε ένα αντίγραφο του αρχικού dataframe της χρονοσειρας s24 στο dflrt στην συνέχεια χρησιμοποιούμε τον ίδιο τρόπο με τα μοντέλα Arima για να ορίσουμε το train και το test set το οποίο περιλαμβάνει τις τελευταίες 30 τιμές της χρονοσειρας. Στην συνέχεια μετατρέπουμε το dataframe σε NumPy array ώστε να μπορέσουν να εισαχθούν τα δεδομένα στο μοντέλο. Εν συνέχεια θέτουμε ως T την τιμή 90 δηλαδή το μοντέλο χρησιμοποιεί κάθε φορά τις τιμές των προηγουμένων 90 ημέρων για να προβλέψει την τιμή της επομένης ημέρας και όλες αυτές οι τιμές εισέρχονται σε ένα δυσδιάστατο πίνακα(γενικά επιλέγοντας έναν μεγάλο αριθμό ως T κάνει το μοντέλο πιο σταθερό δηλαδή το μοντέλο αναγνωρίζει καλυτέρα την συνολική τάση των δεδομένων βέβαια η εκπαίδευση του μοντέλου απαιτεί περισσότερο χρόνο από την άλλη πλευρά επιλέγοντας ένα μικρό T το μοντέλο γίνεται πιο agile δηλαδή μπορεί και προσαρμόζεται γρηγορότερα σε αλλαγές της χρονοσειρας(μεγάλη μεταβλητότητα των δεδομένων) γενικά ο καλύτερος τροπές επιλογής του T είναι μέσο της διαδικασίας try and error δηλαδή ο ερευνητής πρέπει να δοκιμάσει πολλές διαφορετικές τιμές για το T ώστε να βρεθεί εκείνο το μοντέλο οπού μοντελοποιει αποτελεσματικά τα δεδομένα αποφεύγοντας το overfitting και το underfitting). Στην συνέχεια χωρίζουμε τα δεδομένα σε training και test set και εκπαιδεύουμε με αυτά το μοντέλο της γραμμικής παλινδρόμησης το οποίο χρησιμοποιούμε για να προβλέψουμε τις τιμές του lr_train και lr_test. Έχοντας κάνει τα παραπάνω μπορεί να υπολογιστεί το mse μεταξύ των πραγματικών τιμών και του train set και των πραγματικών τιμών και του test set. Και ολοκληρώνουμε την διαδικασία δημιουργώντας 30 νέες ημερομηνίες μέσα στο dataframe ώστε να τοποθετηθούν σε αυτές οι προβλέψεις του μοντέλου. Έχοντας κάνει όλα τα παραπάνω δημιουργούμε το αντίστοιχο διάγραμμα με τις προβλέψεις για το train , test set και τις προβλέψεις των επομένων 30 ημέρων.

```
from sklearn.linear_model import LinearRegression
from sklearn.svm import SVR
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error
```

```
dflrt=s24.copy()
```

```
dflrt
```

	price
2024-02-02	640
2024-02-03	634
2024-02-04	634
2024-02-05	630
2024-02-06	630

```
ntest=30
train=dflrt.iloc[:-ntest]
test=dflrt.iloc[-ntest:]
```

```
series=dflrt.to_numpy()
T=90
X=[]
Y=[]
for t in range(len(series)-T):
    x=series[t:t+T]
    X.append(x)
    y=series[t+T]
    Y.append(y)
X=np.array(X).reshape(-1,T)
Y=np.array(Y)
N=len(X)
```

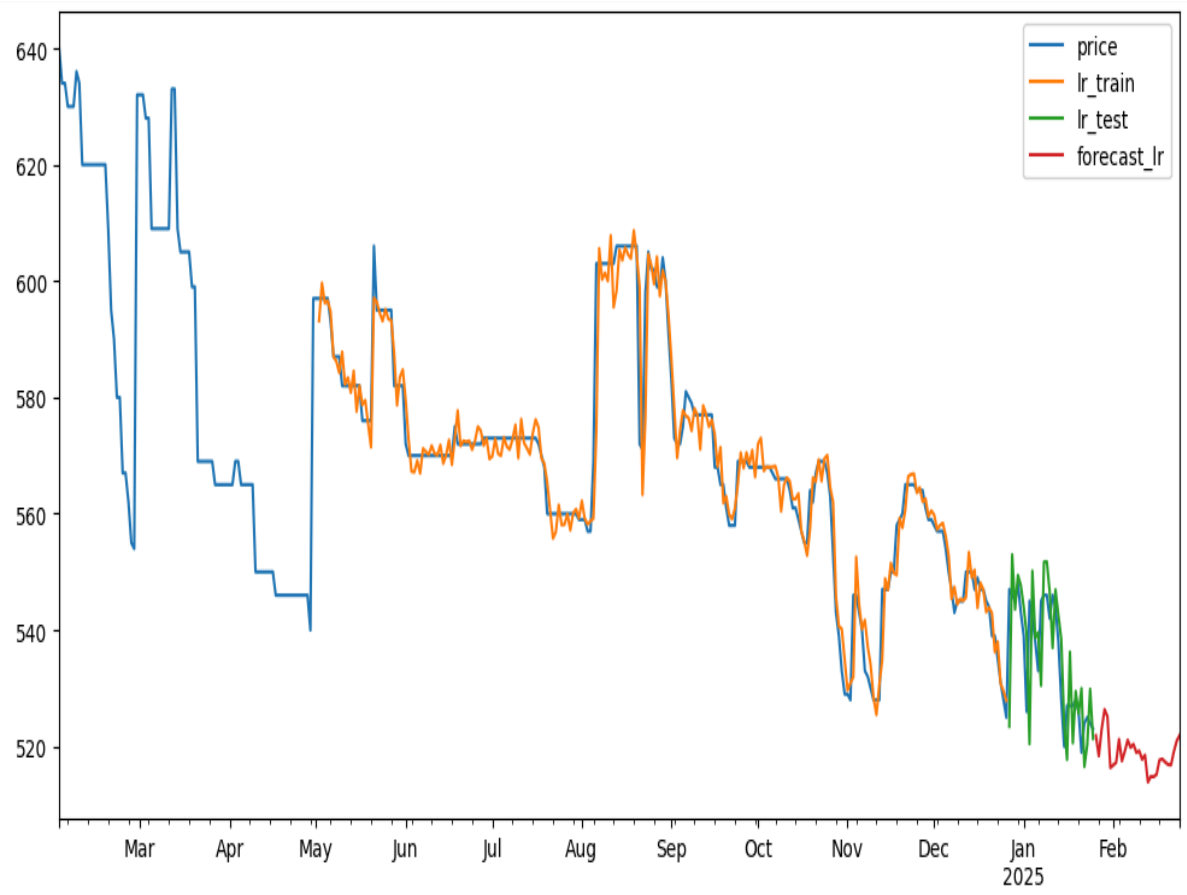
```
X_train=X[:-ntest]
Y_train=Y[:-ntest]
X_test=X[-ntest:]
Y_test=Y[-ntest:]
```

```
lr=LinearRegression()
lr.fit(X_train,Y_train)
train_idx=dflrt.index<=train.index[-1]
test_idx=~train_idx
train_idx[:T]=False
dflrt=pd.DataFrame(dflrt)
dflrt.loc[train_idx,'lr_train']=lr.predict(X_train)
dflrt.loc[test_idx,'lr_test']=lr.predict(X_test)
```

```
mselrtrain=mean_squared_error(dflrt['price'].loc[train_idx].iloc[T:],dflrt['lr_train'].loc[train_idx].iloc[T:])
mselrtest = mean_squared_error(dflrt['price'].loc[test_idx], dflrt['lr_test'].loc[test_idx])
print('mselrtest:',mselrtest)
print('mselrtrain:',mselrtrain)
```

```
mselrtest: 84.17910758570112
mselrtrain: 26.266987584747273
```

```
last_values = dflrt['price'].values[-T:]
last_values = last_values.reshape(1, -1)

future_dates = pd.date_range(start=dflrt.index[-1] + pd.DateOffset(days=1), periods=30, freq='D')
future_df = pd.DataFrame(index=future_dates, columns=['price'])
dflrt = pd.concat([dflrt, future_df])

forecast_lr = []
for i in range(30):
  next_value = lr.predict(last_values)
  forecast_lr.append(next_value[0])
  last_values = np.append(last_values[0,1:], next_value)
  last_values = last_values.reshape(1, -1)

dflrt['forecast_lr'] = None
dflrt.loc[future_dates, 'forecast_lr'] = forecast_lr

dflrt[['price', 'lr_train', 'lr_test', 'forecast_lr']].plot(figsize=(12, 6))
plt.show()
```



```
dflrt.tail(32)
```

	price	lr_train	lr_test	forecast_lr
2025-01-24	524	NaN	529.843311	None
2025-01-25	523	NaN	521.364231	None
2025-01-26	NaN	NaN	NaN	521.846232
2025-01-27	NaN	NaN	NaN	518.377631
2025-01-28	NaN	NaN	NaN	523.044857
2025-01-29	NaN	NaN	NaN	526.354308
2025-01-30	NaN	NaN	NaN	525.127554
2025-01-31	NaN	NaN	NaN	516.362097
2025-02-01	NaN	NaN	NaN	516.795687
2025-02-02	NaN	NaN	NaN	517.240946
2025-02-03	NaN	NaN	NaN	521.241176
2025-02-04	NaN	NaN	NaN	517.509384
2025-02-05	NaN	NaN	NaN	519.159649
2025-02-06	NaN	NaN	NaN	521.097977
2025-02-07	NaN	NaN	NaN	519.826769
2025-02-08	NaN	NaN	NaN	520.391052
2025-02-09	NaN	NaN	NaN	518.953053

Εικονες 131 : κώδικάς και εφαρμογή γραμμικής παλινδρόμησης

Το μοντέλο αποδίδει καταπληκτικές προβλέψεις για τα δεδομένα του test set καθώς επιτυγχάνει ένα mse κοντά στο 26 χωρίς αυτά τα δεδομένα να του έχουν δοθεί κατά την φάση της εκπαίδευσης (για να οδηγηθούμε σε αυτά έγιναν πολλές επαναλήψεις της εκπαίδευσης για να βρεθεί ο κατάλληλος αριθμός T) το ίδιο καλές είναι και οι προβλέψεις για τις 30 επόμενες ημέρες εκτός του dataset καθώς η τιμή κυμαίνεται κοντά στα 520 ευρώ(για τον μηνά Φεβρουάριο).

5.2.10 SUPPORT VECTOR REGRESSION MACINE LEARNING METHOD

Στην συνέχεια θα μοντελοποιησουμε τα δεδομένα με το μοντέλο του support vector regression ο τρόπος εφαρμογής του στην python είναι παρόμοιος με εκείνο της γραμμικής παλινδρόμησης πιο συγκεκριμένα για άλλη μια φορά προετοιμάζουμε τα δεδομένα δηλαδή δημιουργούμε το train και test set μόνο που αυτή την φορά θέτουμε ως T τις τρεις προηγούμενες περιόδους καθώς με την επιλογή παραπάνω περίοδών το μοντέλο κάνει overfit τα δεδομένα με αποτέλεσμα οι προβλέψεις για το test set να είναι είτε εκθετικά αυξανόμενες είτε μια ευθεία γραμμή (με ταυτόχρονη επιλογή των default τιμών δηλαδή χωρίς να επιλεχθεί χειροκίνητα το Kernal το c και το epsilon) . Στην συνέχεια δημιουργούμε τα arrays των ζευγαριών μεταξύ των ερμηνευτικών μεταβλητών(χ) και της

εξαρτημένης μεταβλητής(Y). Ένα παράδειγμά για την κατανόησή του Τ γίνεται στον παρακάτω πίνακα με τυχαίες τιμές.

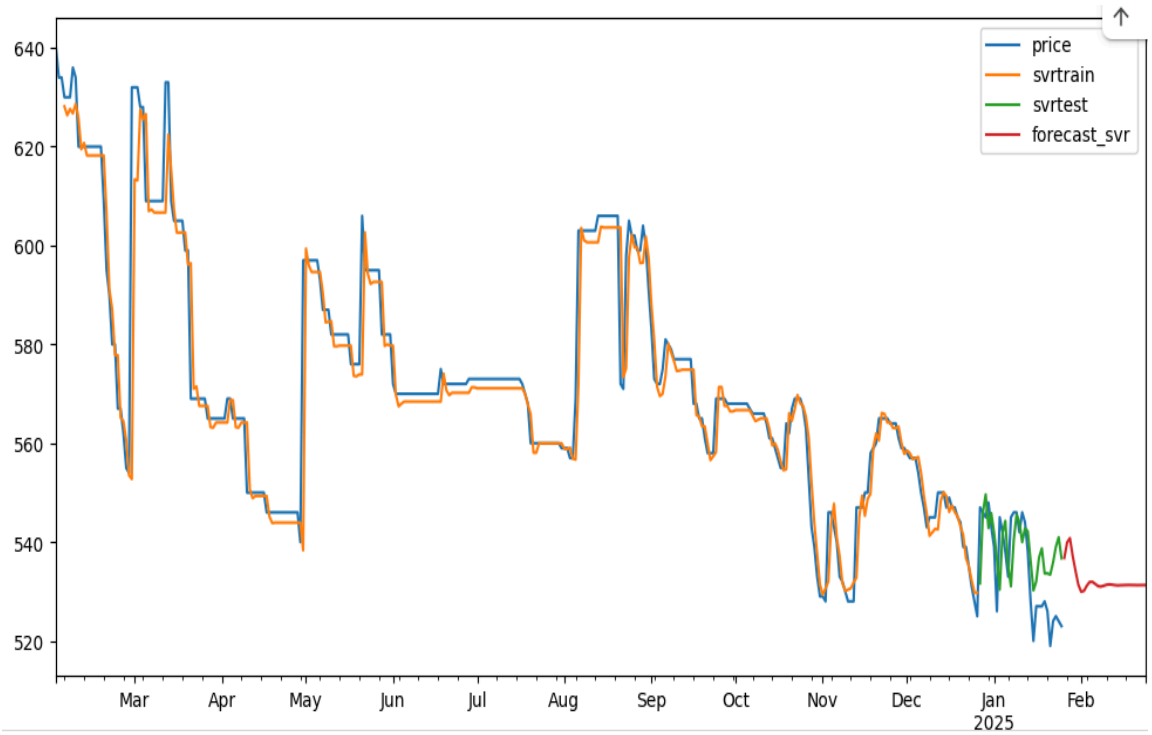
X(ΕΡΜΗΝΕΥΤΙΚΗ ΜΕΤΑΒΛΗΤΗ)	Y(ΕΞΑΡΤΗΜΕΝΗ ΜΕΤΑΒΛΗΤΗ)
[101,103,105]	107
[103,105,107]	109
[105,107,109]	111

Εν συνέχεια χρησιμοποιούμε τον standardscaler για την κανονικοποιηση των δεδομένων τα οποία εισάγουμε στο μοντέλο. Εφόσον έχουν γίνει τα παραπάνω προβλέπουμε τις τιμές για το train και test set και τις επόμενες 30 ημέρες μετα το τέλος του dataset και τις επαναφέρουμε στην αρχική τους μορφή ώστε να μπορεί να υπολογιστεί το mse μεταξύ των πραγματικών και τιμών που πρόβλεψε το μοντέλο. Η διαδικασία ολοκληρώνεται με την δημιουργία των επομένων 30 ημερομηνιών για την τοποθέτηση τους μέσα στο dataframe και η δημιουργία του αντίστοιχου διαγράμματος.

```
from sklearn.preprocessing import StandardScaler
from sklearn.svm import SVR
from sklearn.metrics import mean_squared_error
import matplotlib.pyplot as plt
ntest = 30
train = dfsvr.iloc[:-ntest]
test = dfsvr.iloc[-ntest:]
series = dfsvr.to_numpy()
T = 3
X = []
Y = []
for t in range(len(series) - T):
    x = series[t:t + T]
    X.append(x)
    y = series[t + T]
    Y.append(y)
X = np.array(X).reshape(-1, T)
Y = np.array(Y)
X_train = X[:-ntest]
Y_train = Y[:-ntest]
X_test = X[-ntest:]
Y_test = Y[-ntest:]
scaler_X = StandardScaler()
scaler_Y = StandardScaler()
X_train_scaled = scaler_X.fit_transform(X_train)
X_test_scaled = scaler_X.transform(X_test)
Y_train_scaled = scaler_Y.fit_transform(Y_train.reshape(-1, 1)).ravel()
```

```
sv = SVR()
sv.fit(X_train_scaled, Y_train_scaled)
Y_train_pred_scaled = sv.predict(X_train_scaled)
Y_test_pred_scaled = sv.predict(X_test_scaled)
Y_train_pred = scaler_Y.inverse_transform(Y_train_pred_scaled.reshape(-1, 1)).ravel()
Y_test_pred = scaler_Y.inverse_transform(Y_test_pred_scaled.reshape(-1, 1)).ravel()
train_idx = dfsvr.index <= train.index[-1]
test_idx = ~train_idx
train_idx[:T] = False
dfsvr = pd.DataFrame(dfsvr)
dfsvr['svrtrain'] = None
dfsvr['svrtest'] = None
dfsvr.loc[train_idx, 'svrtrain'] = Y_train_pred
dfsvr.loc[test_idx, 'svrtest'] = Y_test_pred
mse_trainsvr = mean_squared_error(Y_train, Y_train_pred)
mse_testsvr = mean_squared_error(Y_test, Y_test_pred)
print(f"Train MSE: {mse_trainsvr}")
print(f"Test MSE: {mse_testsvr}")
```

```
last_values = dfsvr['price'].values[-T:]
last_values_scaled = scaler_X.transform(last_values.reshape(1, -1))
future_dates = pd.date_range(start=dfsvr.index[-1] + pd.DateOffset(days=1), periods=30, freq='D')
future_df = pd.DataFrame(index=future_dates, columns=['price'])
dfsvr = pd.concat([dfsvr, future_df])
forecast_svr = []
for i in range(30):
  next_value_scaled = sv.predict(last_values_scaled)
  next_value = scaler_Y.inverse_transform(next_value_scaled.reshape(-1, 1))[0,0]
  forecast_svr.append(next_value)
  last_values = np.append(last_values[1:], next_value)
  last_values_scaled = scaler_X.transform(last_values.reshape(1, -1))
dfsvr['forecast_svr'] = None
dfsvr.loc[future_dates, 'forecast_svr'] = forecast_svr
dfsvr[['price', 'svrtrain', 'svrtest', 'forecast_svr']].plot(figsize=(12, 6))
plt.show()
dfsvr.tail(20)
```




```
Train MSE: 59.81644579513859
Test MSE: 83.21770827899459
```

Εικόνες 133:Κώδικας και εφαρμογή svr

Οι προβλέψεις του μοντέλου δεν είναι τόσο καλές όσο εκείνες της γραμμικής παλινδρόμησης σε ορούς μέσου τετραγωνικού σφάλματος στο test set και η αιτία είναι ότι έχει χρησιμοποιηθεί μικρός αριθμός περίοδών στο T με αποτέλεσμα όταν ζητάμε από το μοντέλο να προβλέψει 60 τιμές μετα από τα δεδομένα οπού έχει εκπαιδευτεί να μην μπορεί να αποδώσει τόσο καλές προβλέψεις. Ένας τρόπος για την βελτίωση του παραπάνω είναι η αλλαγή του Kernal από rbf σε άλλη μέθοδο όπως την linear ,η επιλογή μικρότερου c(η default τιμή είναι 10) ώστε να επιτρέπει στο μοντέλο να κάνει περισσότερα λάθη κατά την εκπαίδευση και κατά επέκταση να αναγνωρίσει καλυτέρα την χρονοσειρα και αντίστοιχα ένας μεγαλύτερος αριθμός epsilon (η default τιμή είναι 0.1) ώστε να επιτρέπονται μεγαλύτερες αποκλίσεις από τις πραγματικές τιμές και αρά το μοντέλο μπορεί να αναγνωρίσει καλυτέρα την τάση των δεδομένων.

5.2.11 RANDOM FOREST REGRESSION MACHINE LEARNING METHOD

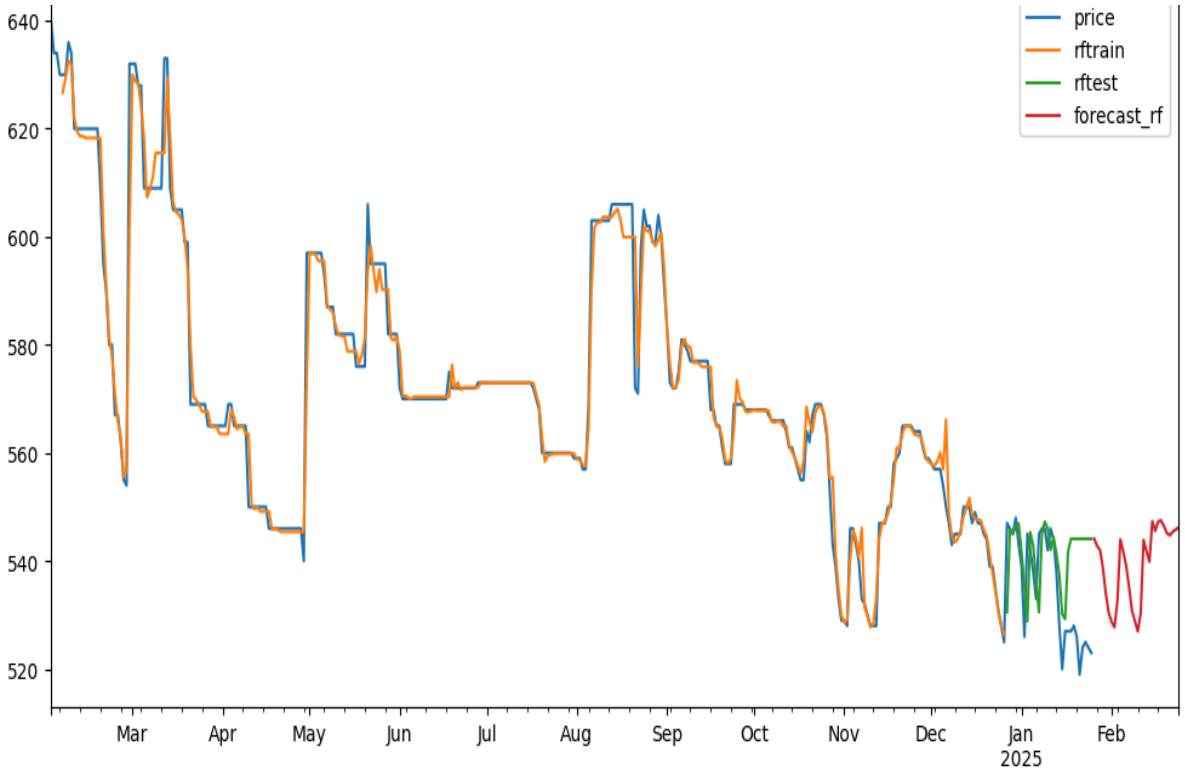
Την ιδιά μεθοδολογία ακολουθούμε και για την εφαρμογή του μοντέλου random forest regression οπού ακόμα μια φορά διαχωρίζουμε τα δεδομένα σε train και test set και επιλέγουμε ως T της τέσσερεις περιόδους. Εν συνέχεια κανονικοποιηουμε τα δεδομένα με τον standardscaler και τα μοντελοποιουμε με την συνάρτηση randomforestregressor. Στην συνέχεια προβλέπουμε τις τιμές για το test και train set και τις επαναφέρουμε στην αρχική τους μορφή , επίσης κάνουμε την αντίστοιχη πρόβλεψη για τις επόμενες 30 ημέρες εκτός του dataframe και όλες μαζί τις παρουσιάζουμε σε ένα διάγραμμα.

```
dfrf=s24.copy()
```

```
ntest = 30
train = dfrf.iloc[:-ntest]
test = dfrf.iloc[-ntest:]
series = dfrf.to_numpy()
T = 4
X = []
Y = []
for t in range(len(series) - T):
    x = series[t:t + T]
    X.append(x)
    y = series[t + T]
    Y.append(y)
X = np.array(X).reshape(-1, T)
Y = np.array(Y)
X_train = X[:-ntest]
Y_train = Y[:-ntest]
X_test = X[-ntest:]
Y_test = Y[-ntest:]
scaler_X = StandardScaler()
scaler_Y = StandardScaler()
X_train_scaled = scaler_X.fit_transform(X_train)
X_test_scaled = scaler_X.transform(X_test)
Y_train_scaled = scaler_Y.fit_transform(Y_train.reshape(-1, 1)).ravel()
```

```
rf = RandomForestRegressor()
rf.fit(X_train_scaled, Y_train_scaled)
Y_train_pred_scaled = rf.predict(X_train_scaled)
Y_test_pred_scaled = rf.predict(X_test_scaled)
Y_train_pred = scaler_Y.inverse_transform(Y_train_pred_scaled.reshape(-1, 1)).ravel()
Y_test_pred = scaler_Y.inverse_transform(Y_test_pred_scaled.reshape(-1, 1)).ravel()
dfrf = pd.DataFrame(dfrf)
dfrf['rftrain'] = None
dfrf['rftest'] = None
train_idx = dfrf.index <= train.index[-1]
test_idx = ~train_idx
train_idx[:T] = False
dfrf.loc[train_idx, 'rftrain'] = Y_train_pred
dfrf.loc[test_idx, 'rftest'] = Y_test_pred
mse_train_rf = mean_squared_error(Y_train, Y_train_pred)
mse_test_rf = mean_squared_error(Y_test, Y_test_pred)
print(f"Train MSE (Random Forest): {mse_train_rf}")
print(f"Test MSE (Random Forest): {mse test rf}")
```

```
last_values = dfrf['price'].values[-T:]
last_values_scaled = scaler_X.transform(last_values.reshape(1, -1))

future_dates = pd.date_range(start=dfrf.index[-1] + pd.DateOffset(days=1), periods=30, freq='D')
future_df = pd.DataFrame(index=future_dates, columns=['price'])
dfrf = pd.concat([dfrf, future_df])

forecast_rf = []
for i in range(30):
  next_value_scaled = rf.predict(last_values_scaled)
  next_value = scaler_Y.inverse_transform(next_value_scaled.reshape(-1, 1))[0,0]
  forecast_rf.append(next_value)
  last_values = np.append(last_values[1:], next_value)
  last_values_scaled = scaler_X.transform(last_values.reshape(1, -1))

dfrf['forecast_rf'] = None
dfrf.loc[future_dates, 'forecast_rf'] = forecast_rf

dfrf[['price', 'rftrain', 'rftest', 'forecast_rf']].plot(figsize=(12, 6))
```




```
Train MSE (Random Forest): 17.066657968796793
Test MSE (Random Forest): 149.67603019912158
```

Εικονες 134:Κωδικας και εφαρμογη random forest

Το αποτέλεσμα του παραπάνω αλγορίθμου χωρίς να γίνουν αλλαγές στον τρόπο εκτέλεσης του δηλαδή να μην επιλέγουν χειροκίνητα ο αριθμός των δέντρων(n_estimators) η συνάρτηση υπολογισμού του σφάλματος(criterion) ή το βάθος του δέντρου(max_depth) δείχνει ότι το μοντέλο κάνει overfit στα δεδομένα εκπαίδευσης με αποτέλεσμα το αντίστοιχο mse να είναι πολύ μικρό κοντά στο 17 ενώ από την άλλη πλευρά οι προβλέψεις του μοντέλου για το test σετ να είναι χειρότερες από εκείνες του Arima. Αναφορικά με τις προβλέψεις για τις 30 τιμές εκτός του dataset φαίνεται το μοντέλο να αντιγράφει την μείωση της τιμής κατά τους μήνες Νοέμβριο και Δεκέμβριο του 2024.

5.2.12 FACEBOOK PROPHET MACHINE LEARNING METHOD

Σε αυτή την ενότητα θα χρησιμοποιήσουμε το Facebook prophet για την πρόβλεψη της μελλοντικής τιμής του κινητού τηλεφώνου, πριν όμως από αυτό θα εξηγήσουμε το τρόπο με τον οποίο λειτουργεί/οδηγείται στις εκάστοτε προβλέψεις το συγκεκριμένο μοντέλο μηχανικής μάθησης. Το Facebook prophet είναι μια βιβλιοθήκη ανοικτού κώδικα που

αναπτύχθηκε από την ερευνητική ομάδα του Facebook το 2017. Στόχος της βιβλιοθήκης είναι να βοηθήσει μικρές αλλά και μεγάλες επιχειρήσεις στην ανάλυση της ζήτησης των προϊόντων της , την διαχείριση αποθεμάτων , την αποδοτικότητα διασωστικών και άλλων ειδών προώθησης προϊόντων και γενικότερα η ανάλυση χρονολογικών σειρών. Το θετικό της χρήσης της βιβλιοθήκες πέραν από το γεγονός ότι είναι δωρεάν η χρήση της είναι το γεγονός ότι είναι πολύ εύκολο να χρησιμοποιηθεί από άτομα που δεν έχουν καμία προηγουμένη επαφή με τις μεθοδολογίες μηχανικής μάθησης και τον προγραμματισμό καθώς όπως θα δούμε και στην συνέχεια ο κώδικας που απαιτείται να γραφτεί για την δημιουργία προβλέψεων αποτελείται από περίπου πέντε γραμμές. Όσον αφορά την λειτουργεία του μοντέλου αυτό μπορεί και αναλύει την τάση την εποχικότητα την και την κυκλικότητα των δεδομένων, επίσης υπάρχει μια μεταβλητή h που προκύπτει από την λέξη holidays για την ανάλυση των δεδομένων σε περιόδους γιορτών όπως τα Χριστούγεννα το Πάσχα το καλοκαιρή την περίοδο των εκπτώσεων όπως το black Friday το Halloween κλπ. και ουσιαστικά το μοντέλο προσπαθεί να μοντελοποιησει τον τρόπο με τον οποίο επηρεάζουν (μεταβάλουν) την εξαρτημένη τιμή(πχ τιμή του Samsung galaxy s24) οι συγκεκριμένες γιορτές. Η μαθηματική formula με την οποία προκύπτουν οι προβλέψεις της βιβλιοθήκης του Facebook prophet είναι η παρακάτω

$$y(t) = g(t) + s(t) + h(t) + \varepsilon t$$

Οπού Το g(t) είναι η συνάρτηση που αναλύει της τάση και μοντελοποιει τις μη περιοδικές αλλαγές στην αλλαγή των τιμών τις χρονοσειρας.

Το s(t) είναι η συνάρτηση που μοντελοποιει τις περιοδικές αλλαγές στην χρονοσειρα όπως την ετήσια, μηνιαία ή και την εβδομαδιαία εποχικότητα.

Το η(τ) αναπαριστά την επιρροή των διακοπών(holidays) οι οποίες εμφανίζονται σε διαφορετικές περιόδους ανά έτος στο σύνολο της χρονοσειρας.

Το ε συμβολίζει το σφάλμα το οποίο είναι κανονικοποιημενο δηλαδή παίρνει τιμές από 0 έως 1.

Πιο αναλυτικά το g(t) στο Facebook prophet είναι αποτέλεσμα της παρακάτω εξίσωσης

$$g(t) = (k + a(t) \mid \delta)t + (m + a(t)\text{^T}\gamma)$$

αυτή η εξίσωση έχει τις ρίζες της από το logistic growth model

$$g(t) = \frac{c}{1+exp(-k(t-m)}$$

Μόνο που σε αντίθεση με αυτό η παράμετρος c(δηλαδή το συνολικό επίπεδο growth πχ τα άτομα που έχουν κινήτο σε μια πόλη) του prophet δεν έχει σταθερή τιμή αλλά αντίθετος αυξάνεται η μειώνεται ανάλογος με τα δεδομένα της χρονοσειρας(δηλαδή όπως και σε μια πόλη μπορούν να αυξηθούν οι χρήστες κινητών συσκευών). Επίσης η άλλη μεταβλητή που διαφέρει από την κλασσική συνάρτηση είναι ο συντελεστής ανάπτυξης οπού και αυτός δεν είναι σταθερός στο prophet αλλά αλλάζει ανάλογα τα changepoints της χρονοσειρας τα οποία υπολογίζονται από τον παρακάτω τύπο.

$$\gamma_j = \left(s_j - m - \sum_{l<j} \gamma_l\right)\left(1 - \frac{k + \sum_{l<j} \delta_l}{k + \sum_{l \leq j} \delta_l}\right).$$

Οπού το m είναι μια παράμετρος αντιστάθμισης και το δ προκύπτει από την κατανομή LaPlace (0,t)

Όσον αφορά το s(t) δηλαδή την εποχικότητα αυτή υπολογίζεται από μια συνάρτηση Fourier οπού π είναι η περίοδος και παίρνει την τιμή 7 για εβδομαδιαία δεδομένα και 365,25 για ετήσια. Το ν είναι οι παράμετροι Fourier οπού το prophet επιλεγεί αυτόματα ν=10 για ετήσια δεδομένα και ν=3 για εβδομαδιαία έτσι ώστε να αποφευχθεί το overfitting.

$$s(t) = \sum_{n=1}^{N} \left(a_n \cos\left(\frac{2\pi nt}{P}\right) + b_n \sin\left(\frac{2\pi nt}{P}\right)\right)$$

Και τέλος σχετικά με τις διακοπές και τα μεγάλα γεγονότα (events) και αυτά λαμβάνονται υπόψη στο μοντέλο του prophet καθώς επηρεάζουν την ζήτηση των προϊόντων. Συγκεκριμένα οι διακοπές μπαίνουν μέσα στο μοντέλο ως dummies δηλαδή ψευδομεταβλητες που όταν ισχύει κάτι περνούν την τιμή 1 αλλιώς είναι ίση με μηδέν. Για

παράδειγμά για να δούμε την επίδραση των Χριστουγέννων σε μια χρονοσειρα μπορούμε να φτιάξουμε μια δεύτερη στήλη οπού όλες οι τιμές της θα είναι 0 εκτός από τις ημέρες που είναι λίγο πριν και λίγο μετα τα Χριστούγεννα (25/12/χχχχ). Το prophet έχει ήδη προ εγκατεστημένα κάποια από τα μεγαλύτερα events σε διάφορες χώρες όπως την Αμερική.

Οι ημερομηνίες των διακοπών εισάγονται σε ένα array

$$Z(t) = [1(t \in D1), \ldots, 1(t \in DL)]$$

Και στην συνέχεια πολλαπλασιάζονται με μια τιμή κ που καθορίζει την επίδραση των διακοπών.

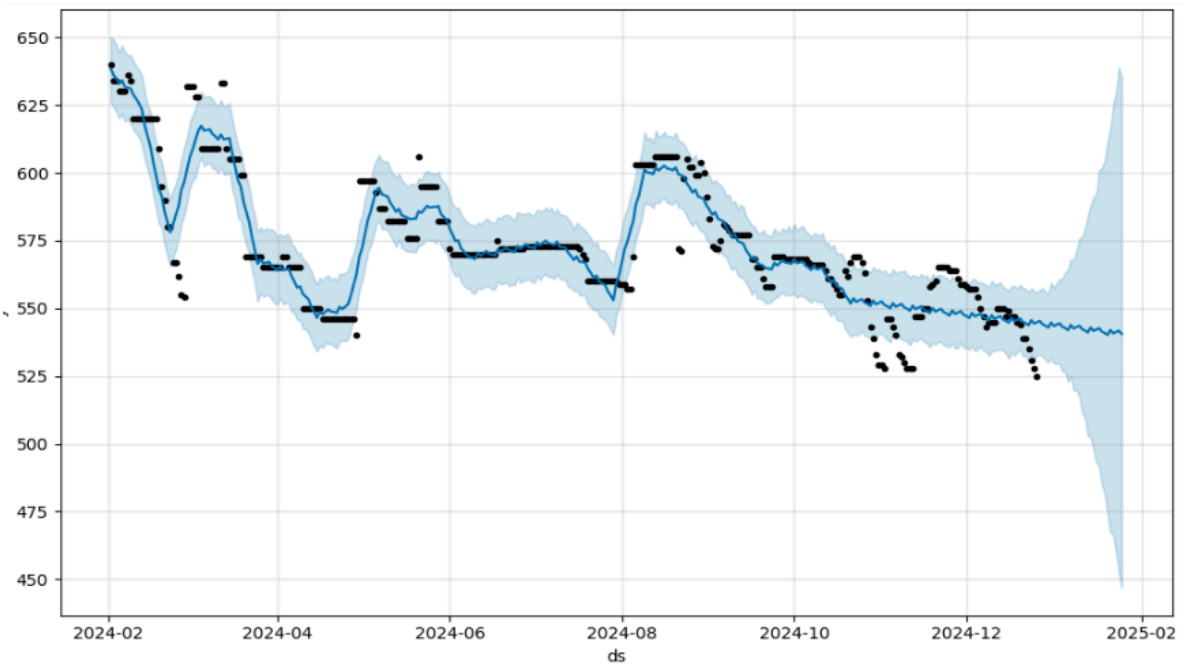
$$h(t) = Z(t)\text{κ}.$$

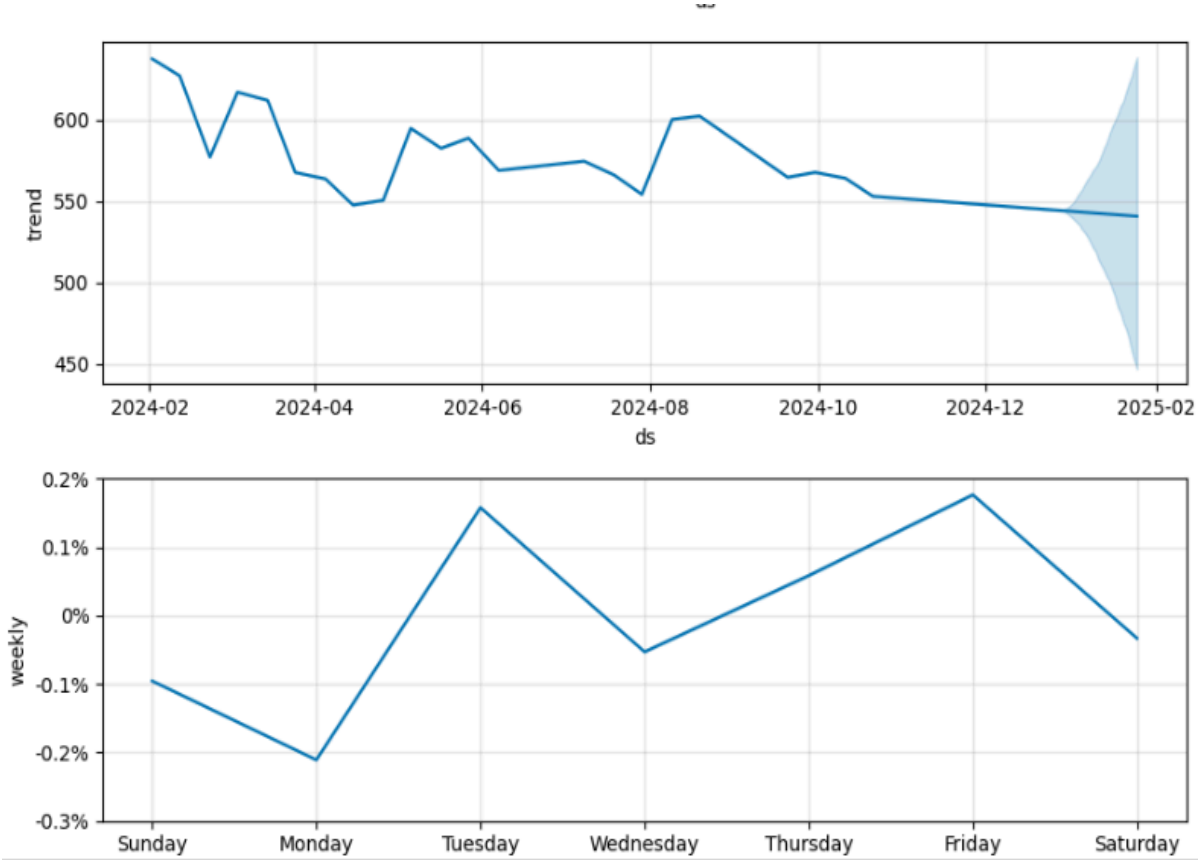
Στην συνέχεια θα εφαρμόσουμε το μοντέλο του Facebook prophet για την πρόβλεψη της τιμής του κινητού τηλεφώνου στην python. Αγρικά εισάγουμε τις βιβλιοθήκες όπως την NumPy την pandas και την matplotlib και κάνουμε εγκατάσταση την βιβλιοθήκη prophet. Εν συνέχεια εισάγουμε ξανά από το excel την χρονοσειρα με τις τιμές του κινητού τηλεφώνου σε ένα dataframe με όνομα dfprophet , αυτό που δεν πρέπει να ξεχάσει ο αναλυτής κατά την εφαρμογή του μοντέλου είναι να μετονομάσει την στήλη με την εξαρτημένη μεταβλητή δηλαδή την μεταβλητή για την οποία θα ζητήσει από το prophet να δημιουργήσει προβλέψεις (στην περίπτωση μας τις τιμές του κινητού τηλεφώνου) σε y και την στήλη με τις ημερομηνίες τις χρονοσειρας σε ds. Το επόμενο βήμα είναι παρόμοιο με όλα τα προηγούμενα μοντέλα και είναι ο διαχωρισμός των δεδομένων της χρονοσειρας σε test και train set. Έχοντας ολοκληρώσει τα παραπάνω δημιουργούμε το μοντέλο prophet το οποίο εισάγουμε στην μεταβλητή m , για το μοντέλο μετα από αρκετές δοκιμές έχει επιλεγεί ως τρόπος πρόβλεψης του μοντέλου (growth) η γραμμική τάση (linear) δηλαδή ότι αναμένουμε η τάση της χρονοσειρας να αυξάνεται ή να μειώνεται με σταθερό ρυθμό , επίσης έχει επιλεχθεί για τιμή του change_prio_scale το 1 δηλαδή το μοντέλο είναι πολύ ευαίσθητο σε αλλαγές τις τάσης και ως seasonality_mode η πολλαπλασιαστική (multiplicative) καθώς αυτή είχε εφαρμοστεί και στο μοντέλο holt winters και απέδωσε καλυτέρα αποτελέσματα σε σχέση με την προσθετική(additive). Επίσης προσθέτουμε και εβδομαδιαία εποχικότητα στο μοντέλο με περίοδο 7 ημέρων και χρησιμοποιούμε 5 συντελεστές Fourier. Ολοκληρώνοντας τα παραπάνω εκπαιδεύουμε το μοντέλο με την εντολή m.fit και δημιουργούμε ένα dataframe στο οποίο θα τοποθετηθούν οι επόμενες 60 τιμές που θέλουμε το μοντέλο να προβλέψει (30 για το test set και 30 για τις ημερομηνίες

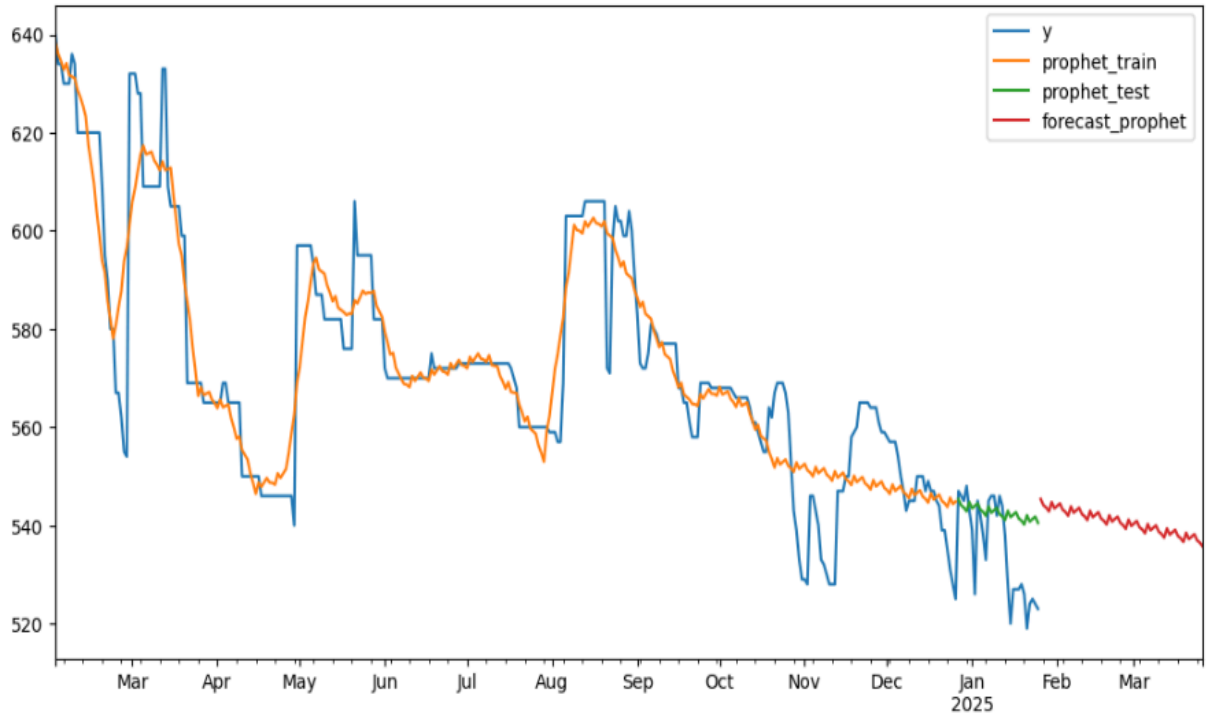
εκτός του dataframe) η πρόβλεψη γίνεται μέσο της εντολής predict. Σε αυτό το σημείο μπορεί να υπολογιστεί και το mse μεταξύ του train test και των πραγματικών δεδομένων και αντίστοιχα του test set με τις πραγματικές τιμές του κινητού. Τέλος ολοκληρώνουμε τον κώδικα δημιουργώντας δυο διαγράμματα το πρώτο δείχνει την εποχικότητα και την τάση της χρονοσειρας, ενώ στο δεύτερο εμφανίζονται οι πραγματικές τιμές της χρονοσειρας και συγκρίνονται με τις προβλέψεις του μοντέλου.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from prophet import Prophet
from sklearn.metrics import mean_squared_error
dfprophet = pd.read_excel('s24_128gb_price.xlsx')
dfprophet = dfprophet.rename(columns={'date': 'ds', 'price': 'y'})
ntest = 30
train = dfprophet.iloc[:-ntest].copy()
test = dfprophet.iloc[-ntest:].copy()
m = Prophet(growth='linear',changepoint_prior_scale=1,seasonality_mode='multiplicative',seasonality_prior_scale=10)
m.add_seasonality(name='weekly', period=7, fourier_order=5)
m.fit(train)
future = m.make_future_dataframe(periods=ntest, freq='D')
forecast = m.predict(future)
train_predictions = forecast['yhat'][:-ntest]
test_predictions = forecast['yhat'][-ntest:]
mse_train = mean_squared_error(train['y'], train_predictions)
mse_test = mean_squared_error(test['y'], test_predictions)
print(f"Train MSE: {mse_train}")
print(f"Test MSE: {mse_test}")
fig1 = m.plot(forecast)
fig2 = m.plot_components(forecast)
dfprophet['prophet_train'] = None
dfprophet['prophet_test'] = None
train_idx = dfprophet['ds'] <= train['ds'].iloc[-1]
```

```
train_idx = dfprophet['ds'] <= train['ds'].iloc[-1]
test_idx = ~train_idx
dfprophet.loc[train_idx, 'prophet_train'] = train_predictions.values
dfprophet.loc[test_idx, 'prophet_test'] = test_predictions.values
future_forecast = m.make_future_dataframe(periods=60, freq='D')
forecast_prophet = m.predict(future_forecast)
forecast_prophet_values = forecast_prophet['yhat'][-60:]
future_dates = pd.date_range(start=dfprophet['ds'].max() + pd.DateOffset(days=1), periods=60, freq='D')
future_df_prophet = pd.DataFrame({'ds': future_dates})
dfprophet = pd.concat([dfprophet, future_df_prophet], ignore_index=True)
dfprophet['forecast_prophet'] = None
dfprophet.loc[dfprophet.index[-60:], 'forecast_prophet'] = forecast_prophet_values.values
dfprophet.plot(x='ds', y=['y', 'prophet_train', 'prophet_test', 'forecast_prophet'], figsize=(12, 6))
plt.show()
```


650
625
600
575
550
525
500
475
450
2024-02
2024-04
2024-06
2024-08
2024-10
2024-12
2025-02
ds


trend
600
550
500
450
2024-02
2024-04
2024-06
2024-08
2024-10
2024-12
2025-02
ds
weekly
0.2%
0.1%
0%
-0.1%
-0.2%
-0.3%
Sunday
Monday
Tuesday
Wednesday
Thursday
Friday
Saturday




```
Train MSE: 93.03458627301437
Test MSE: 132.75280861982006
```

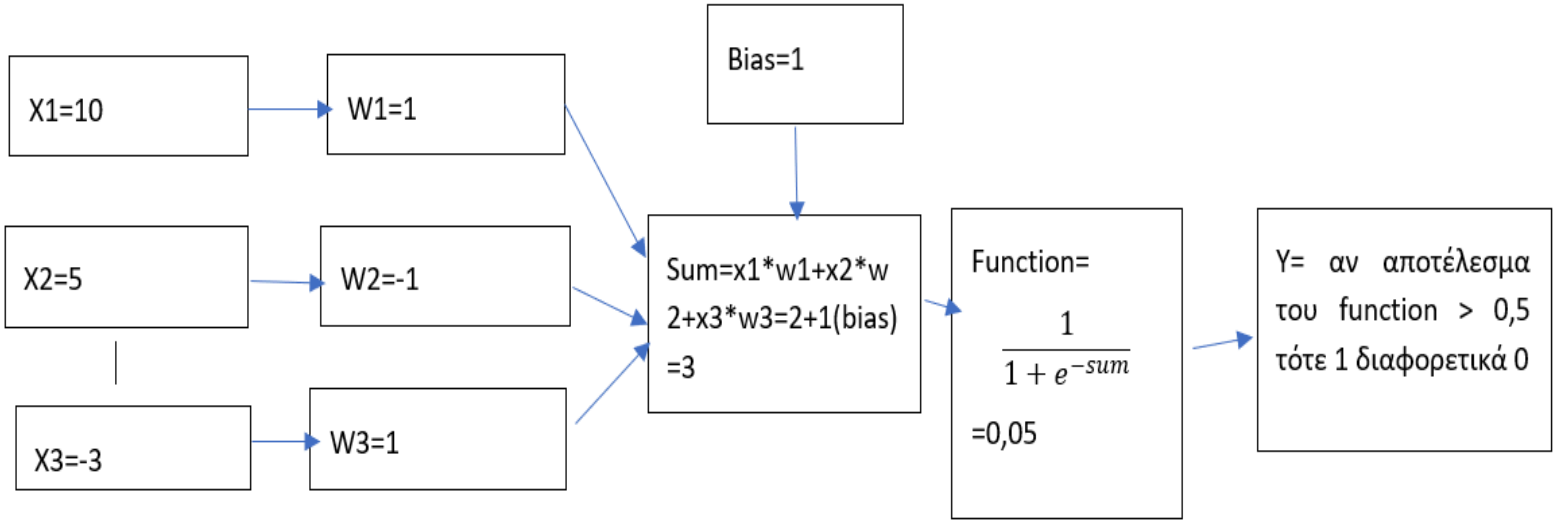
Εικόνες 134: κώδικας και εφαρμογή Facebook prophet

Τα συμπεράσματα τα οποία μπορούν να εξαχθούν από την εφαρμογή του μοντέλου είναι ότι η χρονοσειρα έχει καθοδική τάση σε συνδυασμό με εποχικότητα αναφορικά με την εβδομαδιαία εποχικότητα παρατηρούμε ότι η τιμή του κινητού είναι χαμηλότερη(σε ποσοστιαία βάση) τις δεύτερες ενώ τις τρίτες και τις παρασκευές η τιμή είναι αρκετά υψηλότερη σε σχέση με τις υπόλοιπες ημέρες. Επίσης παρατηρούμε ότι το μοντέλο ακόμα και με την παραμετροποίηση επιτυγχάνει ένα mse στο train set υψηλότερο από τα αντίστοιχα των άλλων μοντέλων μηχανικής μάθησης που έχουμε ήδη χρησιμοποιήσει , για παράδειγμά το μοντέλο δεν έχει κατανοήσει καλά την περίοδο πριν και μετα των εκπτώσεων του black Friday και των Χριστουγέννων οπού η τιμή αυξάνεται και στην συνέχεια μειώνεται δηλαδή αποφεύγεται το overfit αλλά από την άλλη πλευρά το μοντέλο φαίνεται να κάνει underfit τα δεδομένα. Βέβαια αναφορικά με τις προβλέψεις του για το μηνά Φεβρουάριο του 2025 φαίνεται να έχει καταλάβει την πτωτική πορεία της τιμής αλλά όπως θα δούμε και στην συνέχεια προβλέπει πολύ υψηλότερες τιμές από αυτές που όντος διαμορφώθηκαν στα καταστήματα για τον ίδιο μηνά.

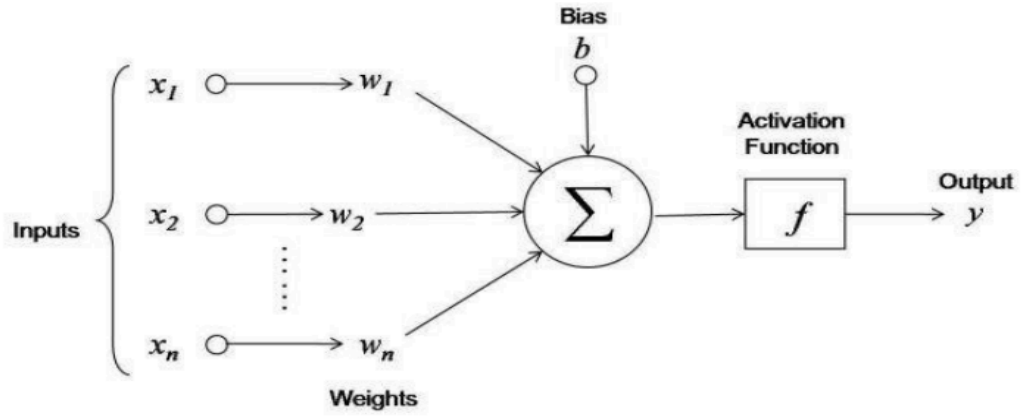
5.2.13 ΕΙΣΑΓΩΓΗ ΣΤΑ ΝΕΥΡΩΝΙΚΑ ΔΥΚΤΙΑ ΚΑΙ ΕΦΑΡΜΟΓΗ ΤΟΥ LSTM

Σε αυτήν την ενότητα θα παρουσιάσουμε τα βασικά χαρακτηριστικά των νευρωνικων δικτυών και στην συνέχεια θα εφαρμόσουμε το μοντέλο του lstm για την πρόβλεψη της τιμής του κινητού τηλεφώνου s24 εξηγώντας ταυτόχρονα τον κώδικα για την ανάπτυξη του μοντέλου. Αρχικά θα παρουσιάσουμε τον πρώτο νευρωνικο δίκτυο με ονομασία perceptron το οποίο αναπτύχθηκε την δεκαετία του 1950 από τον ερευνητή frank Rosenblatt με στόχο να προσομοιώσει την λειτουργία του δικτύου με τις βασικές λειτουργίες του ανθρωπίνου εγκέφαλου. Το συγκεκριμένο νευρωνικό δίκτυο χρησιμοποιήθηκε και χρησιμοποιείται σε μικρότερο βαθμό στις μέρες μας για απλή δυαδική ταξινόμηση των δεδομένων δηλαδή αν τα δεδομένα ανήκουν την κατηγορία 1 ή 0 (ναι ή όχι).

Αναφορικά με τον τρόπο λειτουργείας του perceptron το μοντέλο λαμβάνει ως είσοδο ένα σύνολο δεδομένων χ1,χ2,…χν και στην συνέχεια κάθε ένα από αυτά πολλαπλασιάζεται με τα αντίστοιχα βάρη(W) και μια σταθερά που ονομάζεται bias η όποια βοηθάει το μοντέλο να λάβει την καλύτερη δυνατή απόφαση. Στην συνέχεια το μοντέλο αθροίζει όλες αυτές τις τιμές οι οποίες τοποθετούνται σε μια συνάρτηση ενεργοποίησης (tanh, relu, sigmoid κλπ.) και εάν το αποτέλεσμα που προκύψει είναι μικρότερο του 0,5(ή κάποιες φορές μεγαλύτερο του 0) το μοντέλο ταξινομεί τα δεδομένα στην κατηγορία 0 εάν από την άλλη πλευρά το αποτέλεσμα είναι μεγαλύτερο του 0,5 (ή του μηδενός) τότε το μοντέλο ταξινομεί τα δεδομένα στην κατηγορία 1. Ένα παράδειγμα εφαρμογής του παραπάνω παρουσιάζεται στην παρακάτω εικόνα.



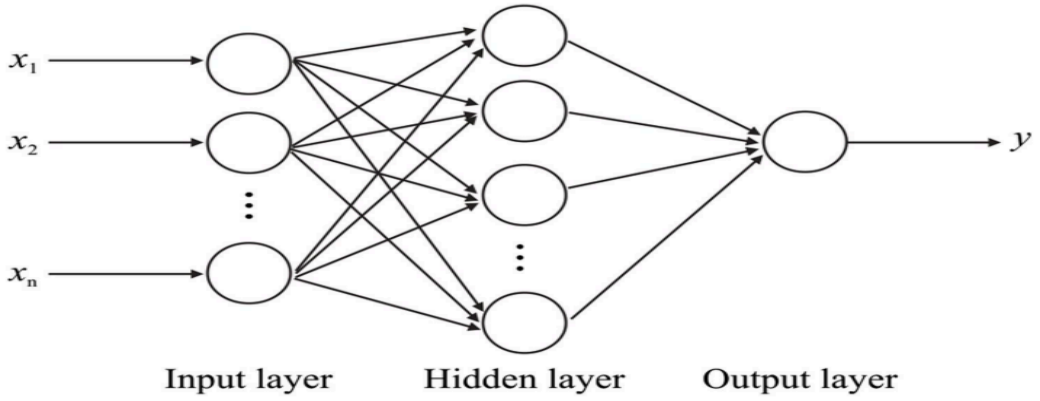
εικόνα 135: παράδειγμα preceptron



εικόνα 136: νευρωνας preceptron

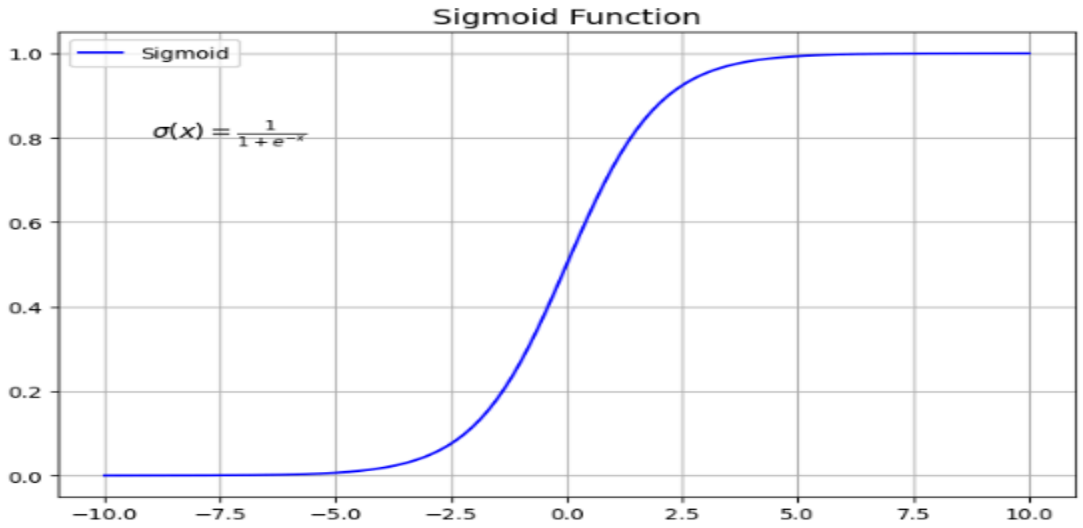
Η εξέλιξη του απλού perceptron είναι τα πολυεπιπεδα perceptron's τα όποια περιλαμβάνουν πολλαπλά στρώματα / επίπεδα νευρώνων για την επίτευξη καλύτερης ταξινόμησης των δεδομένων. Τα συγκεκριμένα μοντέλα αποτελούνται από τρία κύριος επίπεδα. Το πρώτο επίπεδο είναι εκείνο της εισόδου των δεδομένων χ οπού κάθε ερμηνευτική μεταβλητή εισέρχεται σε έναν διαφορετικό νευρώνα από τις υπόλοιπες. Το δεύτερο επίπεδο ονομάζεται hidden layers (κρυφά επίπεδα) το οποίο αποτελείται από πολλαπλούς νευρώνες perceptron οπού ο καθένας εφαρμόζει τις πράξεις που αναφέρθηκαν παραπάνω(πολλαπλασιασμός βαρών προσθήκη bias κλπ.) , εξάγει το αποτέλεσμα και το μεταβιβάζει ως είσοδο στους επομένους νευρώνες. Τέλος το τρίτο επίπεδο ή αλλιώς

επίπεδο εξόδου αποτελείται συνηθώς από έναν νευρώνα ειδικά στην περίπτωση οπού ο νευρώνας χρησιμοποιείται για μοντέλα ανάλυσης χρονοσειρων στο οποίο εξάγεται το τελικό αποτέλεσμα του δικτύου. Αναφορικά με την λειτουργία του δικτύου όπως αναφέρθηκε και παραπάνω αρχικά ο κάθε νευρώνας ξεχωριστά υπολογίζει και εξάγει μια τιμή η όποια είναι αποτέλεσμα του πολλαπλασιασμού των δεδομένων εισόδου χ με τα βάρη και το bias και εφαρμόζοντας την συνάρτηση ενεργοποίησης. Βέβαια σε αυτό το σημείο θα πρέπει να αναφερθούμε στο αλγόριθμο αντίστροφης διάδοσης σφαλμάτων οπού έχει ως στόχο να μετρήσει το σφάλμα που προκύπτει μετα από κάθε εκτέλεση του νευρωσικού δικτύου και να τον καθοδηγήσει στην ελαχιστοποίηση του σφάλματος μέσο της αλλαγής των βαρών που πολλαπλασιάζονται με τις μεταβλητές που εισάγονται. Υπάρχουν διάφοροι backpropagation αλγόριθμοι ο ποιο διαδεδομένος και αυτός που θα χρησιμοποιήσουμε στο παράδειγμά μας είναι ο Adam(adaptive moment estimation) καθώς επιτυγχάνει γρήγορη σύγκληση των δεδομένων μέσο του υπολογισμού των πρώτων παράγωγων της συνάρτησης κόστους και στην συνέχεια ενημερώνει τα βάρη προς την αντίθετη κατεύθυνση από το αποτέλεσμα της παραγωγού ,αυτό το επιτυγχάνει μέσο της εκτίμησης της πρώτης και δεύτερης ροπής (momentum για την ομαλή ενημέρωση των βαρών και rmsprop για την δυναμική προσαρμογή του ρυθμού εκμάθησης).



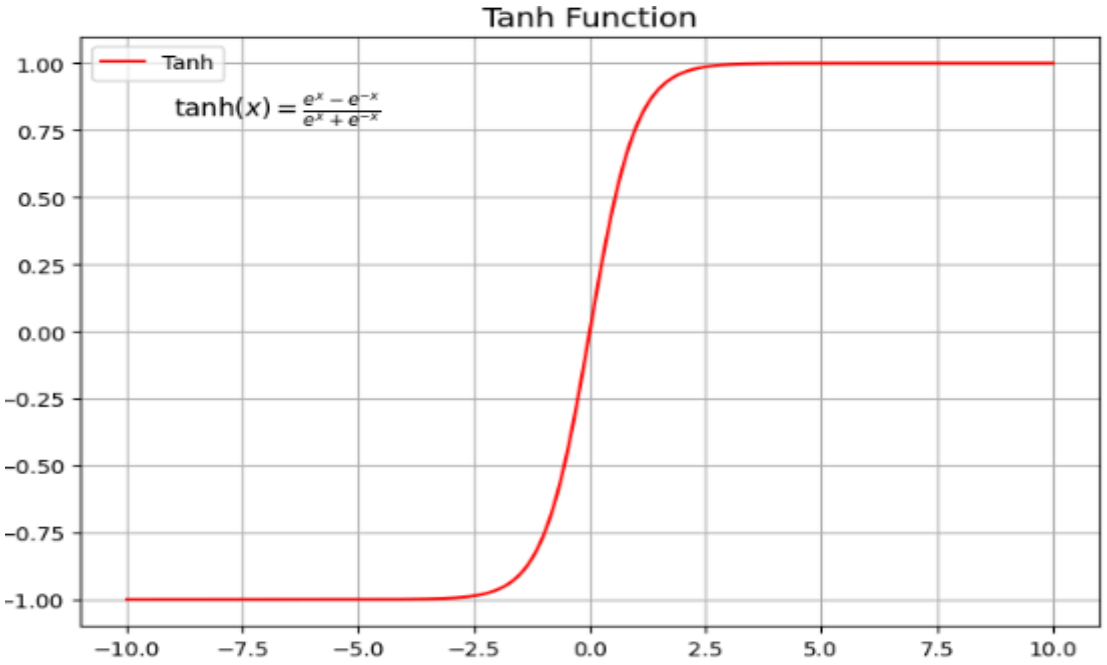
Εικονα 137: multi-layer neural network

Αναφορικά με τις συναρτήσεις ενεργοποίησης των νευρώνων οι πιο γνωστές είναι η sigmoid η όποια χρησιμοποιείται κυρίως σε προβλήματα ταξινόμησης καθώς οι τιμές εξόδου των νευρώνων περνούν τιμές μόνο από 0 έως 1.



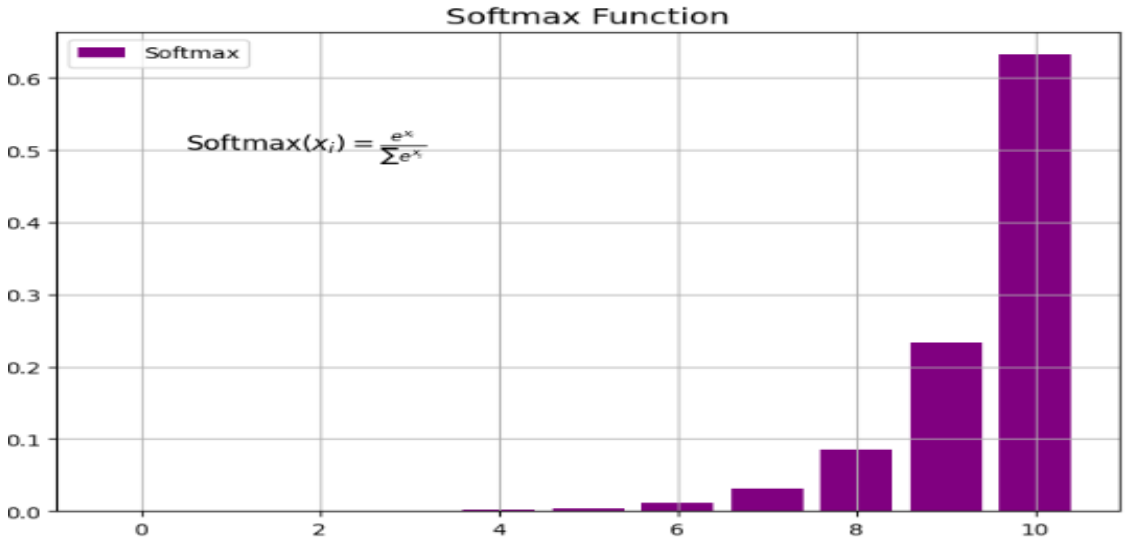
Εικονα 137: sigmoid

Στην συνέχεια υπάρχει η tanh η όποια μοιάζει με την sigmoid αλλά το διάστημα των τιμών που περνούν οι μεταβλητές οι οποίες εισέρχονται σε αυτή είναι από το -1 έως το 1 με αποτέλεσμα ο νευρώνας να ενεργοποιείται περισσότερες φορές σε σχέση με την sigmoid λόγο του μεγαλύτερου εύρους τιμών.



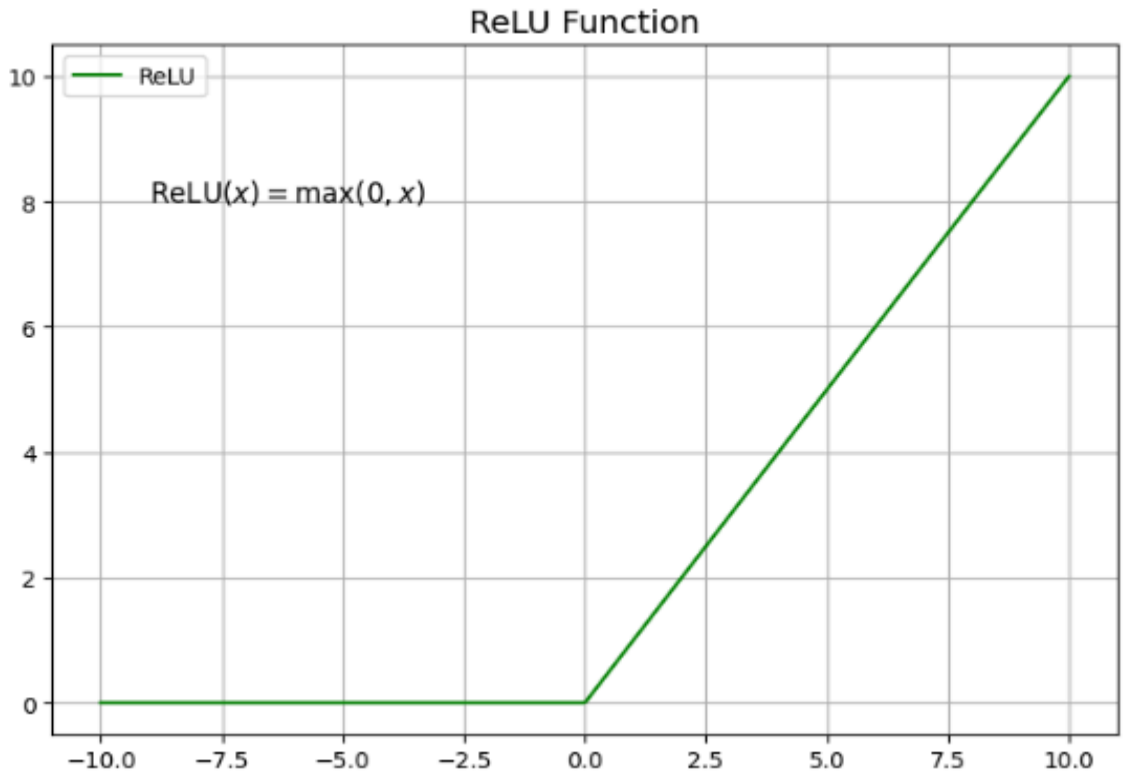
Εικονα 138: tanh

Επιπλέον υπάρχει και η συνάρτηση ενεργοποίησης SoftMax η οπού οι τιμές εξόδου της έχουν ευρέως από 0 έως 1.



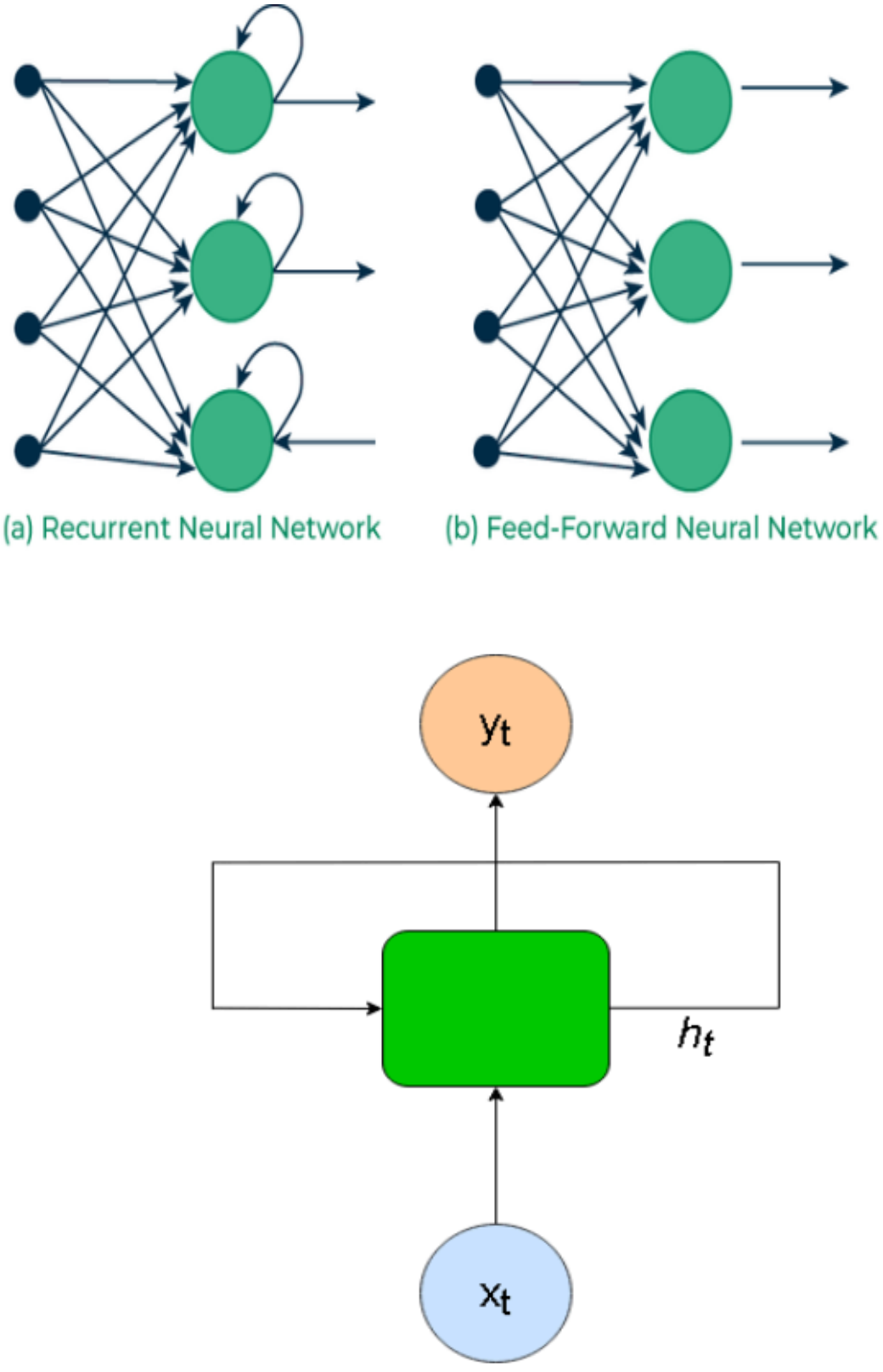
Εικονα 139: softmax

Τέλος υπάρχει και η συνάρτηση διορθωμένης γραμμικής μονάδας Relu οπού δεν ενεργοποιεί τον νευρώνα αν η τιμή του είναι μικρότερη του μηδενός με αποτέλεσμα να αποφεύγεται το πρόβλημα των εξαφανιζόμενων κλίσεων κατά την εκπαίδευση των νευρώνων. Δηλαδή όπως είδαμε και πάνω μέσο του αλγορίθμου backpropagation υπολογίζονται κάποιοι παραγωγοί της loss function αν αυτοί οι παραγωγοί έχουν τιμή μικρότερη του 1 τότε μέσο της πολλαπλασιαστικής διαδικασίας μειώνουν την τιμή και των επομένων νευρώνων με αποτέλεσμα αν αυτή η διαδικασία συνεχιστεί οι παράγωγοι στα πρώτα στάδιά να μηδενιστούν και αρά δεν ενημερώνονται τα βάρη όλου του νευρωνικου δικτύου. Την λύση στο πρόβλημα δίνει η relu οπού απενεργοποιεί τελείως τους νευρώνες που μηδενίζονται καθώς δεν θέλουμε να μην εκπαιδεύονται οι αρχικοί νευρώνες έξαλλου αυτοί είναι οι πρώτοι που έρχονται σε επαφή με τα δεδομένα εισόδου.

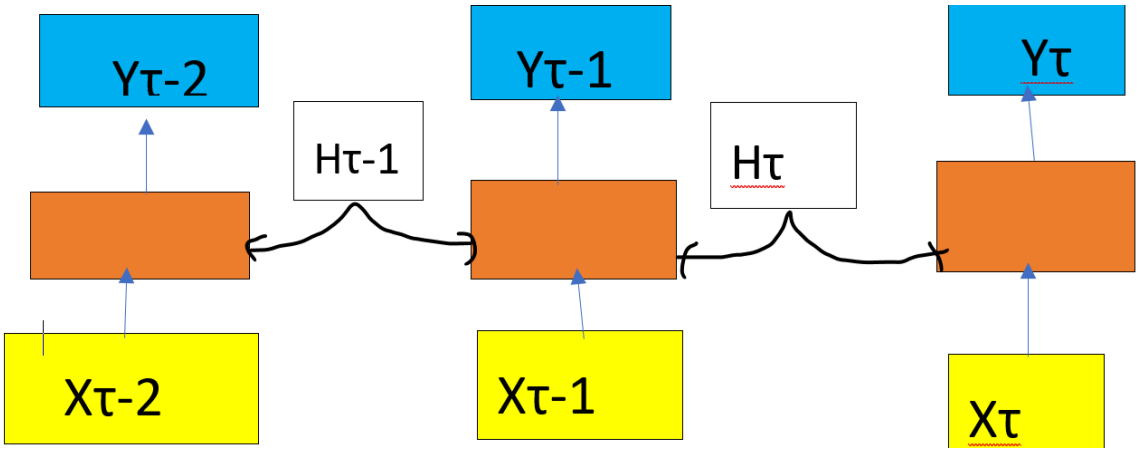


Εικονα 140: relu

Το πρόβλημα που δημιουργείται με τους απλούς perceptron νευρώνες είναι ότι το αποτέλεσμα του νευρώνα γίνεται είσοδος στον νευρώνα του επομένου επιπέδου χωρίς αυτό να μπορεί να αποθηκευτεί στο νευρώνα που δημιούργησε αυτό το output με αποτέλεσμα να μην μπορεί να δημιουργηθεί συσχέτιση μεταξύ των δεδομένων. Το πρόβλημα της μνήμης του νευρώνα το λύνουν τα αναδρομικά νευρωνικα δίκτυα στα όποια το hidden layer τους λειτουργεί ως μνήμη για τον εκάστοτε νευρώνα αλλά και συνολικά για το νευρωνικο δίκτυο καθώς και αυτό ξανά υπολογίζεται κατά την εκπαίδευση του δικτιού μέσο του backpropagation αλγορίθμου. Αρά τώρα όταν ένας νευρώνας προσπαθεί να υπολογίσει την επόμενη τιμή μιας χρονοσειρας για χρησιμοποιεί τα δεδομένα της χρονικής στίμης τ αλλά έχει αποθηκεύσει μέσα του και τα δεδομένα της χρονικής όλων των προηγουμένων νευρώνων μέσο του hidden layer.



Εικονα 141:Recurrent neural network



Οπού το Υτ είναι αποτέλεσμα της έστω function sigmoid(Wy+ht+ bias)

Οπού το Wy (είναι τα υπολογισμένα βάρη)

Και ht είναι ht=sigmoid(wx*xt+wh*ht+bias)

Το wx είναι τα βαρη των μεταβλητων εισοδου τα wh είναι τα βαρη των hidden layouts και wy τα βαρη των δεδομενων εξοδου. αρα παρατηρουμε ότι το ht εχει πλεον ολλη την πληροφορια ολλων των προηγουμρνων νευρωνων. Επισης όπως είναι φανερο ότι κατά την εκπαιδευση των rnn ο backpropagation αλγοριθμος πρεπει κάθε φορα να υπολογιζει όλα τα βαροι και για τις τιμες εισοδου και για τις τιμες εξοδου και για τα hidden layers με αποτελεσμα κατά την εκτελεση της παραγωγου για την ελαχιστοποιησει της loss fanction αν οι τιμες των δεδομενων είναι μηδεν(vanish gradient) πολα από τα βαρη να μην ενημερωνωνται με αποτελεσμα να μην είναι αποδοτικη η εκαπιδευση. Την λυση σε αυτό το προβλημα την δινουν οι νευρωνες μακρας βραχειας μνημης lstm.

Τα lstm ανήκουν στην ευρύτερη οικογένεια των rnn νευρωνικων δικτυών . Η βασική διαφορά τους σε σχέση με τα απλά rnn είναι ότι μπορούν να ξεχωρίζουν τα πρόσφατα δεδομένα σε σχέση με τα προγενέστερα δεδομένα δίνοντας στο καθένα διαφορετικά βάρη ενώ ταυτόχρονα μπορούν να διαγράφουν πληροφορικά που δεν την θεωρούνε σημαντικοί για την πρόβλεψή της επόμενης τιμής. Τα lstm κύτταρα έχουν τρεις πύλες οι οποίες διαχειρίζονται την μνήμη του κυττάρου. Αυτές οι τρεις είναι η πύλη εισόδου η πύλη εξόδου και η πύλη forget. Η πρώτη και η τελευταία εκφράζουν την μακροπρόθεσμη μνήμη του κυττάρου ενώ η πύλη εξόδου είναι αυτή που διαχειρίζεται το hidden layer του κυττάρου. Πιο συγκεκριμένα οι πύλες ρυθμίζουν την εισαγωγή νέων δεδομένων στο κύτταρο ή την διαγραφή ήδη υπαρχόντων. Ο τρόπος με τον οποίο

ενεργοποιούνται είναι μέσο της sigmoid συνάρτησης. Αυτό που κάνουν είναι να επιτρέπουν ή όχι την ροή πληροφορίας μέσα στο κύτταρο. Αν το αποτέλεσμα της εκάστοτε πύλης είναι ίσο με ένα τότε το δεόμενο μπαίνει μέσα στο κύτταρο και χωρίς κάτι να το τροποποιήσει. Αν η τιμή είναι ίση με 0 τότε δεν του επιτρέπετε η είσοδος στο κύτταρο, τα δεδομένα είναι είτε προηγούμενες τιμές ητ-1 είτε δεδομένα εισαγωγής χτ στο τρέχον χρονικό βήμα.

Ανάλυση της πύλης forget. Σε αυτή την πύλη γίνεται μια ανάλυση μεταξύ της τιμής της προηγουμένης εσωτερικής κατάστασης ητ-1 σε συνδυασμό με την τιμή εισόδου του τρέχοντος χρονικού διαστήματος χτ. Μέσο αυτής της πύλης το κύτταρο αποφασίζει ποια πληροφορικά από τις δυο είναι πιο σημαντικοί για το μοντέλο και αυτό το δείχνει μέσο μιας σιγμοειδής συνάρτησης που παίρνει τιμές από το 0 έως το 1.

$$ft = sigmoid\,(Wf \cdot [ht - 1,\ xt\,] + bias\ forget)$$

Όσον αφορά την πύλη εισαγωγής αυτή έχει ως είσοδο το ητ-1 και το χτ και χρησιμοποιεί σιγμοειδής συνάρτηση για να αποφασίσει ποιες τιμές πρέπει να ενημερωθούν με νέα βάρη και ποια θα είναι τα βάρη τις εκάστοτε τιμής.

$$it = sigmoid\,(Wi \cdot [ht - 1,\ xt\,] + bias\ input\,)$$

Πριν αναφέρουμε την πύλη εξόδου θα πρέπει να αφαιρούμε μερικές πληροφορίες όσον αφορά την κατάσταση του κυττάρου CT. Σε αυτό πάλι έχουμε ως είσοδο το χτ και το ητ-1 μόνο που αυτή την φορά απλά φιλτράρουμε τα δεδομένα πριν τα βάλουμε στην πύλη εισόδου με μια συνάρτηση tanh.

$$C\,t = \tanh\,(WC \cdot [ht{-}1, xt] + bC)$$

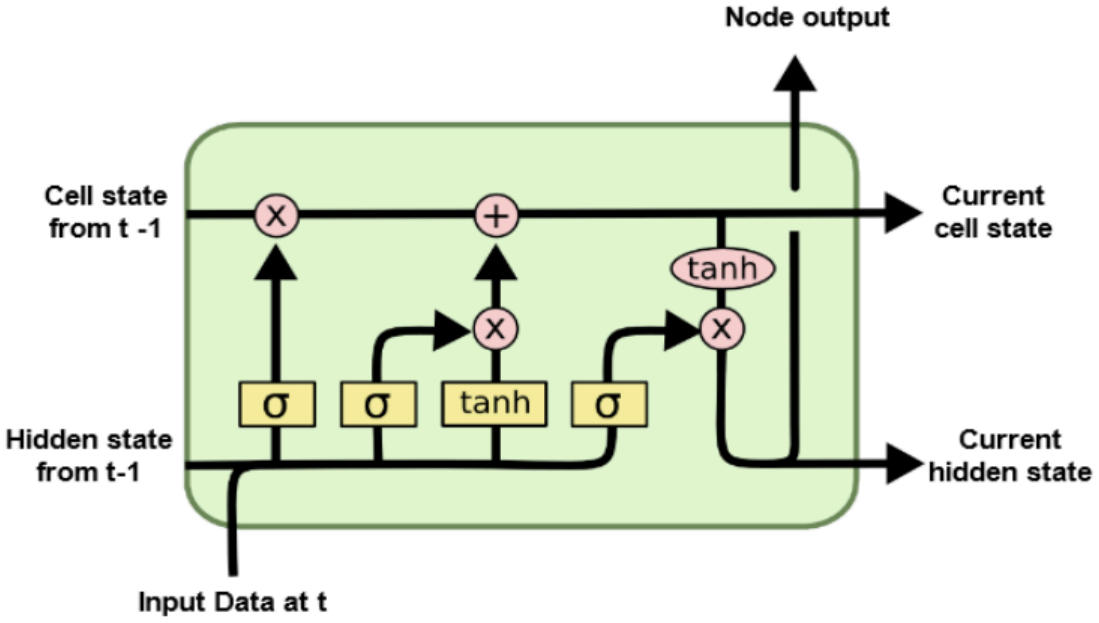
Στην συνείχα με βάση το αποτέλεσμα τις παραπάνω συνάρτησης πρέπει να ενημερωθεί η τωρινή κατάσταση του κυττάρου για να γίνει αυτό πολλαπλασιάζουμε την παλιά κατάσταση με το αποτέλεσμα της πύλης forget δηλαδή τα δεδομένά που είναι προς διαγράφη από το κύτταρό στην συνεχεία πολλαπλασιάζουμε τα αποτελέσματα της tanh συνάρτησης με τα αποτελέσματα της πύλης εισόδου και τέλος προσθέτουμε αυτά τα δυο και έχουμε ως αποτέλεσμα την νέα κατάσταση του κυττάρου.

$$Ct = ft * Ct - 1 + it * C\tau$$

Και τέλος τα αποτελέσματα που θα μπουν στο επόμενο lstm κύτταρό είναι αυτά του CT πολλαπλασιασμένα με αυτά τις πύλης εξόδου οτ

$$ot = \sigma\,(Wo\,[ht - 1,\, xt\,] + bias\, output\,)$$

$$ht = ot * tanh(Ct\,)$$



Εικονα 141: lstm cell

Τέλος θα εφαρμόσουμε ένα απλό νευρωνικο δίκτυο lstm για την πρόβλεψή της κατώτατης τιμής του κινητού τηλεφώνου s24 στην python. Πριν όμως ξεκινήσουμε θα πρέπει να αναφερθούμε σε δυο βασικούς παράγοντες κατά την εκπαίδευση των νευρωνικων δικτυών ο ένας είναι το πλήθος των epochs και ο δεύτερος το batch size. Τα epochs είναι ο αριθμός των φορών που όλο το dataset (εξαρτημένες και ερμηνευτικές μεταβλητές) περνάει μέσα από το νευρωνικο δίκτυο κατά την εκπαίδευση δηλαδή το μοντέλο εξάγει μια πρόβλεψή την συγκρίνει με τις πραγματικές τιμές υπολογίζει το loss function και στην συνεχεία ο αλγόριθμος backpropagation κάνει τις παραγογίσεις και υπολογίζει τα νέα βάρη και το δίκτυο εκπαιδεύεται ξανά με τα νέα βάρη. Αυτή η διαδικασία επαναλαμβάνεται τόσες φορές όσες και ο αριθμός των epochs που έχουν επιλεγεί. Εάν ο αναλυτής επιλέξει μεγάλο αριθμό epochs τότε επειδή η διαδικασία της εκπαίδευσης θα επαναληφθεί πολλές φορές υπάρχει υψηλή πιθανότητα το μοντέλο να κάνει overfit τα δεδομένά , επίσης η εκπαίδευση θα διαρκέσει αρκετή ώρα βέβαια το μοντέλο θα μπορέσει να μάθει πιο σύνθετα μοτίβα της χρονοσειρας δηλαδή θα κατανοήσει καλυτέρα την τάση την κυκλικότητα και την εποχικότητα των δεδομένων. Από την άλλη πλευρά μικρός αριθμός epochs οδηγεί συνηθώς σε underfitting δηλαδή σε όχι και τόσο καλές προβλέψεις καθώς το μοντέλο δεν έχει προλάβει να κατανοήσει τα δεδομένά βέβαια η εκπαίδευση του μοντέλου είναι πιο σύντομη. για να μπορέσει ο αναλυτής να επιλέξει τον βέλτιστο αριθμό epochs μπορεί να παρακολουθήσει το validation loss μετα από κάθε επανάληψή εκπαίδευσης του νευρώνα με νέα βάρη εάν το loss δεν μειώνεται μετα από έναν χ αριθμό επαναλήψεων καλό είναι να σταματήσει την εκπαίδευση. Το batch size είναι ο αριθμός των δεδομένων που το νευρωνικο δίκτυο επεξεργάζεται ταυτόχρονα κατά την εκπαίδευση του, δηλαδή αν επιλεγεί batch size ίσο με ένα τότε τα βάρη θα υπολογιστούν τόσες φορές όσες και ο αριθμός των epochs εάν όμως για παράδειγμά έχουν επιλεγεί 200 epochs και το batch size είναι ίσο με 20 τότε τα βάρη θα υπολογιστούνε 10 φορές. Αν ο αναλυτής θέλει το μοντέλο να αποφεύγει τοπικά ελάχιστα και να είναι πιο σταθερό καλό είναι να επιλέξει μικρό αριθμό epochs εάν όμως δεν έχει τον χρόνο και την υπολογιστική ισχύ για αυτήν την εκπαίδευση μπορεί να αυξήσει τον αριθμό του batch size.

Επίσης μια σημαντικοί διαδικασία πριν την είσοδο της χρονοσειρας στο νευρωνικο δίκτυο είναι η κανονικοποιηση των δεδομένων που την απαρτίζουν. Στον κώδικα που εκτελέσαμε το lstm μοντέλο εφαρμόστηκε η μεθοδολογία κανονικοποιησης minmaxscaler οπού τα δεδομένά περνούν τιμές από 0 έως 1. Ο τύπος εφαρμογής του είναι ο παρακάτω οπού xmax και xmin είναι αντίστοιχα η μικρότερη και η μεγαλύτερη τιμή της χρονοσειρας.

$$minmaxscaler \text{ για ολα τα χι της χρονοσειρας } = \frac{\chi\iota - \chi min}{\chi max - \chi min}$$

Για την εφαρμογή του μοντέλου lstm αρχικά εισάγουμε της γνωστές βιβλιοθήκες όπως την pandas NumPy matplotlib αλλά και την TensorFlow με την όποια μπορούν να εφαρμοστούν τα νευρωνικα δίκτυα. Εν συνεχεία εισάγουμε από την αρχή τα δεδομένά της χρονοσειρας από το φύλο excel και τα μορφοποιούμε σε NumPy array. Έχοντας κάνει τα παραπάνω κανονικοποιουμε τα δεδομένά με την συνάρτηση minmaxscaler και χωρίζουμε τα δεδομένα σε train και test set το οποίο περιλαμβάνει όπως και στα προηγούμενα μοντέλα τις τελευταίες 30 τιμές του κινητού, επίσης ορίζουμε ως T=30 δηλαδή το μοντέλο χρησιμοποιεί τις προηγούμενες 30 τιμές για να προβλέψει την επόμενη και τέλος μορφοποιούμε τα δεδομένά σε τέτοια μορφή ώστε να μπορούν εισέλθουν στο μοντέλο. Αναφορικά με το μοντέλο αυτό αποτελείται από τέσσερα επίπεδα το επίπεδο εισόδου των μεταβλητών το δεύτερο επίπεδο από 128 κύτταρα lstm τα όποια έχουν συνάρτηση ενεργοποίησης την relu ώστε να αποφευχθεί το πρόβλημα του vanishing gradient και έχει οριστεί ως επίπεδο dropout το 0,2 ώστε να αποφευχθεί το overfitting. Το τριπλό επίπεδο αποτελείται από 64 νευρώνες lstm με συνάρτηση ενεργοποίησης την relu και επίπεδο dropout το 0,2. Τέλος το τέταρτο επίπεδο είναι εκείνο της εξόδου του αποτελέσματος με dense=1. Αναφορικά με το κομμάτι της εκπαίδευσης του μοντέλου έχει χρησιμοποιηθεί ως optimizer ο Adam και έχουν οριστεί 200 epochs και το batch size είναι ίσο με 16. Εφόσον ολοκληρωθεί η εκπαίδευση του δικτύου γίνονται οι προβλέψεις για το train set και test set και στην συνεχεία υπολογίζεται το mse μεταξύ των προβλέψεων και των πραγματικών τιμών. Τέλος χρησιμοποιούμε το μοντέλο για να προβλέψουμε τις 30 τιμές εκτός του dataset δηλαδή για τις τιμές του Φεβρουάριου του 2025 στις οποίες εφαρμόζεται μια ποσοστιαία μείωση της τάξης του 2% στην τιμή. Όλα αυτά παρουσιάζονται στο τέλος σε ένα διάγραμμα.

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.preprocessing import MinMaxScaler
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import LSTM, Dense, Dropout
data = pd.read_excel('s24_128gb_price.xlsx')
data['date'] = pd.to_datetime(data['date'])
data.set_index('date', inplace=True)
prices = data['price'].values.reshape(-1, 1)
scaler = MinMaxScaler(feature_range=(0, 1))
prices_scaled = scaler.fit_transform(prices)
T = 30
train_size = len(prices_scaled) - T
train_data = prices_scaled[:train_size]
test_data = prices_scaled[train_size - T:]
def create_sequences(data, T):
    X, Y = [], []
    for i in range(len(data) - T):
        X.append(data[i:i + T])
        Y.append(data[i + T])
    return np.array(X), np.array(Y)
X_train, Y_train = create_sequences(train_data, T)
X_test, Y_test = create_sequences(test_data, T)
model = Sequential([
    LSTM(128, activation='relu', input_shape=(T, 1), return_sequences=True),
    Dropout(0.2),
    LSTM(64, activation='relu'),
    Dropout(0.2),
    Dense(1)
```

```
history = model.fit(X_train, Y_train, epochs=200, batch_size=16, verbose=1, validation_data=(X_test, Y_test))
train_predictions = model.predict(X_train)
test_predictions = model.predict(X_test)
train_predictions = scaler.inverse_transform(train_predictions)
test_predictions = scaler.inverse_transform(test_predictions)
Y_train_actual = scaler.inverse_transform(Y_train)
Y_test_actual = scaler.inverse_transform(Y_test)
future_values = []
last_sequence = prices_scaled[-T:].reshape(1, T, 1)
for _ in range(30):
    next_value = model.predict(last_sequence)
    future_values.append(next_value[0, 0])
    decay_factor = 0.98
    next_value_decreased = next_value * decay_factor
    next_value_scaled = np.append(last_sequence[0, 1:], next_value_decreased).reshape(1, T, 1)
    last_sequence = next_value_scaled
future_values = scaler.inverse_transform(np.array(future_values).reshape(-1, 1))
train_idx = data.index[:train_size]
test_idx = data.index[train_size:]
future_idx = pd.date_range(start=data.index[-1], periods=31, freq='D')[1:]
from sklearn.metrics import mean_squared_error
train_mse = mean_squared_error(Y_train_actual, train_predictions)
test_mse = mean_squared_error(Y_test_actual, test_predictions)
print(f"Train MSE: {train_mse}")
print(f"Test MSE: {test_mse}")
plt.figure(figsize=(14, 7))
plt.plot(data.index, data['price'], label='Actual Price', marker='o')
plt.plot(train_idx[T:], train_predictions.flatten(), label='Train Predictions', linestyle='--')
plt.plot(test_idx, test_predictions.flatten(), label='Test Predictions', linestyle='--')
plt.plot(future_idx, future_values.flatten(), label='Forecast (Next 30 Days)', marker='x', color='red')
plt.legend()
```

```
19/19 ━━━━━━━━━━━━━━━━━━━━ 6s 130ms/step - loss: 0.1257 - val_loss: 0.0119
Epoch 2/200
19/19 ━━━━━━━━━━━━━━━━━━━━ 1s 69ms/step - loss: 0.0204 - val_loss: 0.0243
Epoch 3/200
19/19 ━━━━━━━━━━━━━━━━━━━━ 2s 42ms/step - loss: 0.0182 - val_loss: 0.0195
Epoch 4/200
19/19 ━━━━━━━━━━━━━━━━━━━━ 1s 41ms/step - loss: 0.0181 - val_loss: 0.0124
Epoch 5/200
19/19 ━━━━━━━━━━━━━━━━━━━━ 1s 41ms/step - loss: 0.0160 - val_loss: 0.0085
Epoch 6/200
19/19 ━━━━━━━━━━━━━━━━━━━━ 1s 41ms/step - loss: 0.0129 - val_loss: 0.0072
Epoch 7/200
19/19 ━━━━━━━━━━━━━━━━━━━━ 1s 40ms/step - loss: 0.0143 - val_loss: 0.0094
Epoch 8/200
19/19 ━━━━━━━━━━━━━━━━━━━━ 1s 40ms/step - loss: 0.0109 - val_loss: 0.0093
Epoch 9/200
19/19 ━━━━━━━━━━━━━━━━━━━━ 1s 42ms/step - loss: 0.0114 - val_loss: 0.0079
Epoch 10/200
19/19 ━━━━━━━━━━━━━━━━━━━━ 1s 41ms/step - loss: 0.0121 - val_loss: 0.0112
Epoch 11/200
19/19 ━━━━━━━━━━━━━━━━━━━━ 1s 47ms/step - loss: 0.0116 - val_loss: 0.0090
Epoch 12/200
19/19 ━━━━━━━━━━━━━━━━━━━━ 1s 70ms/step - loss: 0.0166 - val_loss: 0.0108
Epoch 13/200
19/19 ━━━━━━━━━━━━━━━━━━━━ 1s 64ms/step - loss: 0.0097 - val_loss: 0.0113
Epoch 14/200
19/19 ━━━━━━━━━━━━━━━━━━━━ 1s 65ms/step - loss: 0.0092 - val_loss: 0.0088
Epoch 15/200
19/19 ━━━━━━━━━━━━━━━━━━━━ 2s 41ms/step - loss: 0.0100 - val_loss: 0.0090
Epoch 16/200
19/19 ━━━━━━━━━━━━━━━━━━━━ 1s 41ms/step - loss: 0.0112 - val_loss: 0.0073
```

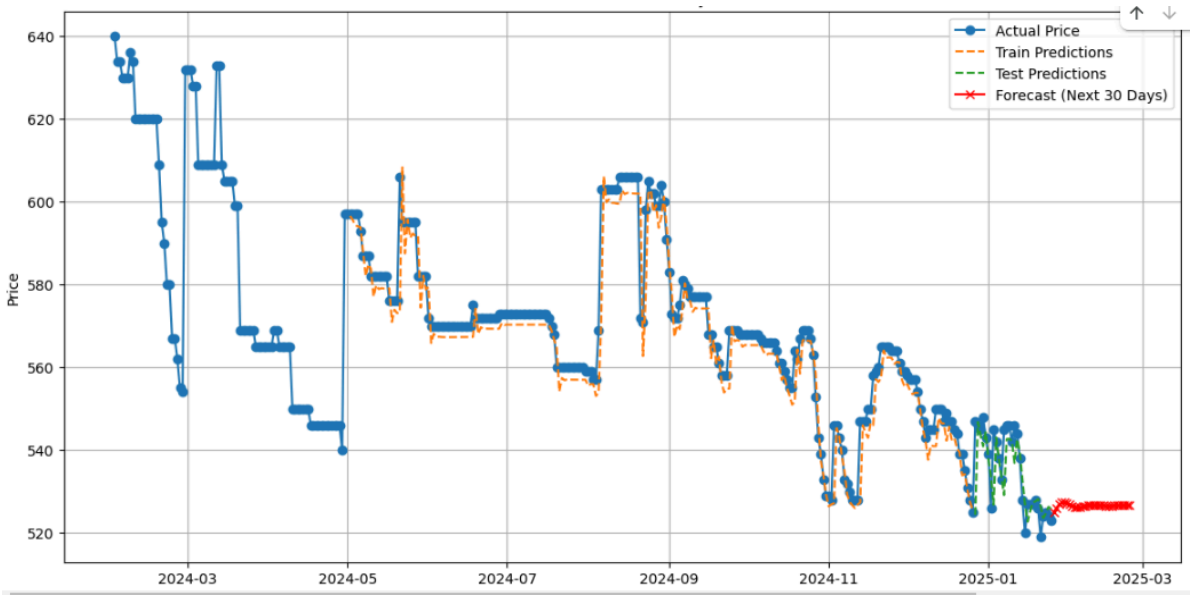
```
Train MSE: 44.08311101004332
Test MSE: 66.13947844207287
```

Εικονες 143: κωδικας και εφαρμογη lstm

Όπως παρατηρούμε το μοντέλο δίνει πολύ καλές προβλέψεις για το test set με mse ίσο με 66 η όποια είναι η δεύτερη καλύτερη επίδοση σε σχέση με όλα τα μοντέλα που εφαρμοστήκαν σε αυτό το κεφάλαιο καθώς βρίσκεται πίσω μόνο από το μοντέλο της γραμμικής παλινδρόμησης. Βέβαια οι προβλέψεις του για την τιμή που θα επικρατήσει τον Φεβρουάριο του 2025 δεν είναι καλές είναι απλά μια ευθεία γραμμή, αυτό οφείλεται και στην εκπαίδευση του μοντέλου ίσως θα έπρεπε να είχαν χρησιμοποιεί λιγότερα epochs με μικρότερο batch size, επίσης μπορεί να οφείλεται και στο γεγονός ότι έχει χρησιμοποιηθεί μικρός αριθμός T δηλαδή η πρόβλεψή της επόμενης τιμής θα έπρεπε να εξαρτιόταν από περισσότερες προγενέστερες τιμές.

Για να λυθεί η παραπάνω απορία έγινε επανεκτελεση του κώδικα αυτή την φορά με T=90 epochs=50 και batch size=1και όπως βλέπουμε το mse του test set μειώθηκε αλλά οι προβλέψεις έκτος του dataset συνεχίζουν να μην είναι οι αναμενόμενες (πτωτική πορεία της τιμής)

```
Train MSE: 32.37484767305115
Test MSE: 49.67345074514548
```

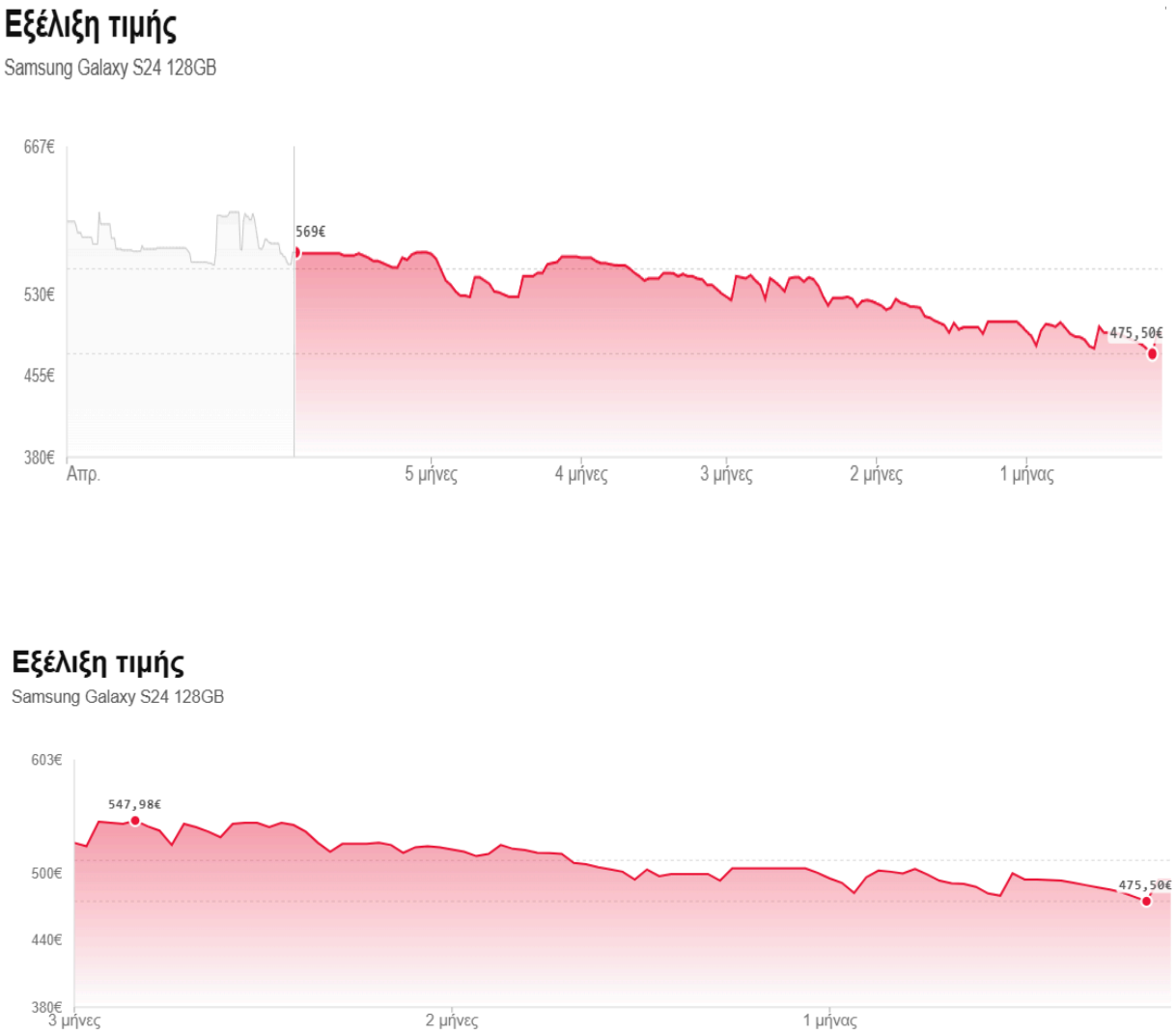


Εικονα 144:δεύτερη εφαρμογή lstm

5.2.14 ΤΙ ΕΓΙΝΕ ΣΤΗΝ ΠΡΑΓΜΑΤΙΚΟΤΗΤΑ

Σε αυτήν την τελευταία ενότητα του κεφαλαίου θα παρατηρήσουμε πως διαμορφώθηκε η τιμή του κινητού τηλεφώνου για τον μηνά Φεβρουάριο του 2025 και θα την συγκρίνουμε με την τιμή που πρόβλεψαν τα μοντέλα. Αρχικά να αναφέρουμε ότι η τιμή για το μηνά Φεβρουάριο είχε πτωτική πορεία όπως ήταν αναμενόμενο εξαιτίας της κυκλοφορίας του νέου μοντέλου s25. Η τιμή που διαμορφώθηκε για την 25/2/2025 ήταν 496 ευρώ. Οπότε το μοντέλο το οποίο πρόβλεψε πολύ καλά αυτήν την πτωτική πορεία της τιμής ανεξαρτήτως της επίδοσης του στο test set (30 ημέρες του Ιανουάριου) ήταν το sarima πάρα το γεγονός ότι ανήκει στην κατηγορία των στατιστικών μοντέλων, μια επίσης καλή πρόβλεψή ήταν αυτή του μοντέλου της γραμμικής παλινδρόμησης ενώ από τα ποιο απλά μοντέλα εκείνο που ξεχωρίζει για τις προβλέψεις του είναι το holt winters. Προφανώς ο ποιο σωστός τρόπος πρόβλεψης των για ένα μηνά μετα το τέλος των δεδομένων είναι να δοθούν όλα τα δεδομένά (και train και test set) στο καλύτερο μοντέλο και να επαναληφθεί η πρόβλεψή καθώς όλα τα παραπάνω μοντέλα προβλέπουν τις τιμές για τις επόμενες 60 ημέρες (30 ημέρες του test set και 30 ημέρες οι προβλέψεις για τον Φεβρουάριο) με αποτέλεσμα να μειώνεται η προβλεπτική τους ικανότητα καθώς για να προβλέψουν τις τιμές του Φεβρουάριου δεν χρησιμοποιούν τα πραγματικά δεδομένά αλλά τις τιμές που τα ιδιά

πρόβλεψαν προηγούμενος. Επίσης όπως αναφέρθηκε και στο κεφάλαιο με το sarima η προσθήκη εξωγενών μεταβλητών στα μοντέλα επίσης θα βοηθούσε την προβλεπτική τους ικανότητα.



Εικονες 144: πραγματική εξέλιξη τιμής s24

6. ΤΙΜΟΛΟΓΙΣΗ ΦΟΡΥΤΩΝ ΥΠΟΛΟΓΙΣΤΩΝ ΜΕ ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ

Στο προηγούμενο κεφάλαιο είδαμε αναλυτικά μεθόδους ανάλυσης χρονολογικών σειρών για την πρόβλεψή των μελλοντικών τιμών ενός κινητού τηλεφώνου , δηλαδή χρησιμοποιήσαμε αποκλειστικά και μόνο προηγούμενες τιμές (σαν ερμηνευτική μεταβλητή) του προϊόντος στην πάροδο του χρόνου. Στο συγκεκριμένο κεφάλαιο θα προσπαθήσουμε να τιμολογήσουμε διάφορους φορητούς υπολογιστές με βάση διαφορά χαρακτηριστικά όπως το μοντέλο του επεξεργαστή ,την μάρκα του λάπτοπ ,το μέγεθος της οθόνης ,το λειτουργικό σύστημα και άλλες ερμηνευτικές μεταβλητές. Έτσι ώστε όταν έρχεται ένα νέο λαπτοπ στο κατάστημα προς πώληση να τιμολογείται άμεσα και με τον αποδοτικότερο τρόπο μέσο μεθόδων μηχανικής μάθησης όπως την γραμμική παλινδρόμηση τα support vector machine και τα δέντρα αποφάσεων. Μέσα από την παραπάνω ανάλυση θα δούμε πια χαρακτηριστικά στοιχεία των λαπτοπ διαδραματίζουν σημαντικό ρολό στην αύξηση της τιμής τους και θα κάνουμε διάφορους στατιστικούς ελέγχους για την επιβεβαίωση τους. Η παραπάνω μεθοδολογία είναι σημαντική στην ανάλυση της ζήτησης καθώς ένας από τους σημαντικότερους παράγοντες αύξησης η μείωσης της ζήτησης είναι η τιμή του προϊόντος και η σωστή τιμολόγηση ειδικά όταν πρόκειται για προϊόντα στα όποια δεν υπάρχουν ιστορικά δεδομένα όπως προηγούμενες τιμές ή δεν υπάρχουν δεδομένα συνολικής ζήτησης ανά ημέρα μηνά κλπ. Προφανώς η μεθοδολογία που θα αναλύσουμε μπορεί να χρησιμοποιηθεί και σε άλλες αγορές όπως την αγορά ακίνητων πχ με βάση ερμηνευτικές μεταβλητές όπως τα τετραγωνικά μετρά την περιοχή που βρίσκεται η κατοικία το όροφο στην πολυκατοικία τον αριθμό των υπνοδωματίων το έτος κατασκευής και αλλά μπορεί να τιμολογηθεί ένα σπίτι και αλλά προϊόντα όπως τα μεταχειρισμένα αυτοκίνητα με βάση την μάρκα την τελική ταχύτητα την κατανάλωση καυσίμου τα τέλη κυκλοφορίας το αν είναι ηλεκτρικό το μέγεθος του την επιτάχυνση κλπ. μπορεί και αυτά να τιμολογηθούν. Επιστρέφοντας στο παράδειγμά μας βρίσκοντας πια είναι εκείνα τα χαρακτηριστικά οπού αυξάνουν την τιμή πώλησης ενός λαπτοπ μπορεί το κατάστημα να επικεντρωθεί στην πώληση των συγκεκριμένων προϊόντων αυξάνοντας με αυτό τον τρόπο το περιθώριο κέρδους. Επίσης εταιρείες κατασκευής λαπτοπ όπως για παράδειγμά η dell και η Lenovo μπορούν να επικεντρωθούν σε εκείνα τα χαρακτηριστικά των λαπτοπ τα όποια προτιμούν οι καταναλωτές με αποτέλεσμα να κάνουν την εφοδιαστική τους αλυσίδα πιο <<lean>> μειώνοντας το εύρος

των κομματιών(αντί να προσφέρεται το ίδιο λαπτοπ με 10 διαφορετικούς επεξεργαστές και 10 διαφορετικές κάρτες γραφικών ,οθόνες , σκληρούς δίσκους κλπ. να προσφέρεται με λιγότερους συνδυασμούς αυτών βάση τις καταναλωτικές προτιμήσεις) που χρησιμοποιούνται στην κατασκευή των λαπτοπ και με αυτό το τρόπο την να μειωθεί η διακρατική μεγάλων ποσοτήτων αποθεμάτων.

Τα δεδομένά για την συγκεκριμένη ανάλυση αποκτήθηκαν όπως και τα δεδομένά του κεφαλαίου 6 από την ιστοσελίδα Kaggle και οποιοσδήποτε θέλει να τα εξερευνήσει παραπάνω ή θέλει να χρησιμοποιήσει άλλες τεχνικές μηχανικής μάθησης όπως η xlboost κλπ. δεν έχει πάρα να πατήσει στον παρακάτω σύνδεσμο για να αποκτήσει πρόσβαση σε αυτά.

https://www.kaggle.com/datasets/owm4096/laptop-prices

Για την ανάλυση των δεδομένων για άλλη μια φορά θα χρησιμοποιηθεί η γλώσσα προγραμματισμού python. Το πρώτο βήμα μετα την λήψη των δεδομένων από το Kaggle και το ανέβασμα αυτών στο google collab κάνουμε import τις σημαντικότερες βιβλιοθήκες όπως έχουμε ήδη δει όπως την pandas την NumPy την matplotlib και την seaborn. Στην συνεχεία χρησιμοποιώντας την pandas εισάγουμε τα δεδομένά σε ένα pd dataframe. Τα δεδομένά αποτελούνται από 23 στήλες και ποιο συγκεκριμένα 22 είναι οι ειρηνευτικές μεταβλητές είτε κατηγορηματικές όπως η κατασκευαστική εταιρεία των λαπτοπ ,η κατηγορία του, κατηγορία της οθόνης κλπ. είτε αριθμητικές όπως το βάρος του λαπτοπ η συχνότητα του επεξεργαστή κλπ. Η εξαρτημένη μεταβλητή δηλαδή η μεταβλητή η όποια θέλουμε να αναλύσουμε και να δούμε πια ή ποιες ερμηνευτικές μεταβλητές την μεταβάλουν κατά κύριο λόγο είναι η τιμή του εκάστοτε λαπτοπ. Τα δεδομένά αποτελούνται από 1275 γραμμές δηλαδή υπάρχουν 23 διαφορετικές μεταβλητές για 1275 διαφορετικά λαπτοπ. Στην συνεχεία ελέγχουμε για το εάν υπάρχουν κενές (null) τιμές μέσα στο dataset.όπως μπορούμε να δούμε δεν υπάρχει κανένα κενό κελί. Εν συνεχεία θα ομαδοποιήσουμε μια μια τις μεταβλητές με βάση το ποσοστό εμφανής της κάθε μιας τιμής στην εκάστοτε στήλη , καθώς υπάρχουν τιμές οπού εμφανίζονται πολύ λίγες φορές μέσα σε αυτές με αποτέλεσμα να κάνουν την μοντελοποιηση πιο δύσκολη καθώς δεν προσφέρουν κάποια σημαντική επιπλέον πληροφορία.

```
[ ] import pandas as pd
    import numpy as np
    import matplotlib.pyplot as plt
    import seaborn as sns
```

```
laptop=pd.read_csv('laptop_prices.csv')
```

```
laptop
```

	Company	Product	TypeName	Inches	Ram	OS	Weight	Price_euros	Screen	ScreenW
0	Apple	MacBook Pro	Ultrabook	13.3	8	macOS	1.37	1339.69	Standard	2560
1	Apple	Macbook Air	Ultrabook	13.3	8	macOS	1.34	898.94	Standard	1440
2	HP	250 G6	Notebook	15.6	8	No OS	1.86	575.00	Full HD	1920
3	Apple	MacBook Pro	Ultrabook	15.4	16	macOS	1.83	2537.45	Standard	2880
4	Apple	MacBook Pro	Ultrabook	13.3	8	macOS	1.37	1803.60	Standard	2560
...	...	...	...	...	...	...	...	...	...	...

1275 rows × 23 columns

```
laptop.columns
```

```
Index(['Company', 'Product', 'TypeName', 'Inches', 'Ram', 'OS', 'Weight',
       'Price_euros', 'Screen', 'ScreenW', 'ScreenH', 'Touchscreen',
       'IPSpanel', 'RetinaDisplay', 'CPU_company', 'CPU_freq', 'CPU_model',
       'PrimaryStorage', 'SecondaryStorage', 'PrimaryStorageType',
       'SecondaryStorageType', 'GPU_company', 'GPU_model'],
      dtype='object')
```

Εικονα 146: εισαγωγή δεδομένων

Πιο αναλυτικά η στήλη company περιλαβαίνει τα ονόματα των κατασκευαστικών εταιρειών των λαπτοπ , η στήλη product είναι κατηγορική μεταβλητή με το μοντέλο του εκάστοτε λαπτοπ ,η στήλη typename δείχνει την κατηγορία του λαπτοπ για παράδειγμά ότι είναι gaming ultrabook κλπ., η στήλη inces είναι αριθμητική μεταβλητή και δείχνει το μέγεθος της οθόνης σε ίντσες , η στήλη ram είναι και αυτή αριθμητική μεταβλητή και περιλαβαίνει το μέγεθος της μνήμης ram του εκάστοτε λαπτοπ, η στήλη os είναι κατηγορική μεταβλητή και δείχνει πιο λειτουργικό σύστημα είναι εγκατεστημένο στο εκάστοτε λαπτοπ, η στήλη weight είναι αριθμητική μεταβλητή και δείχνει το βάρος του λαπτοπ, στην συνεχεία έχουμε την εξαρτημένη και αριθμητική μεταβλητή των δεδομένων την στήλη price_euros που περιλαμβάνει την τιμή του εκάστοτε λαπτοπ, η στήλη screen

είναι κατηγορική μεταβλητή που περιλαβαίνει την κατηγορία της ανάλυσης της οθόνης του λαπτοπ , οι στήλες sreenw sreenh είναι αριθμητικές μεταβλητές και δείχνουν την ανάλυση της οθόνης σε πλάτος και ύψος αντίστοιχα, η στήλη touchscreen είναι κατηγορική μεταβλητή και σηματοδοτεί την ύπαρξη ή όχι οθόνης αφής στο λαπτοπ, εν συνεχεία έχουμε την στήλη ipspanel και retinadisplay που και αυτές εκφράζουν την ύπαρξη η όχι του αντίστοιχου τύπου οθόνης. Η στήλη cpu company είναι κατηγορική μεταβλητή όπως και η στήλη cpu model οπού δείχνουν το μοντέλο και την κατασκευαστική εταιρεία του επεξεργαστή του εκάστοτε λαπτοπ από την άλλη η στήλη cpu_freq είναι αριθμητική μεταβλητή και εκφράζει την συχνότητα του επεξεργαστή σε ghz, η στήλη primarystorage είναι αριθμητική μεταβλητή και δείχνει το αποθηκευτικό χώρο του κάθε λαπτοπ μετρημένο σε gb , η στήλη primarystoragetype είναι κατηγορική μεταβλητή και εκφράζει το τύπο του αποθηκευτικού χώρου δηλαδή εάν το λαπτοπ έχει ssd , hdd ή nvme σκληρό δίσκο , την ιδιά πληροφορία εκφράζουν και οι μεταβλητές secondarystorage και secondarystoragetype αναφορικά με το αν το λαπτοπ έχει και δεύτερο αποθηκευτικό χώρο, και τέλος οι στήλες gpu_company και gpu_model είναι και οι δυο κατηγορικές και περιλαμβάνουν το μοντέλο και την κατασκευαστική εταιρεία της κάρτας γραφικών του εκάστοτε φορητού υπολογιστή.

```
laptop.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1275 entries, 0 to 1274
Data columns (total 23 columns):
 #   Column                Non-Null Count  Dtype  
---  ------                --------------  -----  
 0   Company               1275 non-null   object 
 1   Product               1275 non-null   object 
 2   TypeName              1275 non-null   object 
 3   Inches                1275 non-null   float64
 4   Ram                   1275 non-null   int64  
 5   OS                    1275 non-null   object 
 6   Weight                1275 non-null   float64
 7   Price_euros           1275 non-null   float64
 8   Screen                1275 non-null   object 
 9   ScreenW               1275 non-null   int64  
 10  ScreenH               1275 non-null   int64  
 11  Touchscreen           1275 non-null   object 
 12  IPSpanel              1275 non-null   object 
 13  RetinaDisplay         1275 non-null   object 
 14  CPU_company           1275 non-null   object 
 15  CPU_freq              1275 non-null   float64
 16  CPU_model             1275 non-null   object 
 17  PrimaryStorage        1275 non-null   int64  
 18  SecondaryStorage      1275 non-null   int64  
 19  PrimaryStorageType    1275 non-null   object 
 20  SecondaryStorageType  1275 non-null   object 
 21  GPU_company           1275 non-null   object 
 22  GPU_model             1275 non-null   object 
dtypes: float64(4), int64(5), object(14)
memory usage: 229.2+ KB
```

```
laptop.isna().sum()
```

	0
Company	0
Product	0
TypeName	0
Inches	0
Ram	0
OS	0
Weight	0
Price_euros	0
Screen	0
ScreenW	0
ScreenH	0
Touchscreen	0
IPSpanel	0
RetinaDisplay	0
CPU_company	0
CPU_freq	0

```
laptop.describe().T
```

	count	mean	std	min	25%	50%	75%	max
Inches	1275.0	15.022902	1.429470	10.10	14.0	15.60	15.60	18.4
Ram	1275.0	8.440784	5.097809	2.00	4.0	8.00	8.00	64.0
Weight	1275.0	2.040525	0.669196	0.69	1.5	2.04	2.31	4.7
Price_euros	1275.0	1134.969059	700.752504	174.00	609.0	989.00	1496.50	6099.0
ScreenW	1275.0	1900.043922	493.346186	1366.00	1920.0	1920.00	1920.00	3840.0
ScreenH	1275.0	1073.904314	283.883940	768.00	1080.0	1080.00	1080.00	2160.0
CPU_freq	1275.0	2.302980	0.503846	0.90	2.0	2.50	2.70	3.6
PrimaryStorage	1275.0	444.517647	365.537726	8.00	256.0	256.00	512.00	2048.0
SecondaryStorage	1275.0	176.069020	415.960655	0.00	0.0	0.00	0.00	2048.0

Εικονα 146: ανάλυση dataset

Στην συνεχεία παρουσιάζεται ένας πίνακας με τιμές όπως ο μέσος όρος η μέγιστη , η ελάχιστη τιμή ,η τυπική απόκλιση και άλλες τιμές , η τιμή του ¼ τεταρτημόριου της κατανομής της εκάστοτε μεταβλητής για όλες τις αριθμητικές μεταβλητές του dataset.

Έχοντας αναλύσει τα παραπάνω θα συνεχίσουμε με αυτό που αναφέραμε πιο πάνω την ομαδοποίηση των τιμών σε κάθε ποιοτική μεταβλητή οπού εμφανίζεται λίγες φορές μέσα σε αυτή (είναι ένας τύπος <<καθαρισμού>> των δεδομένων). Η πρώτη μεταβλητή που θα

ομαδοποιήσουμε(όταν λεμέ για ομαδοποίηση δεν αναφερόμαστε στην τεχνική της μηχανικής μάθησης αυτού του είδους ομαδοποίηση ονομάζεται clustering) τις τιμές είναι η κατασκευαστική εταιρεία των λαπτοπ. αρχικά βλέπουμε το ποσοστό εμφάνισης της κάθε εταιρείας με τον παρακάτω κώδικα.

```
percentages = laptop.Company.value_counts()/laptop.shape[0]*100
percentages
```

Company	count
Dell	22.823529
Lenovo	22.666667
HP	21.019608
Asus	11.921569
Acer	7.921569
MSI	4.235294
Toshiba	3.764706
Apple	1.647059
Samsung	0.705882
Razer	0.549020
Mediacom	0.549020
Microsoft	0.470588
Xiaomi	0.313725
Vero	0.313725
Chuwi	0.235294

Στην συνεχεία αντικαθιστούμε και ομαδοποιούμε όλες εκείνες της εταιρίες οπού έχουν ποσοστό εμφάνισης μικρότερο του 0,5%. Δηλαδή εννοούμε τις εταιρείες Microsoft Xiaomi Vero και chuwi με την τιμή others.

```
def replace_company_name(company):
    if percentages[company]>=0.5:
        return company
    else:
        return 'Others'
laptop['Company'] = laptop['Company'].apply(replace_company_name)
```

Εικονα 147: μετατροπή στήλης company

Συνεχίζουμε με την μεταβλητή TypeName οπού σε αυτήν ομαδοποιούμε τους τύπους λαπτοπ με ποσοστό εμφάνισης μικρότερος ή ίσο του 2,5% δηλαδή ενώνουμε τις τιμές workstation και netbook στην τιμή others

```
percentages = laptop.TypeName.value_counts()/laptop.shape[0]*100
percentages
```

TypeName	count
Notebook	55.450980
Gaming	16.078431
Ultrabook	15.215686
2 in 1 Convertible	9.176471
Workstation	2.274510
Netbook	1.803922

```
def replace_typename(typename):
    if percentages[typename]>=2.5:
        return typename
    else:
        return 'Others'
laptop['TypeName'] = laptop['TypeName'].apply(replace_typename)
```

```
newpercentages = laptop.TypeName.value_counts()/laptop.shape[0]*100
newpercentages
```

TypeName	count
Notebook	55.450980
Gaming	16.078431
Ultrabook	15.215686
2 in 1 Convertible	9.176471
Others	4.078431

dtype: float64

Εικονα 148: μετατροπή στήλης typename

Συνεχίζουμε με την μεταβλητή os δηλαδή το λειτουργικό σύστημα του εκάστοτε λαπτοπ, αυτήν την μεταβλητή δεν την ομαδοποιούμε με βάση το ποσοστό εμφάνισης της κάθε τιμής αλλά αντίθετος την ομαδοποιούμε με βάση την εταιρεία κατασκευής του

λειτουργικού συστήματος δηλαδή είτε το λειτουργικό είναι τα windows 7 , windows 10 κλπ. για την ανάλυση μας θα ομαδοποιηθούν στην τιμή απλά windows καθώς θεωρούμε ότι δεν προσφέρει μεγάλη διαφορά στον προσδιορισμό της τιμής του λαπτοπ αντίθετος αυξάνει την πολυπλοκότητα στους αλγορίθμους μηχανικής μάθησης (προφανώς στην επιστήμη των υπολογιστών και στην μηχανική μάθηση δεν υπάρχει κάτι σωστό ή λάθος οπότε ο κάθε αναλυτής είναι ελεύθερος να πειραματιστεί με τα δεδομένα με οποίον τρόπο επιθυμεί εκείνος για παράδειγμά θα μπορούσε να χρησιμοποιήσει την μέθοδο με τα ποσοστά εμφάνισης ή απλά να μην ομαδοποιήσει καθόλου τα δεδομένα). Το ίδιο κάνουμε και για το λειτουργικό σύστημα των mac φορητών υπολογιστών.

```
percentages = laptop.OS.value_counts()/laptop.shape[0]*100
percentages
```

OS	count
Windows 10	82.196078
No OS	5.176471
Linux	4.549020
Windows 7	3.529412
Chrome OS	2.117647
macOS	1.019608
Mac OS X	0.627451
Windows 10 S	0.627451
Android	0.156863

```
def replace_os_name(os_name):
    if os_name.lower().startswith('windows'):
        return 'Windows'
    elif os_name.lower().startswith('mac'):
        return 'Mac'
    else:
        return os_name
laptop['OS'] = laptop['OS'].apply(replace_os_name)
percentagesnew = laptop.OS.value_counts()/laptop.shape[0]*100
percentagesnew
```

OS	count
Windows	86.352941
No OS	5.176471
Linux	4.549020
Chrome OS	2.117647
Mac	1.647059
Android	0.156863

dtype: float64

Εικονα 149: μετατροπή στήλης os

Την μέθοδο με τα ποσοστά για την ομαδοποίηση των δεδομένων χρησιμοποιούμε ξανά στην μεταβλητή με τα μοντέλα των λαπτοπ οπού όπως μπορούμε να παρατηρήσουμε τα δεδομένα μας περιλαμβάνουν 93 διαφορετικούς επεξεργαστές οπότε σε αυτή την μεταβλητή είναι καλή στρατηγική η ομαδοποίηση των τιμών με βάση το ποσοστό εμφάνισης μικρότερο ίσου του 1%.

```
percentages=laptop.CPU_model.value_counts()/laptop.shape[0]*100
percentages
```

CPU_model	count
Core i5 7200U	15.137255
Core i7 7700HQ	11.529412
Core i7 7500U	10.431373
Core i3 6006U	6.352941
Core i7 8550U	5.725490
...	...
Core M m3	0.078431
E-Series E2-9000	0.078431
Core M M3-6Y30	0.078431
A6-Series 7310	0.078431
A9-Series 9410	0.078431

93 rows × 1 columns

dtype: float64

```
def replace_cpu_model(cpu_model):
    if percentages[cpu_model]>=1:
        return cpu_model
    else:
        return 'Others'
laptop['CPU_model'] = laptop['CPU_model'].apply(replace_cpu_model)
newpercentages = laptop.CPU_model.value_counts()/laptop.shape[0]*100
newpercentages
```

CPU_model	count
Others	18.745098
Core i5 7200U	15.137255
Core i7 7700HQ	11.529412
Core i7 7500U	10.431373
Core i3 6006U	6.352941
Core i7 8550U	5.725490
Core i5 8250U	5.647059
Core i5 6200U	5.333333
Core i7 6500U	3.372549
Core i7 6700HQ	3.215686
Core i3 7100U	2.745098

Εικονα 150: μετατροπή στήλης cpu model

Στην συνεχεία ομαδοποιούμε την κατηγορία του δευτέρου μέσου αποθήκευσης των λαπτοπ και πιο συγκεκριμένα ενώνουμε την τιμή SSD με ποσοστό εμφάνισης 0,31% και της τιμής hybrid με ποσοστό 0,15% στην τιμή others.

```
percentages=laptop.SecondaryStorageType.value_counts()/laptop.shape[0]*100
percentages
```

SecondaryStorageType	count
No	83.686275
HDD	15.843137
SSD	0.313725
Hybrid	0.156863

dtype: float64

```
def replace_secondary_storage_type(secondary_storage_type):
    if percentages[secondary_storage_type]>=10:
        return secondary_storage_type
    else:
        return 'Others'
laptop['SecondaryStorageType'] = laptop['SecondaryStorageType'].apply(replace_secondary_storage_type)
newpercentages = laptop.SecondaryStorageType.value_counts()/laptop.shape[0]*100
newpercentages
```

Εικονα 151: μετατροπή στήλης secondarystoragetype

Με την ιδιά μεθοδολογία ομαδοποιούμε και την μεταβλητή που περιέχει το μοντέλο του εκάστοτε λαπτοπ , που όπως παρατηρούμε υπάρχουν 110 διαφορετικά μοντέλα καρτών γραφικών. Η ομαδοποίηση του μοντέλου γίνεται στην περίπτωση οπού το ποσοστό εμφάνισης ενός συγκείμενου μοντέλου είναι μικρότερο ή ίσο του 1%.

```
percentages=laptop.GPU_model.value_counts()/laptop.shape[0]*100
percentages
```

GPU_model	count
HD Graphics 620	21.882353
HD Graphics 520	14.196078
UHD Graphics 620	5.333333
GeForce GTX 1050	5.176471
GeForce GTX 1060	3.764706
...	...
Radeon R5 520	0.078431
Radeon R7	0.078431
HD Graphics 540	0.078431
Radeon 540	0.078431
Mali T860 MP4	0.078431

110 rows × 1 columns

dtype: float64

```
def replace_gpu_model(gpu_model):
    if percentages[gpu_model]>=1:
        return gpu_model
    else:
        return 'Others'
laptop['GPU_model'] = laptop['GPU_model'].apply(replace_gpu_model)
newpercentages=laptop.GPU_model.value_counts()/laptop.shape[0]*100
newpercentages
```

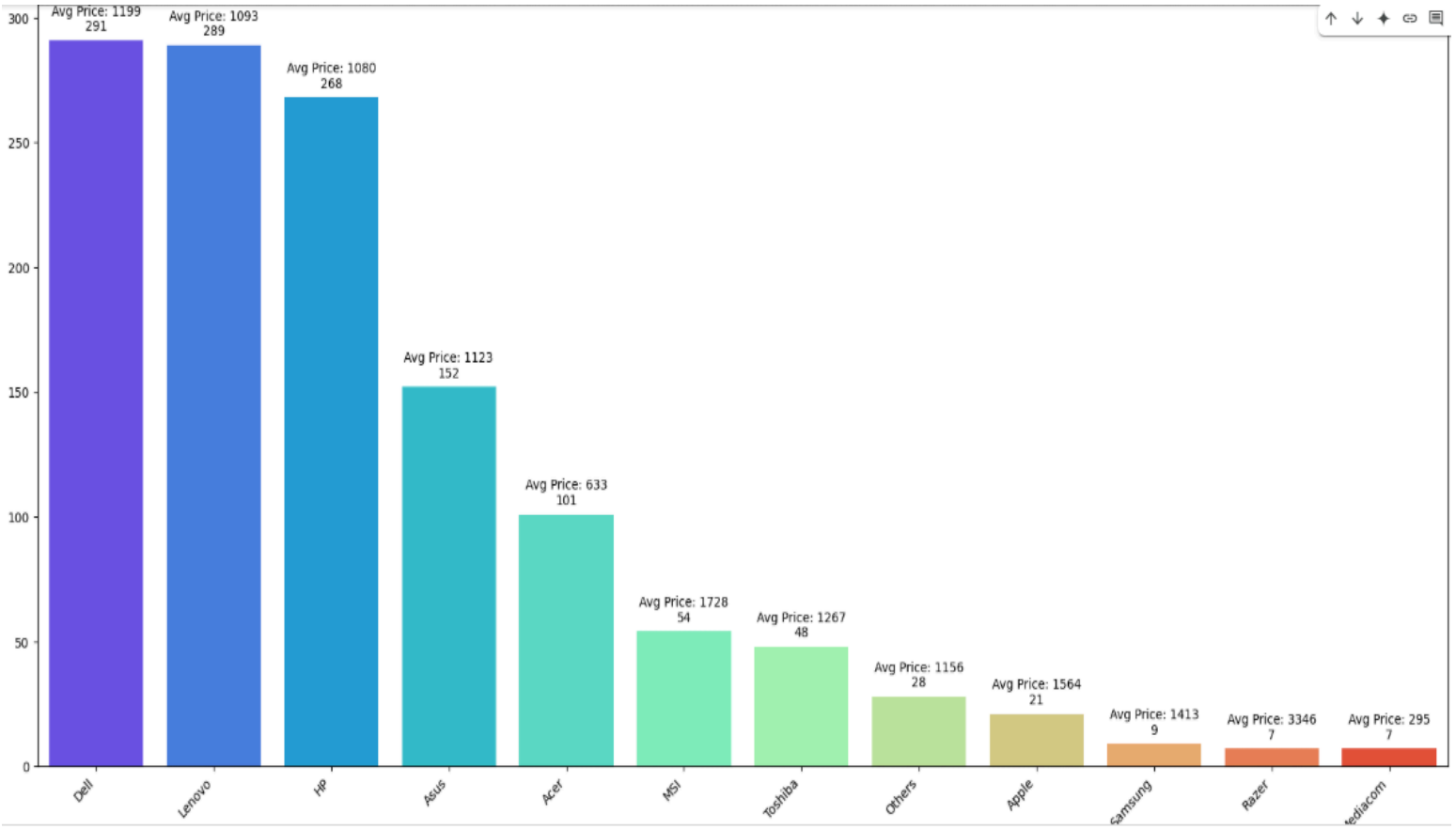
GPU_model	count
HD Graphics 620	21.882353
Others	21.176471
HD Graphics 520	14.196078
UHD Graphics 620	5.333333
GeForce GTX 1050	5.176471
GeForce GTX 1060	3.764706
GeForce 940MX	3.372549
Radeon 530	3.215686
HD Graphics 500	3.058824
HD Graphics 400	2.588235

Εικονα 152: μετατροπή στήλης gpumodel

Με την παραπάνω ομαδοποίηση ολοκληρώνεται ο καθαρισμός των δεδομένων. Οπότε συνεχίζουμε με την γραφική αναπαράσταση των κατηγορικών μεταβλητών με βάση την τιμή πώλησης των λαπτοπ. Ο κώδικας είναι ίδιος για όλες τις κατηγορηματικές μεταβλητές που ακολουθούν το μόνο που αλλάζει είναι το όνομα της στήλης του dataset που θέλουμε

η βιβλιοθήκη seaborn να εμφανίσει .Οι στήλες του διαγράμματος δείχνουν τον αριθμό των παρατηρήσεων της κάθε διαφορετικής τιμής της μεταβλητής company(όσον αφορά τον παρακάτω κώδικα) ταυτόχρονα υπολογίζεται και εμφανίζεται στο διάγραμμα η μέση τιμή των λαπτοπ της εκάστοτε κατασκευάστριας εταιρείας.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
plt.figure(figsize=(18, 9))
splot = sns.countplot(x='Company', data=laptop, order=laptop['Company'].value_counts().index,palette='rainbow')

for p in splot.patches:
    splot.annotate(format(p.get_height(), '.0f'),
                   (p.get_x() + p.get_width() / 2., p.get_height()),
                   ha='center', va='center', xytext=(0, 10),
                   textcoords='offset points')

for company in laptop['Company'].unique():
    avg_price = laptop[laptop['Company'] == company]['Price_euros'].mean()
    plt.text(laptop['Company'].value_counts().index.get_loc(company),
             laptop['Company'].value_counts()[company] + 10,
             f"Avg Price: {int(avg_price)}", ha='center')

plt.xticks(rotation=45, ha='right')
plt.tight_layout()
plt.show()
```



Εικονα 153: countplot company

Στην συνεχεία δημιουργούμε ένα pivot table για την μεταβλητή company οπού για κάθε διαφορετική τιμή (unique value) της μεταβλητής υπολογίζεται η μέγιστη η ελάχιστη και η μέση τιμή της κάθε μιας σε συνδυασμό με τον αριθμό επανεμφάνισης της (count).

```
pivot_table = pd.pivot_table(laptop, values='Price_euros', index='Company',
                             aggfunc=[np.max, np.mean, np.min, 'count'])
pivot_table.columns = ['Max Price', 'Average Price', 'Min Price', 'Count']
pivot_table = pivot_table.sort_values(by='Average Price', ascending=False)
pivot_table = pivot_table.round(1)
pivot_table
```

Company	Max Price	Average Price	Min Price	Count
Razer	6099.0	3346.1	1029.0	7
MSI	2799.0	1728.9	839.0	54
Apple	2858.0	1564.2	898.9	21
Samsung	1849.0	1413.4	269.0	9
Toshiba	2799.0	1267.8	447.0	48
Dell	3659.4	1199.2	274.9	291
Others	2589.0	1156.6	196.0	28
Asus	3975.0	1123.8	191.9	152
Lenovo	4899.0	1093.9	229.0	289
HP	4389.0	1080.3	209.0	268
Acer	2599.0	633.5	174.0	101
Mediacom	389.0	295.0	239.0	7

Εικονα 154: pivot table company

Όπως μπορούμε να δούμε τα πιο ακριβά όσον αφορά την μέγιστη και την μέση τιμή λαπτοπ είναι της εταιρείας razer που κατασκευάζει κατά κύριο λόγο gaming(υπολογιστές οπού η κυριά χρήση τους είναι για την εκτέλεση βιντεοπαιχνιδιών) φορητούς υπολογιστές υψηλών επιδόσεων. Στην δεύτερη θέση με βάση την μέση τιμή των υπολογιστών βρίσκεται επίσης μια εταιρεία που κατασκευάζει gaming υπολογιστές η msi , στην συνεχεία ακολουθούν εταιρείες όπως η apple με τους mac η Samsung και άλλες με αρκετά χαμηλότερη μέση τιμή όπως η dell και η Lenovo , παρόλα αυτά και αυτές οι εταιρείες κατασκευάζουν κάποια πολύ premium λαπτοπ με αντίστοιχα πολύ υψηλές τιμές αυτό φαίνεται στην στήλη με τις μέγιστες τιμές.

Στην συνεχεία ακολουθεί ένας έλεγχος υποθέσεων συγκεκριμένα είναι ένας μονόπλευρος έλεγχος στον οποίο θέλουμε να δούμε εάν τα λαπτοπ της dell είναι στατιστικά ακριβότερα σε επίπεδο σημαντικότητας 95% (α=0,05) σε σχέση με αυτά της apple. Στον έλεγχο

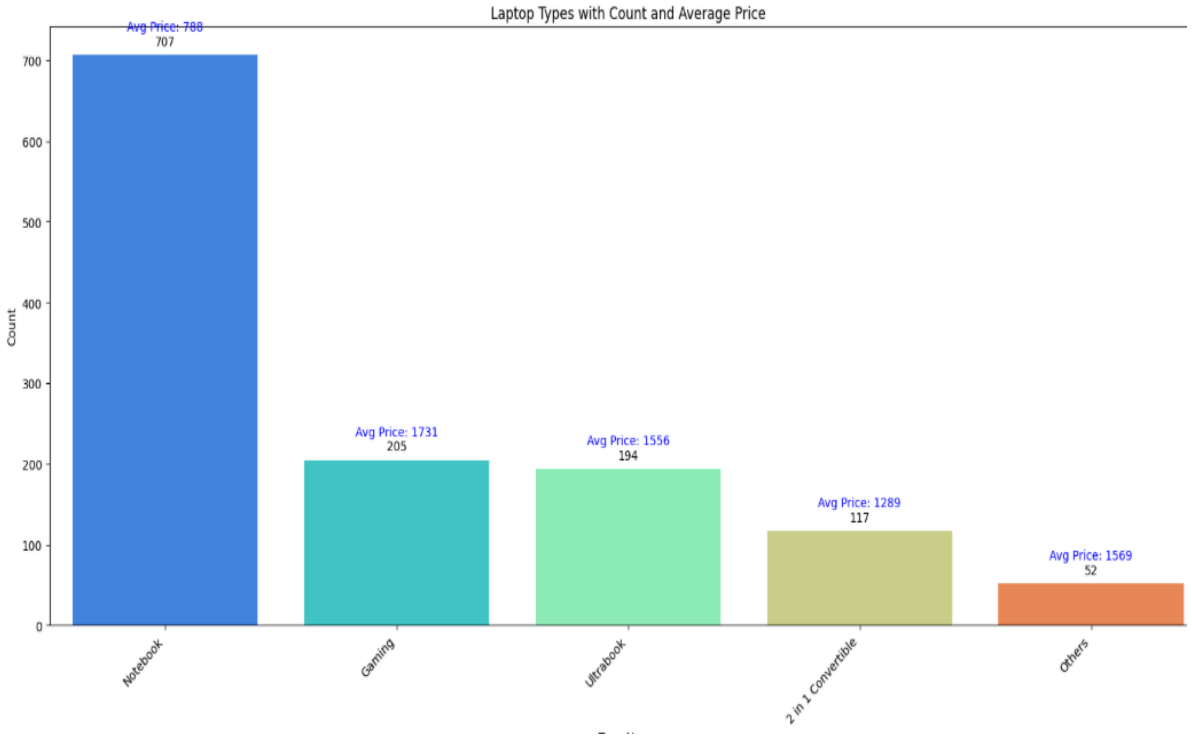
θεωρούμε ότι οι διακυμάνσεις των δειγμάτων είναι άνισες και όπως αποδεικνύεται με βάση τ στατιστικό δεν μπορούμε να δεχτούμε αυτή την υπόθεση και αρά τα dell λαπτοπ δεν είναι ακριβότερα από τα mac.

```
from scipy.stats import ttest_ind, t
dell = laptop[laptop['Company'] == 'Dell']['Price_euros']
apple = laptop[laptop['Company'] == 'Apple']['Price_euros']
t_statistic, p_value_two_tailed = ttest_ind(dell, apple, equal_var=False)
n1 = len(dell)
n2 = len(apple)
s1 = dell.var(ddof=1)
s2 = len(apple)
df = ((s1 / n1 + s2 / n2) ** 2) / (((s1 / n1) ** 2) / (n1 - 1) + ((s2 / n2) ** 2) / (n2 - 1))
alpha = 0.05
t_critical = t.ppf(1 - alpha, df)
print(f"T-statistic: {t_statistic}")
print(f"Critical t-value (one-tailed, alpha={alpha}): {t_critical}")
if t_statistic > t_critical:
    print("Reject the null hypothesis.")
    print("Dell laptops are priced significantly higher than apple laptops.")
else:
    print("Fail to reject the null hypothesis.")
    print("There is no statistically significant evidence that Dell laptops are priced higher than apple laptops.")
```

```
T-statistic: -2.834755115998222
Critical t-value (one-tailed, alpha=0.05): 1.6501181734166772
Fail to reject the null hypothesis.
There is no statistically significant evidence that Dell laptops are priced higher than apple laptops.
```

Εικονα 155: hypothesis testing dell vs apple

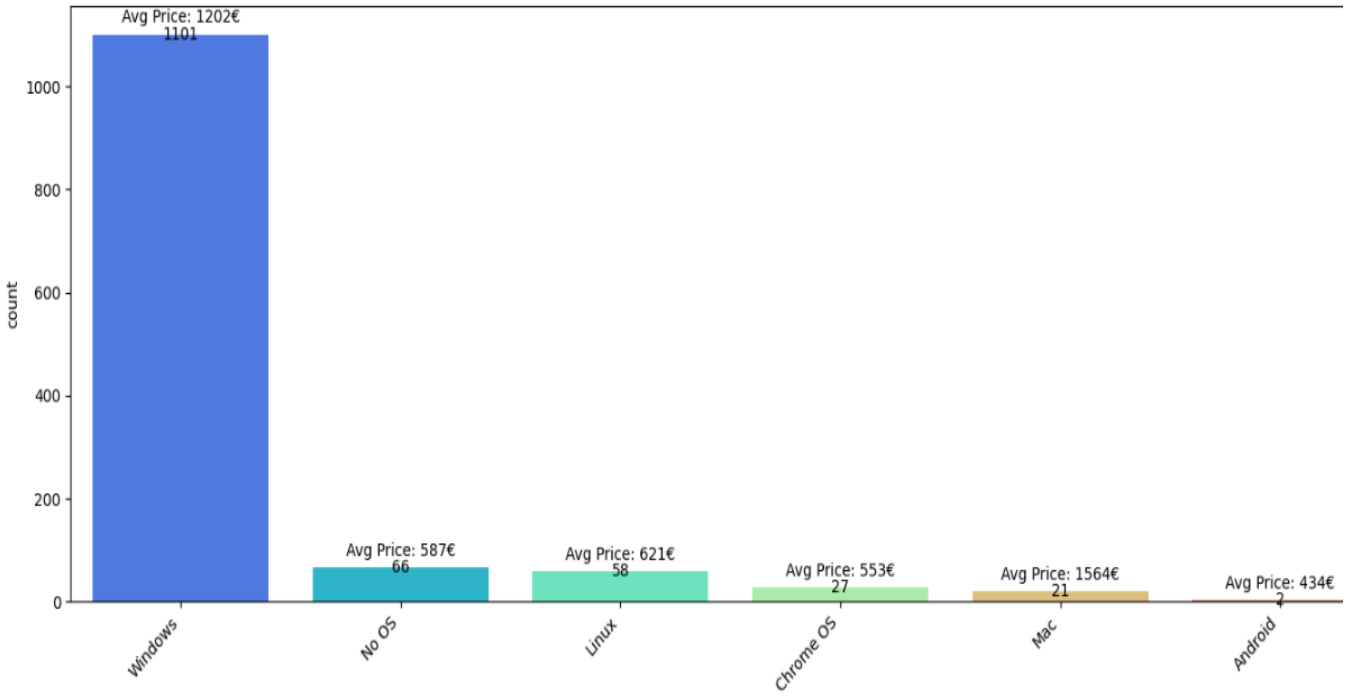
Συνεχίζουμε την ανάλυση μας με την μεταβλητή typename δηλαδή την κατηγορία του κάθε λαπτοπ. Όπως μπορούμε να δούμε τα πιο ακριβά λαπτοπ κατά μέσο ορό είναι τα gaming καθώς για να μπορούν να εκτελέσουν απαιτητικά προγράμματα όπως τα βιντεοπαιχνίδια απαιτείται υψηλή υπολογιστική ισχύ και από τον επεξεργαστή και από την κάρτα γραφικών ενώ ταυτόχρονα απαιτείται οθόνη υψηλού ρυθμού ανανέωσης για τα ανταγωνιστικά(competitive) esports βιντεοπαιχνίδια επίσης το λαπτοπ θα πρέπει να έχει καλό σύστημα ψύξης ώστε ο επεξεργαστής και η κάρτα γραφικών του να μην υπερθερμαίνεται, αποτέλεσμα των παραπάνω είναι να αυξάνεται το κόστος κατασκευής και κατά επέκταση η τελική τιμή τους. Επίσης αρκετά υψηλή μέση τιμή έχουν και τα ultrabooks δηλαδή τα λαπτοπ τα όποια τα χαρακτηρίζει το μικρό τους βάρος οι υψηλές αποδόσεις σε λειτουργίες καθημερινής χρήσης και η μεγάλη διάρκεια της μπαταρίας τους τα όποια τα καταστεί ιδανικά για επαγγελματίες οι οποίοι ταξιδεύουν συχνά και δεν έχουν κάποιο σταθερό χώρο εργασίας, και τέλος τα πιο φθηνά με βάση την μέση τιμή είναι τα notebooks τα όποια θεωρούνται τα πιο value for money laptop καθώς είναι ιδανικά για έναν μέσο χρήστη και ταυτόχρονα έχουν σχετικά χαμηλή τιμή. Στην συνεχεία ακολουθεί το διάγραμμα και το αντίστοιχο pivot table.



TypeName	Max Price	Average Price	Min Price	Count
Gaming	6099.0	1731.4	699.0	205
Others	4389.0	1569.6	174.0	52
Ultrabook	3100.0	1556.7	499.0	194
2 in 1 Convertible	2824.0	1289.7	275.0	117

Εικονα 153: countplot , pivot table typename

Συνεχίζουμε με την ανάλυση του λειτουργικού συστήματος όπως μπορούμε να παρατηρήσουμε αν εξαιρέσουμε μερικές ακραίες τιμές gaming λαπτοπ με λειτουργικό windows την υψηλότερη μέση τιμή έχουν τα λαπτοπ της apple , εν συνεχεία με μεγάλη διακύμανση στην τιμή ακολουθούν τα λαπτοπ με windows. Επίσης υπάρχουν 58 λαπτοπ με Linux δηλαδή ανοιχτού κώδικα λειτουργικό σύστημα και στην συνεχεία ακολουθούν λαπτοπ χωρίς λειτουργικό(no os) που αυτό επιτρέπει στα καταστήματα λιανικής πώλησης να τα πουλήσουν σε χαμηλότερες τιμές και να επωμιστεί στην συνεχεία το κόστος αγοράς του λειτουργικού ο καταναλωτής. Στην συνεχεία ακολουθούν το διάγραμμα και το pivot table.



OS	Max Price	Average Price	Min Price	Count
Mac	2858.0	1564.2	898.9	21
Windows	6099.0	1202.1	191.9	1101
Linux	1099.0	621.9	224.0	58
No OS	1400.0	588.0	252.4	66
Chrome OS	2199.0	553.6	174.0	27
Android	549.0	434.0	319.0	2

Εικονα 154: countplot , pivot table os

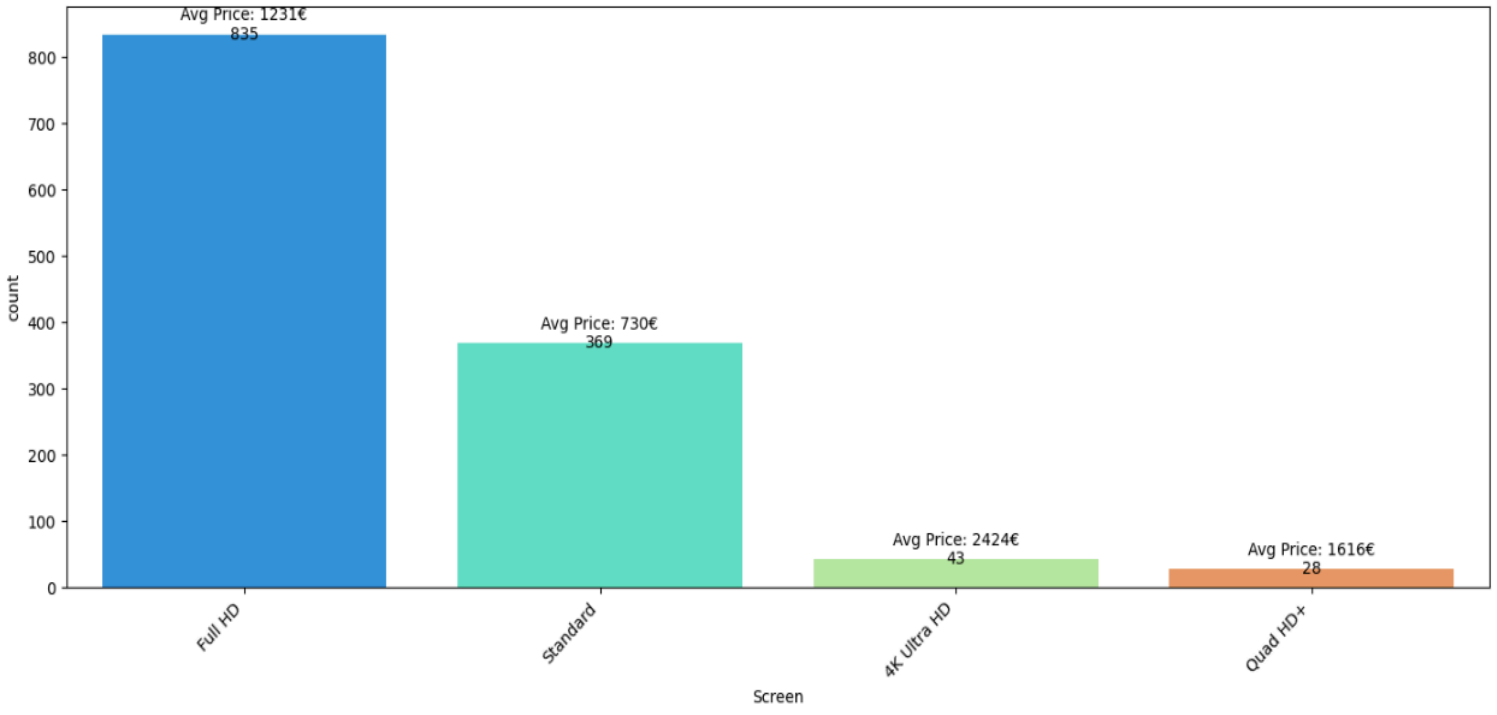
Ακολουθεί ένας ακόμα μονόπλευρος έλεγχος υποθέσεων στον οποίο ελέγχουμε εάν οι τιμές των υπολογιστών με mac λειτουργικό σύστημα είναι χαμηλότερες σε σχέση με εκείνες των φορητών υπολογιστών με windows , ο έλεγχος γίνεται με α=0,05 οι διακυμάνσεις των δειγμάτων θεωρούμε ότι είναι άνισες και το αποτέλεσμα με βάση το τ στατιστικό είναι ότι ο αναλυτής πρέπει να απορρίψει την μηδενική υπόθεση και να δεχτεί την η1 δηλαδή ότι οι υπολογιστές mac είναι στατιστικά ακριβότερή σε σχέση με εκείνους με windows.

```
from scipy.stats import ttest_ind, t
mac = laptop[laptop['OS'] == 'Mac']['Price_euros']
windows = laptop[laptop['OS'] == 'Windows']['Price_euros']
t_statistic, p_value_two_tailed = ttest_ind(mac, windows, equal_var=False)
n1 = len(mac)
n2 = len(windows)
s1 = mac.var(ddof=1)
s2 = windows.var(ddof=1)
df = ((s1 / n1 + s2 / n2) ** 2) / (((s1 / n1) ** 2) / (n1 - 1) + ((s2 / n2) ** 2) / (n2 - 1))
alpha = 0.05
t_critical = t.ppf(1 - alpha, df)
print(f"T-statistic: {t_statistic}")
print(f"Critical t-value (one-tailed, alpha={alpha}): {t_critical}")
if t_statistic > t_critical:
    print("Reject the null hypothesis.")
    print("Laptops with mac os  are priced significantly higher than those with windows.")
else:
    print("Fail to reject the null hypothesis.")
    print("There is no statistically significant evidence that laptops with mac os are priced higher than those with windows.")
```

```
T-statistic: 2.910533811570186
Critical t-value (one-tailed, alpha=0.05): 1.7198857048107121
Reject the null hypothesis.
Laptops with mac os  are priced significantly higher than those with windows.
```

Εικονα 155: hypothesis testing mac vs windows

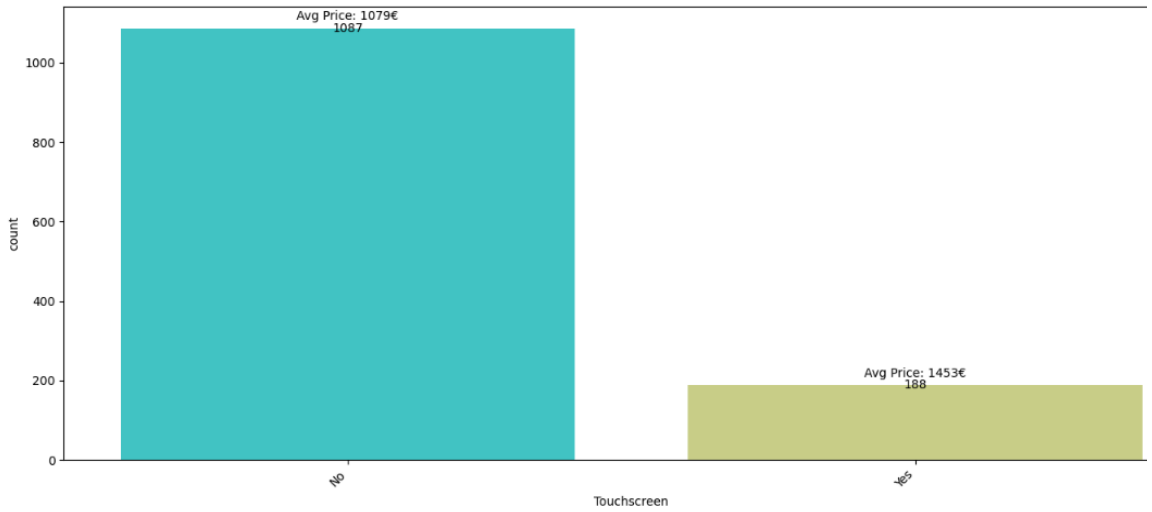
Συνεχίζουμε την ανάλυση της τιμής των λαπτοπ με την μεταβλητή της κατηγορίας της οθόνης που αυτά φέρουν. Τα αποτελέσματα είναι ξεκάθαρα όσο πιο μεγάλη είναι η ανάλυση της οθόνης τόσο μεγαλύτερη είναι και η μέση τιμή τους. Για αυτό το λόγο τα λαπτοπ με 4κ ανάλυση είναι ακριβότερα κατά μέσο ορό από εκείνα με qhd ανάλυση το ίδιο ισχύει και για εκείνα με fullhd ανάλυση σε σύγκριση με τα standard (720p).



Screen	Max Price	Average Price	Min Price	Count
4K Ultra HD	6099.0	2424.8	1029.0	43
Quad HD+	2680.0	1617.0	615.0	28
Full HD	4389.0	1231.2	196.0	835
Standard	2858.0	730.3	174.0	369

Εικονα 156: countplot , pivot table screen

Το ίδιο ισχύει και για τα λαπτοπ οπού έχουν οθόνη αφής όπως φαίνεται η οθόνη αφής είναι ένα χαρακτηριστικό οπού αυξάνει την τιμή των λαπτοπ , και είναι και κάτι σχετικά μη σύνηθες στα λαπτοπ τουλάχιστον σε αυτά που περιλαμβάνονται στο dataset καθώς μόνο 188 από τα 1275 έχουν οθόνη αφής.

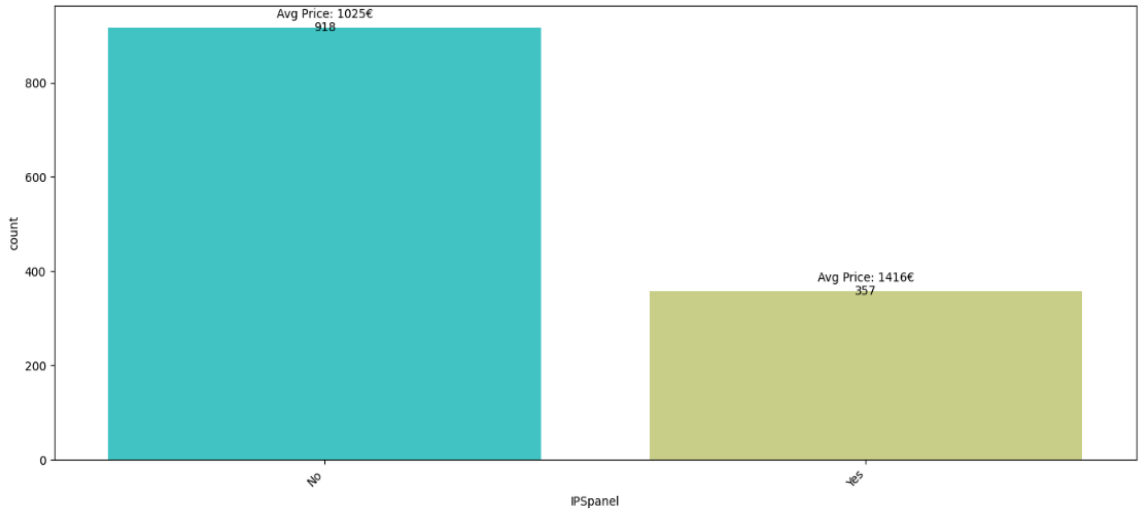


Touchscreen	Max Price	Average Price	Min Price	Count
Yes	6099.0	1453.1	275.0	188
No	4899.0	1079.9	174.0	1087

Εικονα 156: countplot , pivot table touchscreen

Όσον αφορά την ips οθόνη (είναι ένα είδος οθόνης που έχει κατά κύριο λόγο πιο φυσικά χρώματα και επιτρέπει την προβολή της οθόνης οπό πολλές γωνίες θέασης χωρίς να αλλοιώνονται τα χρώματα) δεν φαίνεται ξεκάθαρα να είναι εάν χαρακτηριστικό οπού αυξάνει την τιμή των λαπτοπ παρόλο που τα λαπτοπ που την περιέχουν έχουν υψηλότερη

μέση τιμή. Και για αυτόν ακριβός τον λόγο κάνουμε και έναν μονόπλευρο έλεγχο υποθέσεων συγκεκριμένα θέτουμε στην μηδενική υπόθεση ότι τα λαπτοπ με ips οθόνη είναι φθηνότερα σε σχέση με εκείνα οπού δεν έχουν και η στην η1 το ακριβός αντίθετο ότι τα λαπτοπ με ips οθόνη είναι ακριβότερα σε σχέση με εκείνα οπού δεν έχουν, το αποτέλεσμα του ελέγχου με α=0,05 με άνισες διακυμάνσεις με βάση το τ στατιστικό είναι ότι όντως ο αναλυτής πρέπει να απορρίψει την μηδενική υπόθεση και αρά αυτό σημαίνει ότι τα λαπτοπ με ips οθόνη είναι ακριβότερα σε σχέση με εκείνα που δεν την περιλαμβάνουν.



IPSpanel	Max Price	Average Price	Min Price	Count
Yes	4899.0	1416.6	255.0	357
No	6099.0	1025.4	174.0	918

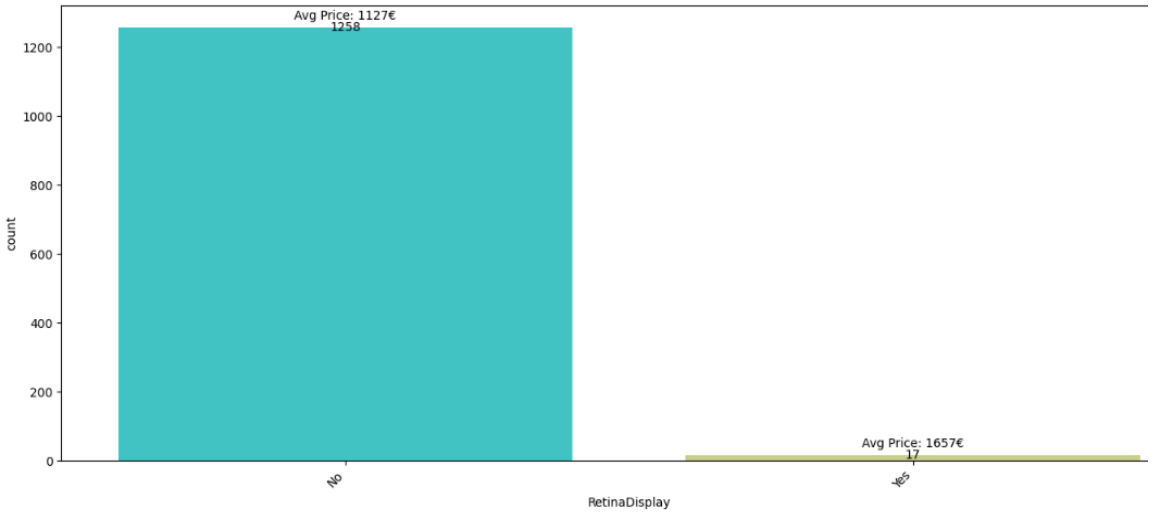
Εικονα 157: countplot , pivot table ipspanel

```python
from scipy.stats import ttest_ind, t

ips_yes = laptop[laptop['IPSpanel'] == 'Yes']['Price_euros']
ips_no = laptop[laptop['IPSpanel'] == 'No']['Price_euros']

t_statistic, p_value_two_tailed = ttest_ind(ips_yes, ips_no, equal_var=False)

n1 = len(ips_yes)
n2 = len(ips_no)
s1 = ips_yes.var(ddof=1)
s2 = ips_no.var(ddof=1)
df = ((s1 / n1 + s2 / n2) ** 2) / (((s1 / n1) ** 2) / (n1 - 1) + ((s2 / n2) ** 2) / (n2 - 1))

alpha = 0.05
t_critical = t.ppf(1 - alpha, df)

print(f"T-statistic: {t_statistic}")
print(f"Critical t-value (one-tailed, alpha={alpha}): {t_critical}")

if t_statistic > t_critical:
    print("Reject the null hypothesis.")
    print("Laptops with ips Display are priced significantly higher than those without.")
else:
    print("Fail to reject the null hypothesis.")
    print("There is no statistically significant evidence that laptops with ips Display are priced higher.")
```

```
T-statistic: 8.867142794160307
Critical t-value (one-tailed, alpha=0.05): 1.6474046669496918
Reject the null hypothesis.
Laptops with ips Display are priced significantly higher than those without.
```

Εικονα 157: hypothesis testing ipspanel vs without ipspanel

Στην συνεχεία όπως με την ips οθόνη αντίστοιχη τεχνολογία οθόνης έχει αναπτύξει η apple για τα δικά της laptop την όποια έχει ονομάσει retina display και όπως ευκολά μπορούμε να παρατηρήσουμε η ύπαρξη της αυξάνει την μέση τιμή σε σχέση με τα υπόλοιπα λαπτοπ που δεν την συμπεριλαμβάνουν.



RetinaDisplay	Max Price	Average Price	Min Price	Count
Yes	2858.0	1657.9	449.0	17
No	6099.0	1127.9	174.0	1258

Εικονα 158: countplot , pivot table retinadisplay

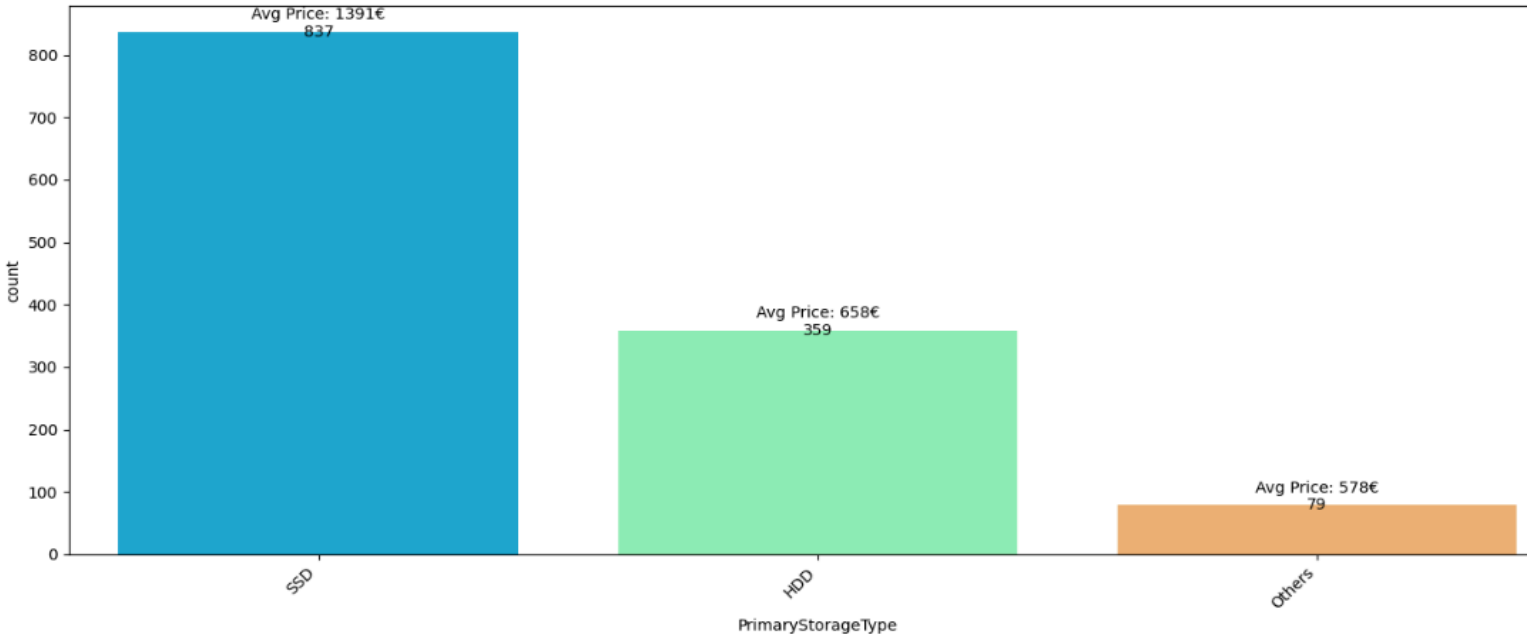
Στην συνεχεία εφόσον δεν κάναμε κάποιον έλεγχο υποθέσεων για το retina display θα ελέγξουμε εάν τα gaming laptop είναι ακριβότερα σε σχέση με τα μη gaming laptop. Η η0 υπόθεση αναφέρει ότι τα gaming είναι φθηνότερα σε σχέση με τα μη gaming ενώ η η1 ότι τα gaming είναι ακριβότερα σε σχέση με τα απλά λαπτοπ. Οπότε ο έλεγχος είναι μονόπλευρος με α=0,05 και θεωρούμε για άλλη μια φορά ότι οι διακυμάνσεις των δειγμάτων είναι άνισες. Το αποτέλεσμα είναι το αναμενόμενο δηλαδή τα gaming laptop με το συγκεκριμένο τ στατιστικό είναι στατιστικά ακριβότερα σε σχέση με τα μη gaming laptop οπότε τα καταστήματα λιανικών πωλήσεων κατά την τιμολόγηση τέτοιων προϊόντων θα πρέπει να τα τιμολογούν σχετικά υψηλοτέρα σε σύγκριση με τα απλά λαπτοπ ώστε να μην υπάρχει κανιβαλισμός των πωλήσεων των απλών λαπτοπ.

```
from scipy.stats import ttest_ind, t

gaming_yes = laptop[laptop['TypeName'] == 'Gaming']['Price_euros']
gaming_no = laptop[laptop['TypeName'] != 'Gaming']['Price_euros']

t_statistic, p_value_two_tailed = ttest_ind(gaming_yes, gaming_no, equal_var=False)

n1 = len(gaming_yes)
n2 = len(gaming_no)
s1 = gaming_yes.var(ddof=1)
s2 = gaming_no.var(ddof=1)
df = ((s1 / n1 + s2 / n2) ** 2) / (((s1 / n1) ** 2) / (n1 - 1) + ((s2 / n2) ** 2) / (n2 - 1))

alpha = 0.05
t_critical = t.ppf(1 - alpha, df)

print(f"T-statistic: {t_statistic}")
print(f"Critical t-value (one-tailed, alpha={alpha}): {t_critical}")

if t_statistic > t_critical:
    print("Reject the null hypothesis.")
    print("Gaming laptops are priced significantly higher than non-gaming laptops.")
else:
    print("Fail to reject the null hypothesis.")
    print("There is no statistically significant evidence that gaming laptops are priced higher.")
```

```
T-statistic: 11.86720816275748
Critical t-value (one-tailed, alpha=0.05): 1.650962468228925
Reject the null hypothesis.
Gaming laptops are priced significantly higher than non-gaming laptops.
```

Εικονα 159: hypothesis testing gaming laptop vs no gaming

Συνεχίζουμε την ανάλυση με την κατηγορία του αποθηκευτικού χώρου όπως ευκολά μπορεί να παρατηρήσει κάνεις τα λαπτοπ με SSD ο οποίος επιτρέπει στους υπολογιστές να ανοίξουν γρηγορότερα κατά την εκκίνηση τους και ταυτόχρονα έχουν υψηλότερες ταχύτητες read write(γρηγορότερο άνοιγμα μεταφορά και εγκατάσταση προγραμμάτων) έχουν υψηλότερη μέση τιμή από τους απλούς hdd σκληρούς δίσκους. Το παραπάνω αποδεικνύεται και από τον αντίστοιχο μονόπλευρο έλεγχο υποθέσεων με α=0,05 και άνισες διακυμάνσεις των δειγμάτων.



PrimaryStorageType	Max Price	Average Price	Min Price	Count
SSD	6099.0	1391.9	174.0	837
HDD	2899.0	658.4	224.0	359
Others	2140.0	578.5	191.9	79

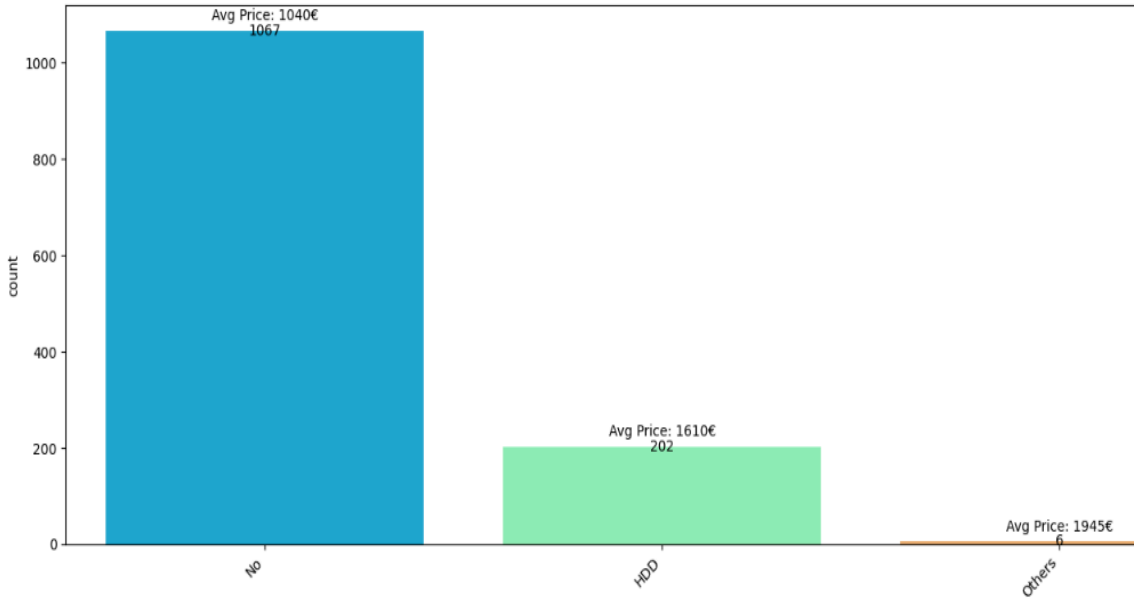
Εικονα 160: countplot , pivot table primarystorage

```
ssd_yes = laptop[laptop['PrimaryStorageType'] == 'SSD']['Price_euros']
ssd_no = laptop[laptop['PrimaryStorageType'] != 'SSD']['Price_euros']

t_statistic, p_value_two_tailed = ttest_ind(ssd_yes, ssd_no, equal_var=False)

n1 = len(ssd_yes)
n2 = len(ssd_no)
s1 = ssd_yes.var(ddof=1)
s2 = ssd_no.var(ddof=1)
df = ((s1 / n1 + s2 / n2) ** 2) / (((s1 / n1) ** 2) / (n1 - 1) + ((s2 / n2) ** 2) / (n2 - 1))

alpha = 0.05
t_critical = t.ppf(1 - alpha, df)

print(f"T-statistic: {t_statistic}")
print(f"Critical t-value (one-tailed, alpha={alpha}): {t_critical}")

if t_statistic > t_critical:
    print("Reject the null hypothesis.")
    print("Laptops with SSDs are priced significantly higher than those without SSDs.")
else:
    print("Fail to reject the null hypothesis.")
    print("There is no statistically significant evidence that laptops with SSDs are priced higher.")
```

```
T-statistic: 25.444780402638475
Critical t-value (one-tailed, alpha=0.05): 1.6460548214334911
Reject the null hypothesis.
Laptops with SSDs are priced significantly higher than those without SSDs.
```

Εικονα 160: έλεγχος υποθέσεων ssd vs no ssd

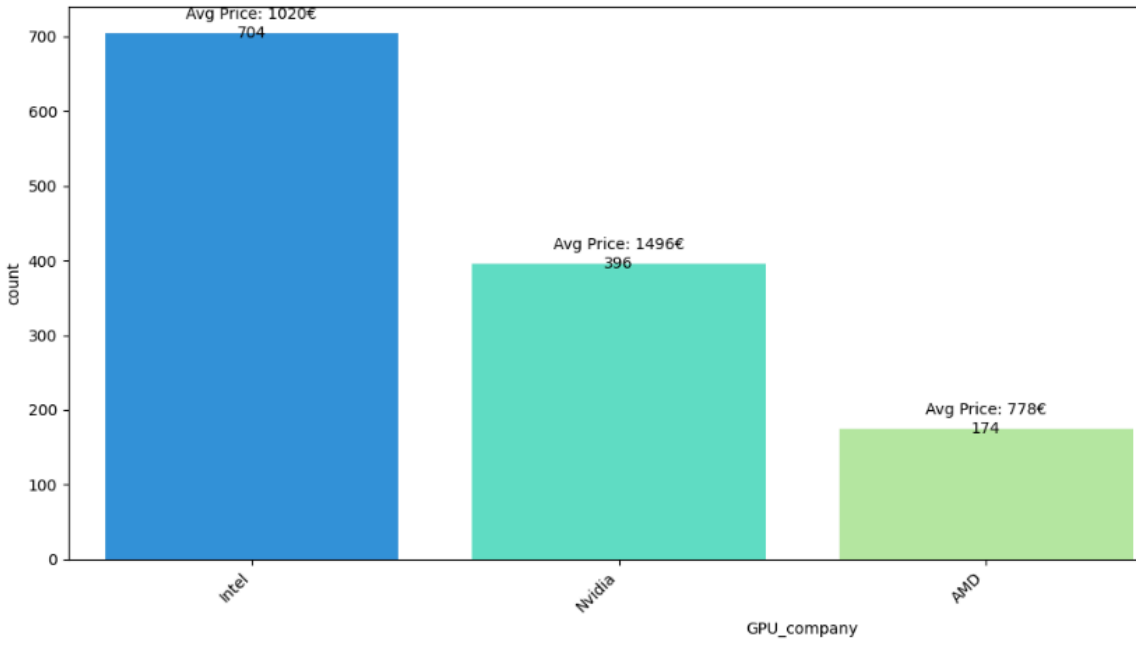
Έχοντας αναλύσει τα παραπάνω συνεχίζουμε την κατηγορία του δευτέρου αποθηκευτικού χώρου, στο οποίο δεν μπορούμε να εξάγουμε ένα ξεκάθαρο συμπέρασμα για την επίδραση του στην τιμή των λαπτοπ. Όπως παρατηρούμε τα περισσότερα λαπτοπ δεν έχουν δεύτερο αποθηκευτικό χώρο και υπάρχει μεγάλη διακύμανση των τιμών σε κάθε κατηγορία καθώς οι ελάχιστες και οι μέγιστες τιμές έχουν μεγάλη απόκλιση μεταξύ τους.



SecondaryStorageType	Max Price	Average Price	Min Price	Count
Others	3240.0	1945.7	1279.0	6
HDD	3890.0	1610.4	499.0	202
No	6099.0	1040.4	174.0	1067

Εικονα 161: countplot , pivot table secondary storage type

Συνεχίζουμε την ανάλυση με τις κατασκευάστριες εταιρείες των καρτών γραφικών όπως μπορούμε να δούμε η μέση τιμή των λαπτοπ που έχουν ξεχωριστή δεύτερη κάρτα γραφικών της Nvidia είναι υψηλότερη σε σχέση με εκείνα τα λαπτοπ που διαθέτουν μόνο την ενσωματωμένη κάρτα γραφικών του επεξεργαστή όπως αυτή της intel , όσον αφορά την διαφορά στην μέση τιμή μεταξύ των ενσωματωμένων γραφικών της intel σε σχέση με τις αντίστοιχες τις amd βλέπουμε ότι η μέση τιμή των πρώτων είναι υψηλότερη. Στον έλεγχο υποθέσεων θέτουμε ως μηδενική υπόθεση ότι τα λαπτοπ με κάρτα γραφικών της Nvidia είναι φθηνότερα σε σχέση με τα λαπτοπ που δεν τις διαθέτουν ενώ η η1 είναι ακριβός η αντίθετη υπόθεση δηλαδή ότι τα λαπτοπ με Nvidia κάρτες γραφικών έχουν υψηλότερη τιμή σε σχέση με αυτά που δεν την διαθέτουν. Ο έλεγχος είναι μονόπλευρος με άνισες διακυμάνσεις των δειγμάτων το α=0,05 και με βάση το τ στατιστικό απορρίπτουμε την μηδενική απόφαση και σε επίπεδο σημαντικότητας 95% ισχύει η η1.



GPU_company	Max Price	Average Price	Min Price	Count
Nvidia	6099.0	1496.7	459.0	396
Intel	3100.0	1020.4	174.0	704
AMD	2899.0	778.0	199.0	174
ARM	659.0	659.0	659.0	1

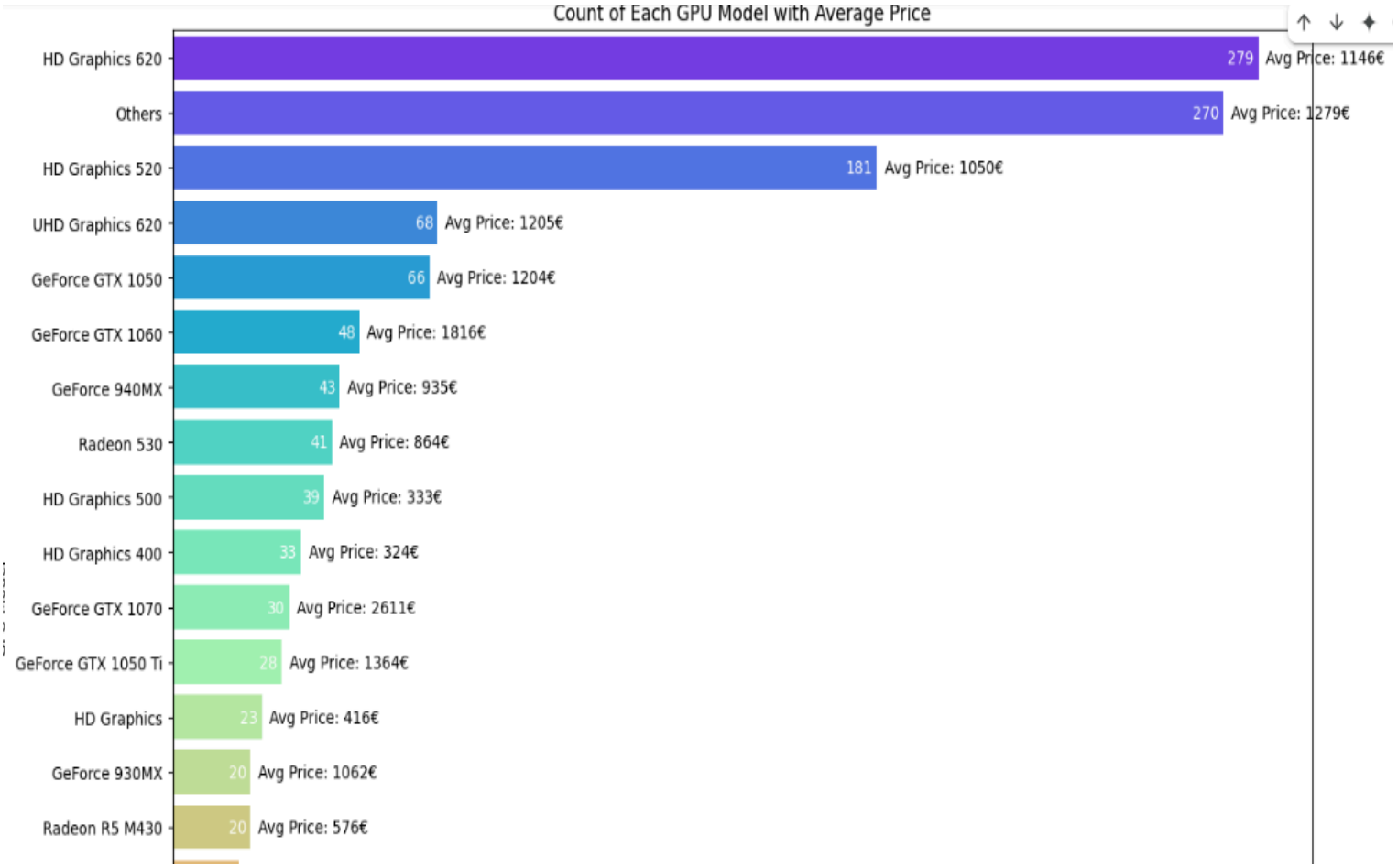
Εικονα 162: countplot , pivot table gpucompany

```
gpunvidia_yes = laptop[laptop['GPU_company'] == 'Nvidia']['Price_euros']
gpunvidia_no = laptop[laptop['GPU_company'] != 'Nvidia']['Price_euros']
t_statistic, p_value_two_tailed = ttest_ind(gpunvidia_yes, gpunvidia_no, equal_var=False)

n1 = len(gpunvidia_yes)
n2 = len(gpunvidia_no)
s1 = gpunvidia_yes.var(ddof=1)
s2 = gpunvidia_no.var(ddof=1)
df = ((s1 / n1 + s2 / n2) ** 2) / (((s1 / n1) ** 2) / (n1 - 1) + ((s2 / n2) ** 2) / (n2 - 1))

alpha = 0.05
t_critical = t.ppf(1 - alpha, df)

print(f"T-statistic: {t_statistic}")
print(f"Critical t-value (one-tailed, alpha={alpha}): {t_critical}")

if t_statistic > t_critical:
    print("Reject the null hypothesis.")
    print("Laptops with Nvidia GPUs are priced significantly higher than those without Nvidia GPUs.")
else:
    print("Fail to reject the null hypothesis.")
    print("There is no statistically significant evidence that laptops with Nvidia GPUs are priced higher.")
```

```
T-statistic: 11.562002740599011
Critical t-value (one-tailed, alpha=0.05): 1.647507838708973
Reject the null hypothesis.
Laptops with Nvidia GPUs are priced significantly higher than those without Nvidia GPUs.
```

Εικονα 163: hypothsis testing nvidia vs no nvidia gpu

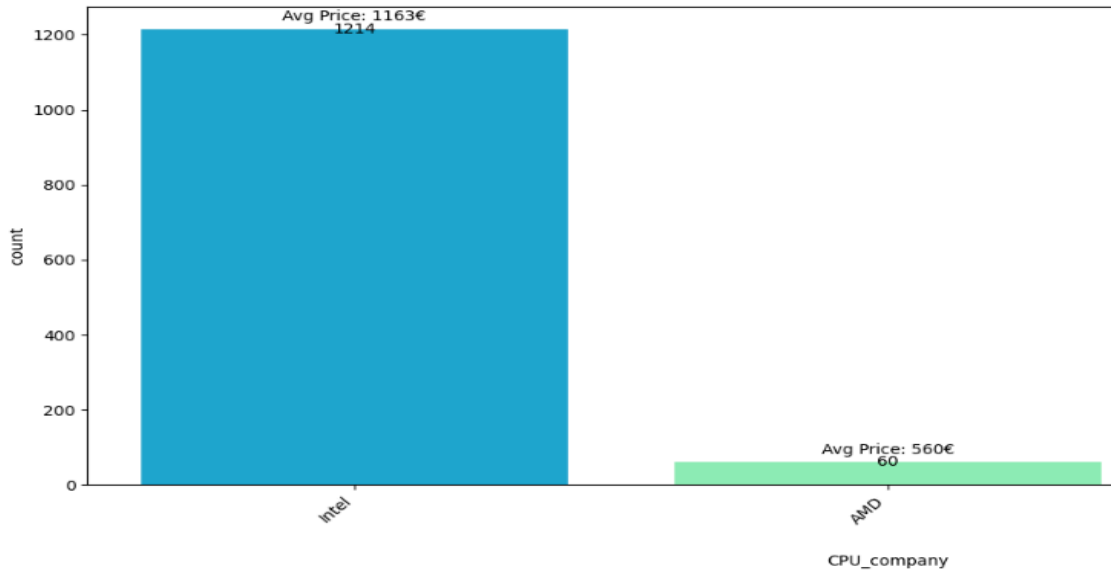
Στην συνεχεία παρουσιάζονται το διάγραμμα μέση τιμής και το αντίστοιχο pivot table για κάθε μοντέλο κάρτας γραφικών ξεχωριστά, και μπορούμε να δούμε ότι τα μοντέλα καρτών γραφικών της Nvidia όπως η gtx 1070 που χρησιμοποιείται σε gaming laptop έχουν πόλη μεγαλύτερη μέση τιμή σε σχέση με τα λαπτοπ με ενσωματωμένη κάρτα γραφικών.



GPU_model	Max Price	Average Price	Min Price	Count
GeForce GTX 1080	6099.0	4007.3	2758.0	6
GeForce GTX 980	3975.0	3975.0	3975.0	1
Quadro M3000M	3949.4	3949.4	3949.4	1
GeForce GTX 1070M	3588.8	3588.8	3588.8	1
Quadro M2000M	4389.0	3379.5	2370.0	2
...	...	...	...	...
Radeon R3	379.0	379.0	379.0	1
HD Graphics 500	615.0	334.0	202.9	39
HD Graphics 400	646.3	324.2	191.9	33
Radeon R2 Graphics	349.0	323.5	297.0	4
Radeon R2	349.0	282.8	199.0	5

Εικονα 164: countplot , pivot table gpumodel

Αναφορικά με της κατασκευάστριες εταιρείες επεξεργαστών παρατηρούμε ότι τα περισσότερα λαπτοπ φέρουν επεξεργαστές της ιντελ και μόνο 60 από τα 1275 έχουν επεξεργαστή της amd επίσης , παρατηρούμε ότι η μέση τιμή των λαπτοπ με ιντελ επεξεργαστή είναι αρκετά υψηλότερη σε σχέση με εκείνα με amd. Η παραπάνω υπόθεση αποδεικνύεται και στατιστικά μέσο του αντίστοιχου μονοπλεύρου ελέγχου οπού έχουμε θέσει ως μηδενική υπόθεση ότι τα λαπτοπ με ιντελ επεξεργαστή είναι φθηνότερα σε σχέση με εκείνα χωρίς ιντελ επεξεργαστή και το αντίθετο δηλαδή ότι είναι ακριβότερα σε σχέση με τα υπόλοιπα στην εναλλακτική η1 υπόθεση. Το αποτέλεσμα του ελέγχου με άνισες διακυμάνσεις και α=0,05 επικυρώνει σε επίπεδο σημαντικότητας 95% ότι η μηδενική υπόθεση δεν ισχύει και αρά τα καταστήματα λιανικών πωλήσεων πρέπει να τιμολογούν τα λαπτοπ με ιντελ επεξεργαστές υψηλοτέρα. Στην συνεχεία ακολουθεί το διάγραμμα με τις μέσες τιμές και τα count αναλυτικά για τα μοντέλα των επεξεργαστών με τις περισσότερες εμφανίσεις στα δεδομένα καθώς όπως μπορούμε να θυμηθούμε στις προηγούμενες παραγράφους τα είχαμε ομαδοποιήσει.



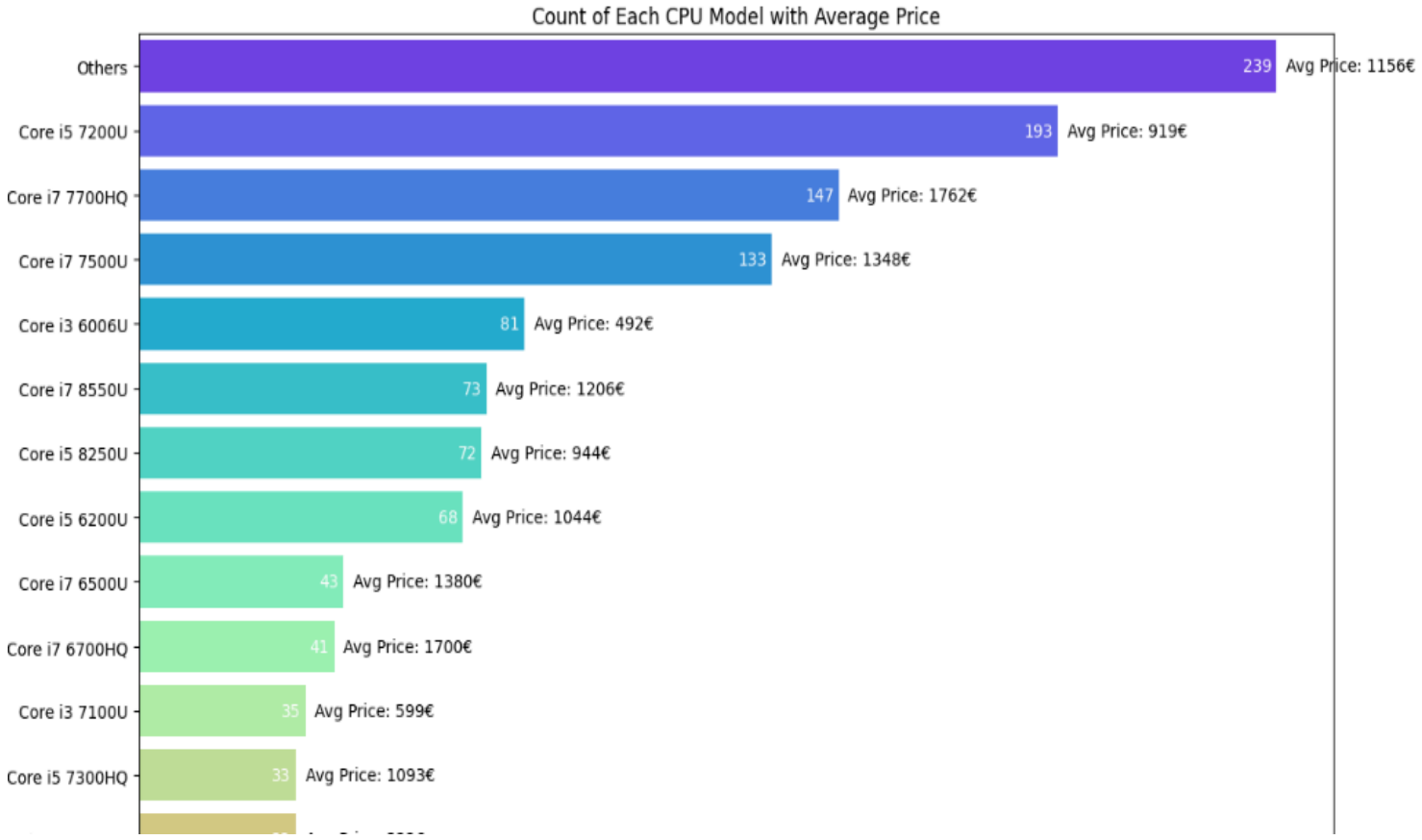
CPU_company	Max Price	Average Price	Min Price	Count
Intel	6099.0	1163.7	174.0	1214
Samsung	659.0	659.0	659.0	1
AMD	2199.0	561.0	199.0	60

Εικονα 165: countplot , pivot table cpucompany

```
intelyes=laptop[laptop['CPU_company']=='Intel']
inteinop=laptop[laptop['CPU_company']!='Intel']
from scipy.stats import ttest_ind, t

intelyes = laptop[laptop['CPU_company'] == 'Intel']['Price_euros']
intelno = laptop[laptop['CPU_company'] != 'Intel']['Price_euros']

t_statistic, p_value_two_tailed = ttest_ind(intelyes, intelno, equal_var=False)

n1 = len(intelyes)
n2 = len(intelno)
s1 = intelyes.var(ddof=1)
s2 = intelno.var(ddof=1)
df = ((s1 / n1 + s2 / n2) ** 2) / (((s1 / n1) ** 2) / (n1 - 1) + ((s2 / n2) ** 2) / (n2 - 1))

alpha = 0.05
t_critical = t.ppf(1 - alpha, df)

print(f"T-statistic: {t_statistic}")
print(f"Critical t-value (one-tailed, alpha={alpha}): {t_critical}")

if t_statistic > t_critical:
    print("Reject the null hypothesis.")
    print("Laptops with Intel CPUs are priced significantly higher than those without Intel CPUs.")
else:
    print("Fail to reject the null hypothesis.")
    print("There is no statistically significant evidence that laptops with Intel CPUs are priced higher.")
```

```
T-statistic: 12.304144902960035
Critical t-value (one-tailed, alpha=0.05): 1.6625768724151184
Reject the null hypothesis.
Laptops with Intel CPUs are priced significantly higher than those without Intel CPUs.
```

Εικονα 166: hypothesis testing cpu intel vs no intel cpu



CPU_model	Max Price	Average Price	Min Price	Count
Core i7 6600U	2620.0	1931.7	1449.0	18
Core i7 7600U	3299.0	1915.7	1099.0	13
Core i7 7700HQ	3659.4	1762.4	779.0	147
Core i7 6700HQ	3949.4	1701.0	799.0	41
Core i7 6500U	2339.0	1380.7	629.0	43
Core i7 7500U	2824.0	1349.0	579.0	133
Core i5 7300U	1760.0	1258.9	869.0	14
Core i7 8550U	2499.0	1206.5	609.0	73
Others	6099.0	1156.5	174.0	239
Core i5 7300HQ	2051.0	1093.7	709.0	33
Core i5 6200U	1690.0	1044.2	398.0	68
Core i5 8250U	1869.0	944.0	521.5	72
Core i5 7200U	2330.0	919.3	393.9	193
Core i3 7100U	949.0	599.6	384.0	35
Core i3 6006U	849.0	492.6	339.0	81
Pentium Quad Core N4200	775.0	429.3	298.0	14
Celeron Dual Core N3350	615.0	322.2	202.9	33
Celeron Dual Core N3060	475.0	307.5	209.0	25

Εικονα 163: countplot , pivot table cpumodel

Στην συνεχεία δημιουργούμε ένα pivot table με τους επεξεργαστές που φέρουν τα gaming laptop.

```
gaming_laptops = laptop[laptop['TypeName'] == 'Gaming']

pivot_table = pd.pivot_table(gaming_laptops, values='Price_euros', index='CPU_model',
                             aggfunc=[np.max, np.mean, np.min, 'count'])
pivot_table.columns = ['Max Price', 'Average Price', 'Min Price', 'Count']

pivot_table = pivot_table.sort_values(by='Average Price', ascending=False)
pivot_table = pivot_table.round(1)
pivot_table
```

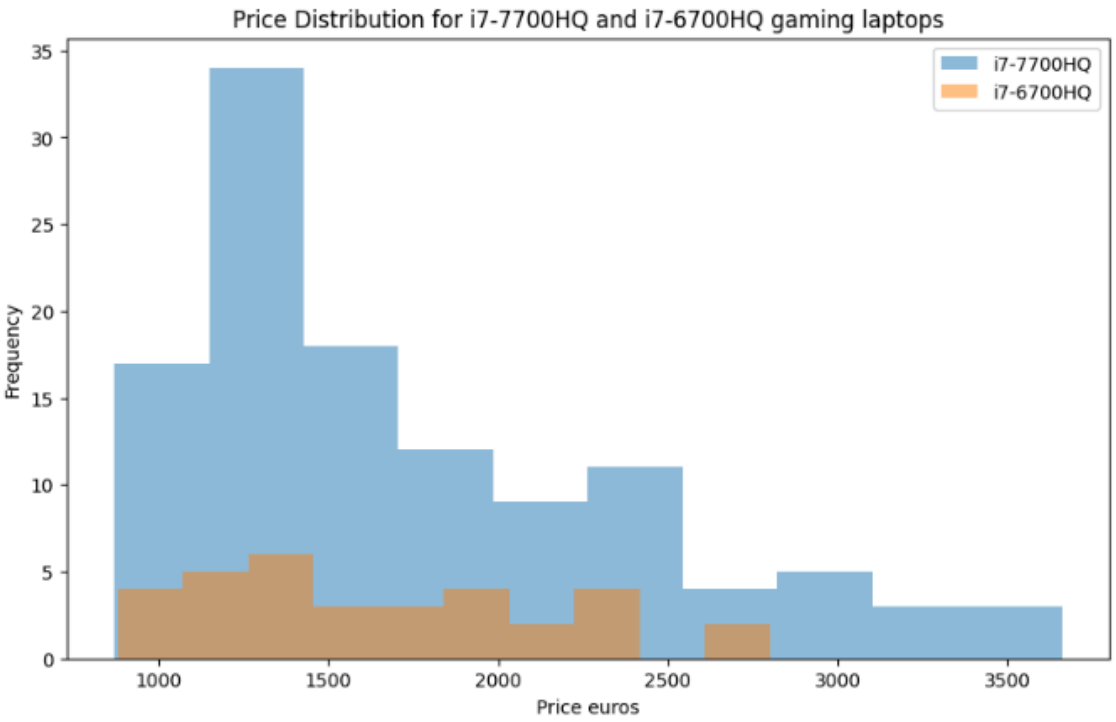
```
<ipython-input-45-a006800f61a1>:7: FutureWarning: The provided callable <function max
  pivot_table = pd.pivot_table(gaming_laptops, values='Price_euros', index='CPU_model'
<ipython-input-45-a006800f61a1>:7: FutureWarning: The provided callable <function mean
  pivot_table = pd.pivot_table(gaming_laptops, values='Price_euros', index='CPU_model'
<ipython-input-45-a006800f61a1>:7: FutureWarning: The provided callable <function min
  pivot_table = pd.pivot_table(gaming_laptops, values='Price_euros', index='CPU_model'
```

CPU_model	Max Price	Average Price	Min Price	Count
Others	6099.0	2539.0	699.0	27
Core i7 7700HQ	3659.4	1735.4	869.0	116
Core i7 6700HQ	2800.0	1642.6	879.0	33
Core i5 7300HQ	2051.0	1072.3	709.0	28
Core i7 7500U	839.0	839.0	839.0	1

Εικονα 164: countplot , pivot table cpu of gaming laptop

Εν συνεχεία δημιουργούμε δυο dataset αποκλειστικά και μόνο με τις τιμές των gaming λαπτοπ που φέρουν τους πιο ισχυρούς επεξεργαστές της ιντελ για το 2017 τον i7 7700hq και αντίστοιχα για το 2016 τον i7 6700hq (τα δεδομένα για τα λαπτοπ είναι του 2017 οπότε δεν έχουν πιο πρόσφατα μοντέλα επεξεργαστών και καρτών γραφικών παρόλα αυτά αν κάποιος θέλει να κάνει κάτι αντίστοιχο με πιο πρόσφατα δεδομένά πχ αποκτώντας τα ψάχνοντας σε ιστοσελίδες όπως το skroutz η μεθοδολογία και η ανάλυση θα ήταν παρόμοια)

```
i77700hqg=gaming_laptops[gaming_laptops['CPU_model']=='Core i7 7700HQ']['Price_euros']
i76700hqg=gaming_laptops[gaming_laptops['CPU_model']=='Core i7 6700HQ']['Price_euros']

plt.figure(figsize=(10, 6))
plt.hist(i77700hqg, bins=10, alpha=0.5, label='i7-7700HQ')
plt.hist(i76700hqg, bins=10, alpha=0.5, label='i7-6700HQ')
plt.xlabel('Price euros')
plt.ylabel('Frequency')
plt.title('Price Distribution for i7-7700HQ and i7-6700HQ gaming laptops')
plt.legend(loc='upper right')
plt.show()
```



Έχοντας τα παραπάνω δεδομένά κάνουμε τον αντίστοιχο μονόπλευρο έλεγχο υποθέσεων στην μηδενική υπόθεση θεωρούμε ότι τα λαπτοπ με τον επεξεργαστή του 2017 έχουν μικρότερη η ίση τιμή με αυτά που φέρουν τον επεξεργαστή του 2016 και αντίθετα στην η1 θεωρούμε ότι τα λαπτοπ με επεξεργαστή του 2017 είναι ακριβότερα σε σχέση με εκείνα του 2016. Θεωρούμε ότι οι διακυμάνσεις μεταξύ των δεδομένων είναι άνισες και το α=0,05 με βάση το τ στατιστικό που προκύπτει δεν μπορούμε να απορρίψουμε την μηδενική υπόθεση οπότε δεν υπάρχει στατιστική διαφορά σε επίπεδο σημαντικότητας 95% στην μέση τιμή μεταξύ του μοντέλου επεξεργαστή του 2016 σε σχέση με εκείνο του 2017.

```
i77700hqg=gaming_laptops[gaming_laptops['CPU_model']=='Core i7 7700HQ']['Price_euros']
i76700hqg=gaming_laptops[gaming_laptops['CPU_model']=='Core i7 6700HQ']['Price_euros']
num_i777 = len(i77700hqg)
num_i767 = len(i76700hqg)
print(f"Number of i7 7700hq gaming laptops: {num_i777}")
print(f"Number of i7 6700hq Gaming laptops: {num_i767}")

from scipy.stats import ttest_ind, t

t_statistic, p_value_two_tailed = ttest_ind(i77700hqg, i76700hqg, equal_var=False)
n1 = len(i77700hqg)
n2 = len(i76700hqg)
s1 = i77700hqg.var(ddof=1)
s2 = i76700hqg.var(ddof=1)
df = ((s1 / n1 + s2 / n2) ** 2) / (((s1 / n1) ** 2) / (n1 - 1) + ((s2 / n2) ** 2) / (n2 - 1))
alpha = 0.05
t_critical = t.ppf(1 - alpha, df)

print(f"T-statistic: {t_statistic}")
print(f"Critical t-value (one-tailed, alpha={alpha}): {t_critical}")

if t_statistic > t_critical:
    print("Reject the null hypothesis.")
    print("i7 7700hq gaming laptops are priced significantly higher than i7 6700hq gaming laptops.")
else:
    print("Fail to reject the null hypothesis.")
    print("There is no statistically significant evidence that i7 7700hq gaming laptops are priced significantly higher than i7 6700hq gaming laptops.")
```

```
Number of i7 7700hq gaming laptops: 116
Number of i7 6700hq Gaming laptops: 33
T-statistic: 0.8394101196876824
Critical t-value (one-tailed, alpha=0.05): 1.668885900221976
Fail to reject the null hypothesis.
There is no statistically significant evidence that i7 7700hq gaming laptops are priced significantly higher than i7 6700hq gaming laptops.
```

Εικονα 165: hypothesis testing i7 7700hq vs i7 6700hq

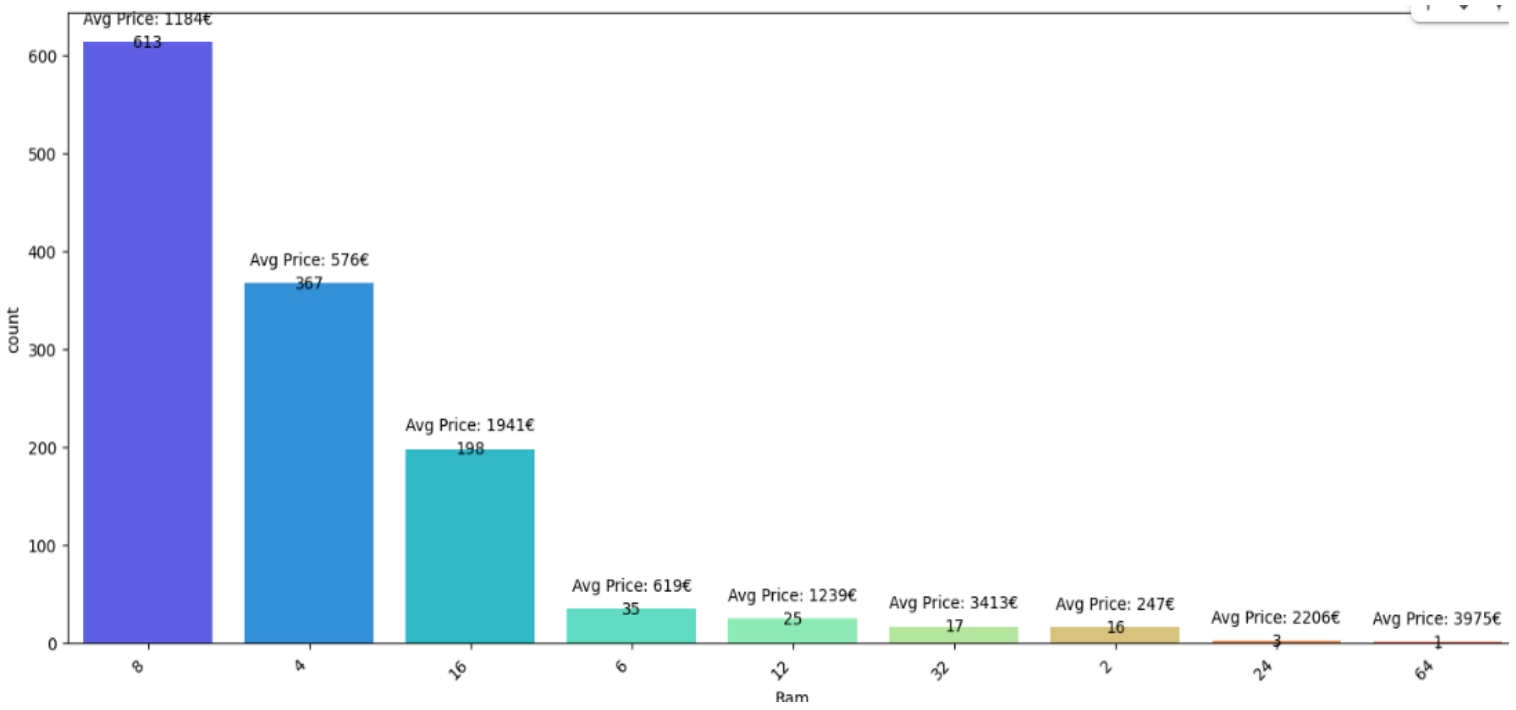
Έχοντας ολοκληρώσει το παραπάνω έλεγχο υποθέσεων θα συνεχίσουμε με άλλον έναν αυτή την φορά θα θεωρήσουμε στην μηδενική υπόθεση ότι τα λαπτοπ της εταιρείας apple είναι φθηνότερα ή έχουν ίση τιμή με τα gaming laptop ενώ στην εναλλακτική υπόθεση θα θεωρήσουμε το ανάποδο δηλαδή ότι τα λαπτοπ της apple είναι κατά μέσο ορό ακριβότερα σε σχέση με τα λαπτοπ για παιχνίδια. Ο έλεγχος για άλλη μια φορά είναι μονόπλευρος με άνισες διακυμάνσεις και α=0,05 το τ στατιστικό που προκύπτει δεν επιτρέπει στον αναλυτή να απορρίψει την μηδενική απόφαση και αρά σε επίπεδο σημαντικότητας 95% τα λαπτοπ της apple δεν είναι ακριβότερα σε σχέση με τα gaming.

```python
apple = laptop[laptop['Company'] == 'Apple']['Price_euros']
gaming_yes = laptop[laptop['TypeName'] == 'Gaming']['Price_euros']
num_apple = len(apple)
num_gaming = len(gaming_yes)
print(f"Number of Apple laptops: {num_apple}")
print(f"Number of Gaming laptops: {num_gaming}")

from scipy.stats import ttest_ind, t

t_statistic, p_value_two_tailed = ttest_ind(apple, gaming_yes, equal_var=False)
n1 = len(apple)
n2 = len(gaming_yes)
s1 = apple.var(ddof=1)
s2 = gaming_yes.var(ddof=1)
df = ((s1 / n1 + s2 / n2) ** 2) / (((s1 / n1) ** 2) / (n1 - 1) + ((s2 / n2) ** 2) / (n2 - 1))
alpha = 0.05
t_critical = t.ppf(1 - alpha, df)

print(f"T-statistic: {t_statistic}")
print(f"Critical t-value (one-tailed, alpha={alpha}): {t_critical}")

if t_statistic > t_critical:
    print("Reject the null hypothesis.")
    print("Apple laptops are priced significantly higher than gaming laptops.")
else:
    print("Fail to reject the null hypothesis.")
    print("There is no statistically significant evidence that Apple laptops are priced higher than gaming laptops.")
```

```
Number of Apple laptops: 21
Number of Gaming laptops: 205
T-statistic: -1.2374147161588762
Critical t-value (one-tailed, alpha=0.05): 1.6983561259710587
Fail to reject the null hypothesis.
There is no statistically significant evidence that Apple laptops are priced higher than gaming laptops.
```

Εικονα 166: hypothesis testing apple vs gaming laptops

Στην συνεχεία αναλύουμε της διάφορες τις μέσες τιμές των λαπτοπ με βάση την μνήμη ραμ που διαθέτουν, όπως μπορούμε ευκολά να δούμε καθώς αυξάνεται η χωρητικότητα της μνήμης αυξάνεται και η μέση τιμή των λαπτοπ.



Ram	Max Price	Average Price	Min Price	Count
64	3975.0	3975.0	3975.0	1
32	6099.0	3413.1	1279.0	17
24	2968.0	2206.3	1269.0	3
16	4389.0	1941.2	859.0	198
12	2299.0	1239.4	609.0	25
8	3949.4	1184.4	329.0	613
6	949.0	619.4	409.0	35
4	1799.0	576.1	196.0	367
2	379.0	247.6	174.0	16

Εικονα 167: countplot , pivot table ram

Στην συνεχεία κάνουμε έναν έλεγχο υποθέσεων με βάση τα gaming laptop που έχουν μνήμη ραμ ίση με 16 gb σε σχέση με τα Ultrabook λαπτοπ με 16 gb ram. Όπως παρατηρούμε οι μέσες τιμές των δυο κατηγοριών είναι σχεδόν ίσες οπότε θα πρέπει να κάνουμε αναλυτικά τον αντίστοιχο έλεγχο υποθέσεων για να επιβεβαιώσουμε στατιστικά αν υπάρχει διαφορά μεταξύ των δυο.

```
ram16 = laptop[laptop['Ram'] == 16]
ram16gaming=ram16[ram16['TypeName']=='Gaming']['Price_euros']
ram16ultrabook=ram16[ram16['TypeName']=='Ultrabook']['Price_euros']
avg_price_gaming = ram16gaming.mean()
min_price_gaming = ram16gaming.min()
max_price_gaming = ram16gaming.max()

avg_price_ultrabook = ram16ultrabook.mean()
min_price_ultrabook = ram16ultrabook.min()
max_price_ultrabook = ram16ultrabook.max()


print("ram16gaming:")
print(f"  Average Price: {avg_price_gaming}")
print(f"  Min Price: {min_price_gaming}")
print(f"  Max Price: {max_price_gaming}")

print("\nram16ultrabooks:")
print(f"  Average Price: {avg_price_ultrabook}")
print(f"  Min Price: {min_price_ultrabook}")
print(f"  Max Price: {max_price_ultrabook}")
```

```
ram16gaming:
  Average Price: 1959.850404040404
  Min Price: 879.01
  Max Price: 3499.0

ram16ultrabooks:
  Average Price: 2048.3125
  Min Price: 1262.0
  Max Price: 2858.0
```

```
ram16 = laptop[laptop['Ram'] == 16]
ram16gaming=ram16[ram16['TypeName']=='Gaming']['Price_euros']
ram16ultrabook=ram16[ram16['TypeName']=='Ultrabook']['Price_euros']
num_ram16gaming = len(ram16gaming)
num_ram16ultrabook = len(ram16ultrabook)
print(f"Number of gaming laptops with 16 gb ram: {num_ram16gaming}")
print(f"Number of ulttranotebook laptops with 16 gb ram: {num_ram16ultrabook}")
from scipy.stats import ttest_ind, t
t_statistic, p_value_two_tailed = ttest_ind(ram16gaming,ram16ultrabook, equal_var=False)

n1 = len(ram16gaming)
n2 = len(ram16ultrabook)
s1 = ram16gaming.var(ddof=1)
s2 = ram16ultrabook.var(ddof=1)
df = ((s1 / n1 + s2 / n2) ** 2) / (((s1 / n1) ** 2) / (n1 - 1) + ((s2 / n2) ** 2) / (n2 - 1))

alpha = 0.05
t_critical = t.ppf(1 - alpha, df)
print(f"T-statistic: {t_statistic}")
print(f"Critical t-value (one-tailed, alpha={alpha}): {t_critical}")

if t_statistic > t_critical:
    print("Reject the null hypothesis.")
    print("gaming  laptops with 16 gb ram are priced significantly higher than ultanotebook laptops with the same amount of ram.")
else:
    print("Fail to reject the null hypothesis.")
    print("There is no statistically significant evidence that gaming  laptops with 16 gb ram are priced significantly higher than ultanotebook laptops with the same amount of ram.")
```

```
Number of gaming laptops with 16 gb ram: 99
Number of ulttranotebook laptops with 16 gb ram: 36
T-statistic: -1.0449107941353415
Critical t-value (one-tailed, alpha=0.05): 1.6609371416183005
Fail to reject the null hypothesis.
There is no statistically significant evidence that gaming  laptops with 16 gb ram are priced significantly higher than ultanotebook laptops with the same amount of ram.
```

Εικονα 167: hypothesis testing gaming vs ultrabook laptop with 16gb ram

Από τον παραπάνω μονόπλευρο έλεγχο με άνισες διακυμάνσεις και α=0,05 διαπιστώνουμε ότι σε επίπεδο σημαντικότητας 95% δεν επιβεβαιώνεται στατιστικά ότι τα gaming laptop με 16 gb ραμ είναι ακριβότερα σε σχέση με τις ιδίας χωρητικότητας ραμ Ultrabook.

Έχοντας αναλύσει της ποιοτικές μεταβλητές θα συνεχίσουμε την ανάλυση μας με τις αριθμητικές μεταβλητές. Ο τρόπος με τον οποίο θα τις αναλύσουμε είναι μέσο της απλής γραμμικής παλινδρόμησης έχοντας ως εξαρτημένη μεταβλητή την τιμή των λαπτοπ και ως ερμηνευτική την εκάστοτε μεταβλητή. Στην συνεχεία θα παρουσιάζουμε και τον συντελεστή συσχέτισης και έχοντας ολοκληρώσει τα παραπάνω θα μετατρέπουμε τις αριθμητικές μεταβλητές σε ποιοτικές μεταβλητές σαν ψευδομεταβλητες (dummy

variables) όπως κάναμε και στο κεφάλαιο 6 στην συνεχεία θα συγκρίνουμε αν τα μοντέλα μηχανικής μάθησης δίνουν καλυτέρα αποτελέσματα με τις αριθμητικές μεταβλητές σαν ψευδομεταβλητες ή αφήνοντας τες στην αρχική τους μορφή. Παρακάτω παρουσιάζεται ο κώδικας δημιουργίας της γραμμικής παλινδρόμησης και του αντίστοιχου διαγράμματος της.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import statsmodels.api as sm

X = laptop['Ram']
y = laptop['Price_euros']
X = sm.add_constant(X)

model = sm.OLS(y, X).fit()

print(model.summary())

plt.figure(figsize=(10, 6))
plt.scatter(laptop['Ram'], laptop['Price_euros'], alpha=0.5)
plt.plot(laptop['Ram'], model.predict(X), color='red', linewidth=2)
plt.xlabel('RAM (GB)')
plt.ylabel('Price (Euros)')
plt.title('Linear Regression: RAM vs. Price')
plt.grid(True)
plt.show()
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:            Price_euros   R-squared:                       0.548
Model:                            OLS   Adj. R-squared:                  0.548
Method:                 Least Squares   F-statistic:                     1544.
Date:                Thu, 16 Jan 2025   Prob (F-statistic):          9.13e-222
Time:                        15:59:11   Log-Likelihood:                -9656.4
No. Observations:                1275   AIC:                         1.932e+04
Df Residuals:                    1273   BIC:                         1.933e+04
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const        276.0273     25.538     10.808      0.000     225.926     326.129
Ram          101.7609      2.590     39.288      0.000      96.679     106.842
==============================================================================
Omnibus:                      262.134   Durbin-Watson:                   2.027
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             1100.387
Skew:                           0.920   Prob(JB):                    1.13e-239
Kurtosis:                       7.163   Cond. No.                         19.2
==============================================================================
```

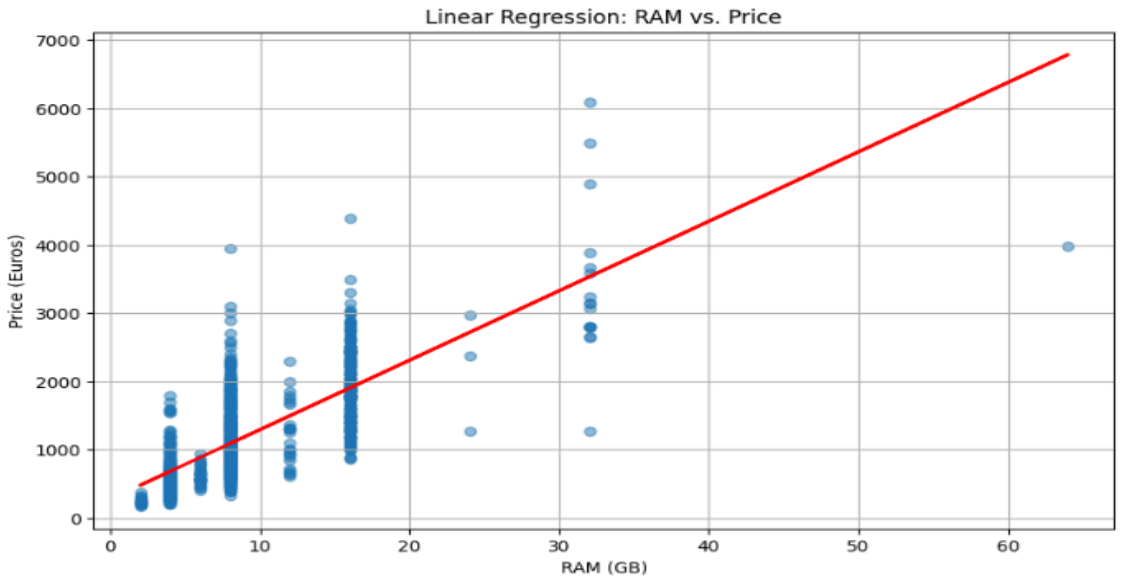
Εικονα 168: γραμμική παλινδρόμηση τιμή και μέγεθος μνήμης ραμ

Το ρ^2 της συγκεκριμένης παλινδρόμησης είναι 0,548 και η μεταβλητή ραμ έχει θετική συσχέτιση (0,74) με την τιμή πώλησης των λαπτοπ , καθώς ένα επιπλέον gigabyte μνήμης

ραμ αυξάνει την τιμή του λαπτοπ κατά 101,76 ευρώ επίσης το p value είναι ίσο με μηδέν αρά η συγκεκριμένη ερμηνευτική μεταβλητή είναι στατιστικά σημαντική.

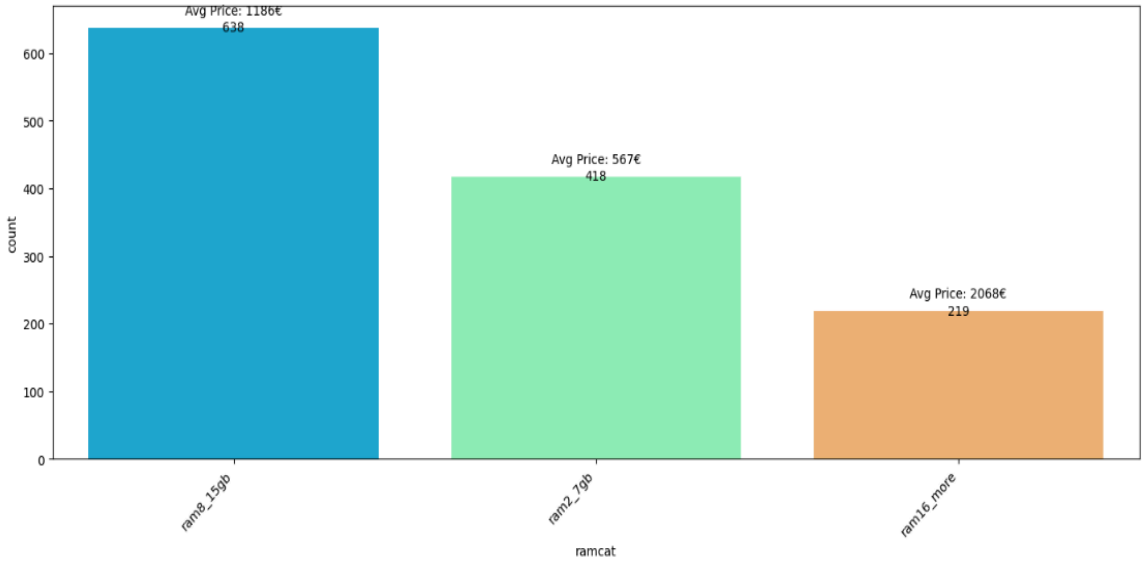
```
corralation=laptop['Ram'].corr(laptop['Price_euros'])
corralation
print(f"The correlation between Ram and Price is: {corralation}")
```

```
The correlation between Ram and Price is: 0.7402865271622695
```



Στην συνεχεία όπως αναφέραμε παραπάνω μετατρέπουμε την αριθμητική μεταβολή σε ποιοτική συγκεκριμένα την κατηγοριοποιούμε σε τρεις ομάδες σε αυτήν οπού τα λαπτοπ έχουν μνήμη ραμ από 2 έως 7 gb η δεύτερη από 8 έως 15 και η τρίτη από 16 gb και πάνω και στην συνεχεία δημιουργούμε το αντίστοιχο countplot με τις μέσες τιμές ανά κατηγορία ραμ και το pivot table.

```
laptopd['ramcat']='a'
laptopd.loc[(laptopd['Ram'] >= 2) & (laptopd['Ram'] <= 7), 'ramcat'] = 'ram2_7gb'
laptopd.loc[(laptopd['Ram'] >= 8) & (laptopd['Ram'] <= 15), 'ramcat'] = 'ram8_15gb'
laptopd.loc[laptopd['Ram'] > 15, 'ramcat'] = 'ram16_more'
```



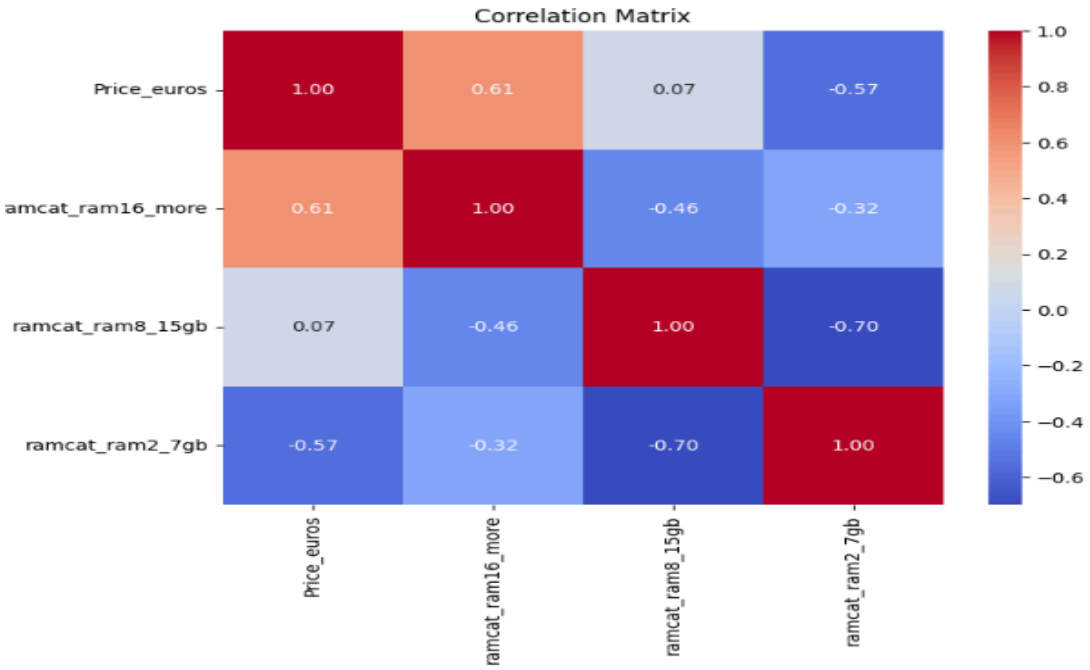
ramcat	Max Price	Average Price	Min Price	Count
ram16_more	6099.0	2068.4	859.0	219
ram8_15gb	3949.4	1186.6	329.0	638
ram2_7gb	1799.0	567.2	174.0	418

Εικονα 169: countplot , pivot table ram

Εν συνεχεία αντικαθιστούμε τις ποιοτικές τιμές με 0 και 1 με την εντολή pd.get_dummies και δημιουργούμε ένα πίνακα συσχέτισης με τον οποίο βλέπουμε ότι η τιμή έχει θετική συσχέτιση με τα λαπτοπ οπού έχουν πάνω από 16 gb μνήμης ραμ ενώ η ύπαρξη από 7 έως 15 gb μνήμης δεν έχει κάποια συσχέτιση με την τιμή και τέλος η μικρή χωρητικότητα μνήμης ραμ οδηγεί στην μείωση της τιμής του λαπτοπ λόγο της αρνητικής συσχέτισης. Αρά τα καταστήματα λιανικής πώλησης για να αυξήσουν τα κέρδη τους θα πρέπει να προμηθεύονται λαπτοπ με τουλάχιστον 8 gb μνήμης ραμ.

```
dummy_columns = pd.get_dummies(laptopd['ramcat'], prefix='ramcat', dtype=int)
laptopd = pd.concat([laptopd, dummy_columns], axis=1)
laptopd.head()
```

```
columns_for_correlation = ['Price_euros', 'ramcat_ram16_more', 'ramcat_ram8_15gb', 'ramcat_ram2_7gb']
correlation_matrix = laptopd[columns_for_correlation].corr()
import matplotlib.pyplot as plt
import seaborn as sns
plt.figure(figsize=(8, 6))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".2f")
plt.title('Correlation Matrix')
plt.show()
```



Εικονα 170: heatmap

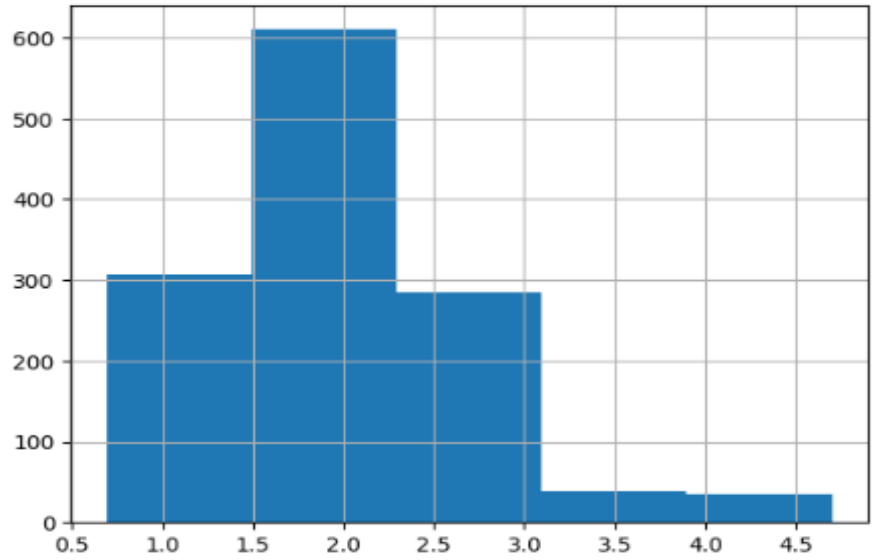
Η επόμενη αριθμητική τιμή είναι το βάρος σε κιλά των λαπτοπ , για να μπορέσουμε να τα μετατρέψουμε σε ποιοτικές μεταβλητές και στην συνεχεία σε ψευδομεταβλητες δημιουργούμε το αντίστοιχο ιστόγραμμα οι ομάδες που δημιουργήθηκαν είναι τα λαπτοπ με βάρος από 0,69 κιλά έως 1,5 κιλά η δεύτερη ομάδα από 1,51 κιλά έως 2,5 και η τρίτη από 2,51 κιλά και άνω. Όσον αφορά την γραμμική παλινδρόμηση βλέπουμε ότι το r^2 είναι πολύ μικρό κάτω από το 1% αρά η συγκεκριμένη μεταβλητή δεν ερμηνεύει καλά την μεταβλητότητα της τιμής. Ο συντελεστής συσχέτισης είναι κοντά στο 0,2 δηλαδή υπάρχει θετική συσχέτιση μεταξύ τιμής και βάρους και το ίδιο θετικό νούμερο είναι και ο συντελεστής β της ερμηνευτικής μεταβλητής ίσος με 221,8. Η παλινδρόμηση ίσος να έχει επηρεαστεί από τα gaming laptop τα όποια έχουν υψηλό βάρος και υψηλή τιμή ενώ ταυτόχρονα τα ultrabooks έχουν χαμηλό βάρος και υψηλή τιμή.

```
unique_ram_values = laptop['Weight'].unique()
unique_ram_values.sort()
print(unique_ram_values)
```

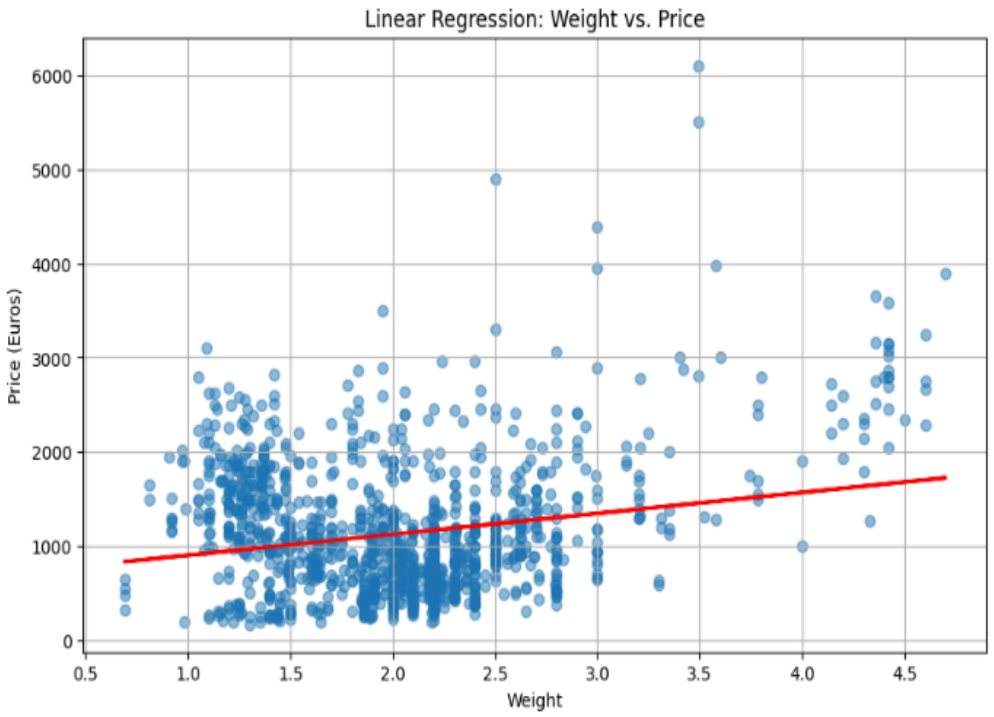
```
[0.69  0.81  0.91  0.92  0.97  0.98  0.99  1.05  1.08  1.09  1.1   1.11
 1.12  1.13  1.14  1.15  1.16  1.17  1.18  1.19  1.2   1.21  1.22  1.23
 1.24  1.25  1.252 1.26  1.27  1.28  1.29  1.3   1.31  1.32  1.34  1.35
 1.36  1.37  1.38  1.39  1.4   1.41  1.42  1.43  1.44  1.45  1.47  1.48
 1.49  1.5   1.54  1.55  1.56  1.58  1.59  1.6   1.62  1.63  1.64  1.65
 1.68  1.7   1.71  1.74  1.75  1.76  1.78  1.79  1.8   1.83  1.84  1.85
 1.86  1.87  1.88  1.89  1.9   1.91  1.93  1.94  1.95  1.96  1.98  1.99
 2.    2.02  2.03  2.04  2.05  2.06  2.07  2.08  2.09  2.1   2.13  2.14
 2.15  2.16  2.17  2.18  2.19  2.191 2.2   2.21  2.23  2.24  2.25  2.26
 2.29  2.3   2.31  2.32  2.33  2.34  2.36  2.37  2.38  2.4   2.43  2.45
 2.5   2.54  2.56  2.59  2.591 2.6   2.62  2.63  2.65  2.67  2.69  2.7
 2.71  2.72  2.73  2.75  2.77  2.79  2.8   2.83  2.9   2.94  2.99  3.
 3.14  3.2   3.21  3.25  3.3   3.31  3.35  3.4   3.42  3.49  3.52  3.58
 3.6   3.74  3.78  3.8   4.    4.14  4.2   4.3   4.33  4.36  4.4   4.42
 4.5   4.6   4.7  ]
```

```
laptop['Weight'].hist(bins=5)
```

```
<Axes: >
```



```
                            OLS Regression Results
==============================================================================
Dep. Variable:            Price_euros   R-squared:                       0.045
Model:                            OLS   Adj. R-squared:                  0.044
Method:                 Least Squares   F-statistic:                     59.84
Date:                Fri, 17 Jan 2025   Prob (F-statistic):           2.08e-14
Time:                        15:52:16   Log-Likelihood:                -10133.
No. Observations:                1275   AIC:                         2.027e+04
Df Residuals:                    1273   BIC:                         2.028e+04
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const        682.2275     61.593     11.076      0.000     561.393     803.062
Weight       221.8750     28.683      7.735      0.000     165.604     278.146
==============================================================================
Omnibus:                      323.099   Durbin-Watson:                   1.993
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              888.559
Skew:                           1.301   Prob(JB):                    1.13e-193
Kurtosis:                       6.156   Cond. No.                         8.27
==============================================================================
```

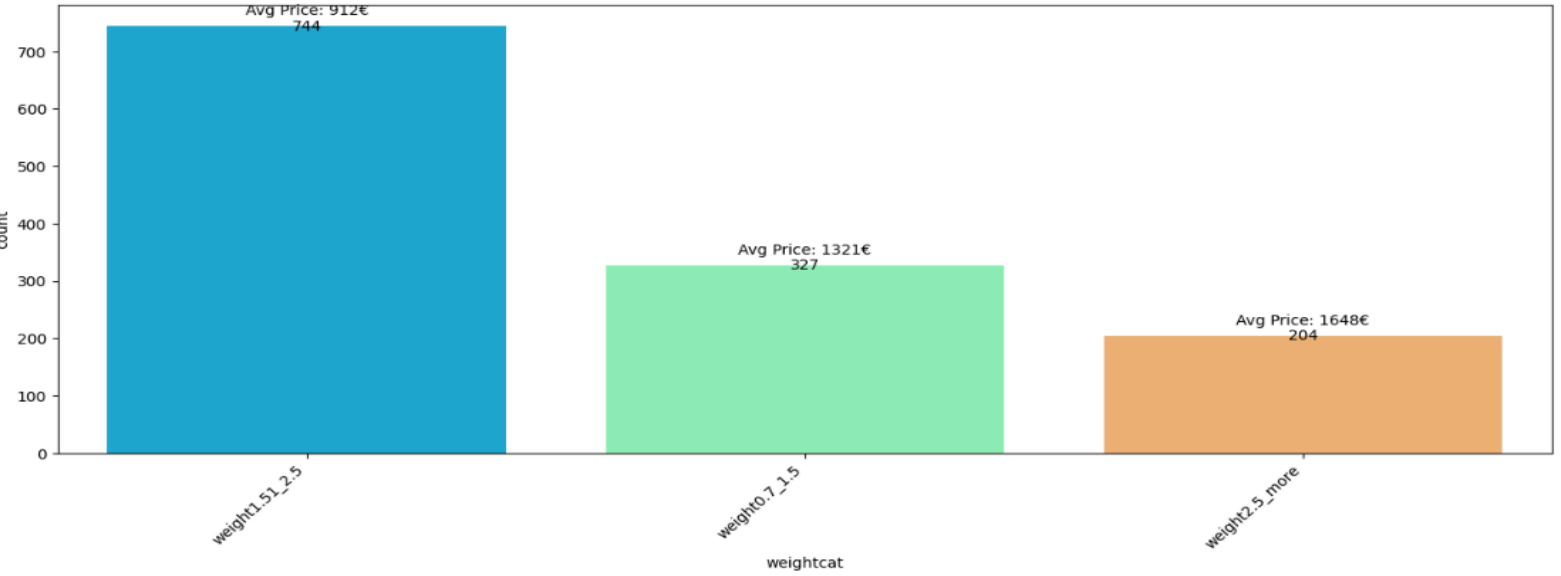



```
corralation=laptop['Weight'].corr(laptop['Price_euros'])
print(f"The correlation between Weight and Price is: {corralation}")
```

```
The correlation between Weight and Price is: 0.21188344492206526
```

Εικονα 171 : γραμμική παλινδρόμηση μεταξύ τιμής και βάρους των λαπτοπ

```
laptopd['weightcat']='a'
laptopd.loc[(laptopd['Weight'] >= 0.69) & (laptopd['Weight'] <= 1.5), 'weightcat'] = 'weight0.7_1.5'
laptopd.loc[(laptopd['Weight'] >= 1.51) & (laptopd['Weight'] <= 2.5), 'weightcat'] = 'weight1.51_2.5'
laptopd.loc[laptopd['Weight'] > 2.5, 'weightcat'] = 'weight2.5_more'
```



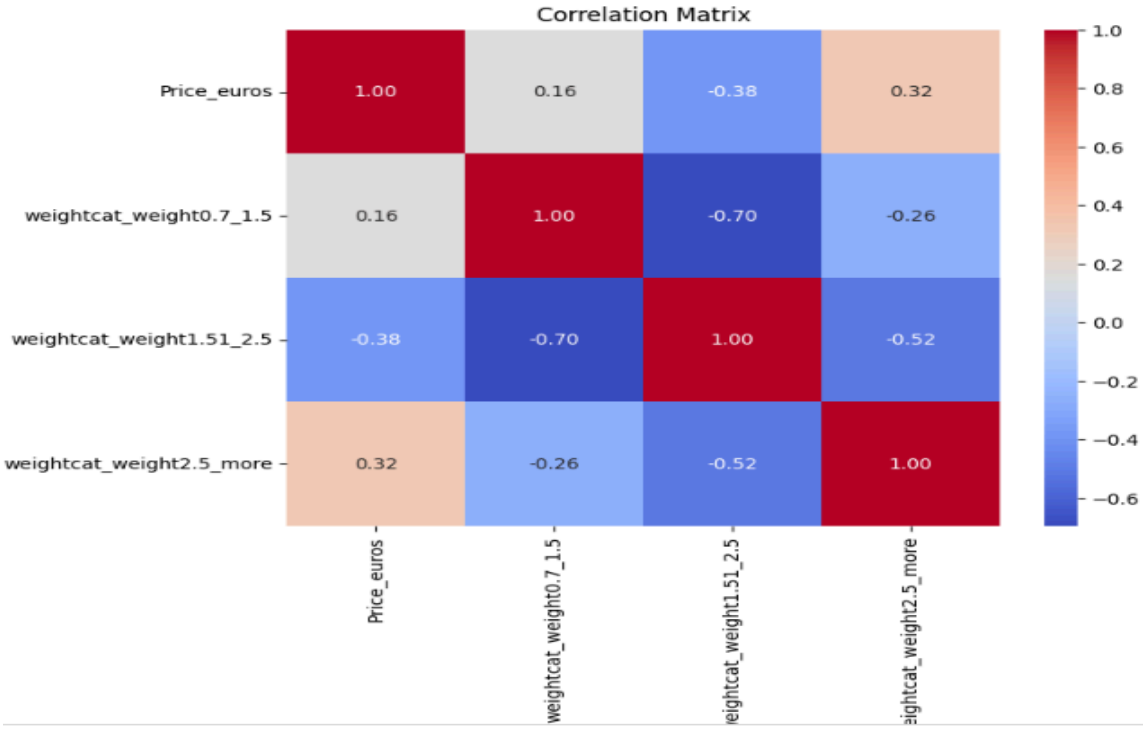
weightcat	Max Price	Average Price	Min Price	Count
weight2.5_more	6099.0	1648.9	309.0	204
weight0.7_1.5	3100.0	1321.1	174.0	327
weight1.51_2.5	4899.0	912.2	199.0	744

```
dummy_columns = pd.get_dummies(laptopd['weightcat'], prefix='weightcat', dtype=int)

laptopd = pd.concat([laptopd, dummy_columns], axis=1)

laptopd.head()
```

Εικονα 172: countplot και pivot table του βάρους των λαπτοπ

Όσον αφορά τον πίνακα συσχετισης βλέπουμε ότι η τιμή έχει θετική συσχέτιση με τα λαπτοπ τα όποια έχουν βάρος άνω των 2,51 κιλών δηλαδή τα gaming και με εκείνα οπού έχουν βάρος μεταξύ των 0,69 και 1,5 κιλών δηλαδή των ultrabooks πού είναι πολύ χρίσμα σε ανθρώπους που πρέπει να εργαστούν ενώ ταυτόχρονα μετακινούνται συχνά, τέλος η τιμή έχει αρνητική συσχέτιση με τα λαπτοπ με βάρος μεταξύ 1,51 κιλών και 2,5 καθώς

λογικά δεν έχουν ούτε πολύ υψηλές αποδόσεις ούτε είναι ελαφριά με μεγάλη χωρητικότητα μπαταρίας που να επιτρέπει στον χρήστη την χρήση τους χωρίς να είναι συνδεδεμένα στην μπριόζα όπως τα ultrabooks. Αρά τα καταστήματα λιανικών πωλήσεων λαπτοπ ίσως θα έπρεπε να εστιάσουν στην προμήθεια λαπτοπ με αυτά τα χαρακτηριστικά ώστε να αυξήσουν και το κέρδος αλλά και να μειώσουν τα αποθέματα τους από τα λαπτοπ με χαμηλή τιμή.



Εικονα 172: heatmap του βάρους

Η επόμενη αριθμητική μεταβλητή είναι το μέγεθος της οθόνης των λαπτοπ σε ίντσες όπως και με την προηγουμένη μεταβλητή έτσι και εδώ δημιουργούμε ένα ιστόγραμμα για να δούμε σε ποιες τιμές της μεταβλητής έχουν τις περισσότερες επανεμφανίσεις ώστε να γίνει με τον καλύτερο δυνατό τρόπο ο διαχωρισμός τους. Συγκεκριμένα στην πρώτη κατηγορία τοποθετήθηκαν τα λαπτοπ με μέγεθος οθόνης από 10 έως 14.1 ίντσες την δεύτερη κατηγορία τα λαπτοπ από 14.2 έως 15.6 ίντσες και στην τρίτη τα λαπτοπ με οθόνη μεγαλύτερη των 15.6 ιντσών. Αναφορικά με την γραμμική παλινδρόμηση το r^2 είναι σχεδόν μηδέν που σηματοδοτεί ότι η συγκεκριμένη μεταβλητή δεν μπορεί να ερμηνεύσει την μεταβλητότητα της τιμής των λαπτοπ, αυτό μπορεί να επιβεβαιωθεί και από το p value οπού είναι ίσο με 0,017 δηλαδή με α=0,01 δηλαδή σε επίπεδο σημαντικότητας 99% η μεταβλητή δεν είναι στατιστικά σημαντική. Παρόλα αυτά το β είναι θετικό ίσο με 32 δηλαδή μια αύξηση κατά μια ίντσα οδηγεί σε αύξηση της τιμής των λαπτοπ κατά 32 ευρώ,

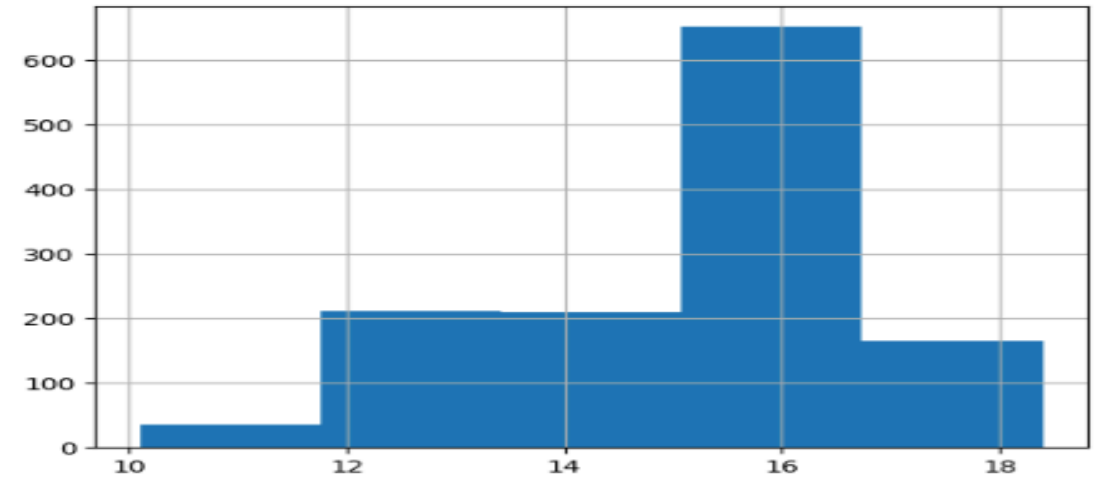
δηλαδή δημιουργεί μικρή αύξηση της τιμής το οποίο επιβεβαιώνεται και από τον συντελεστή συσχέτισης ο οποίος είναι κοντά στο μηδέν. Στην συνεχεία μετα την μετατροπή τις μεταβλητής σε ψευδομεταβλητες δημιουργούμε τον πίνακα συσχέτισης και παρατηρούμε ακριβός το ίδιο φαινόμενο με αυτό που είδαμε και με το βάρος των λαπτοπ , πιο συγκεκριμένα τα λαπτοπ με μικρές οθόνες όπως τα Ultrabook έχουν θετική συσχέτιση με την τιμή το ίδιο ισχύει και με τα λαπτοπ με μεγάλες οθόνες που συνήθως είναι τα gaming , ενώ αρνητική συσχέτιση υπάρχει με τα λαπτοπ οπού δεν έχουν ούτε πολύ μεγάλη ούτε πολύ μικρή οθόνη.

```
unique_ram_values = laptop['Inches'].unique()
unique_ram_values
```

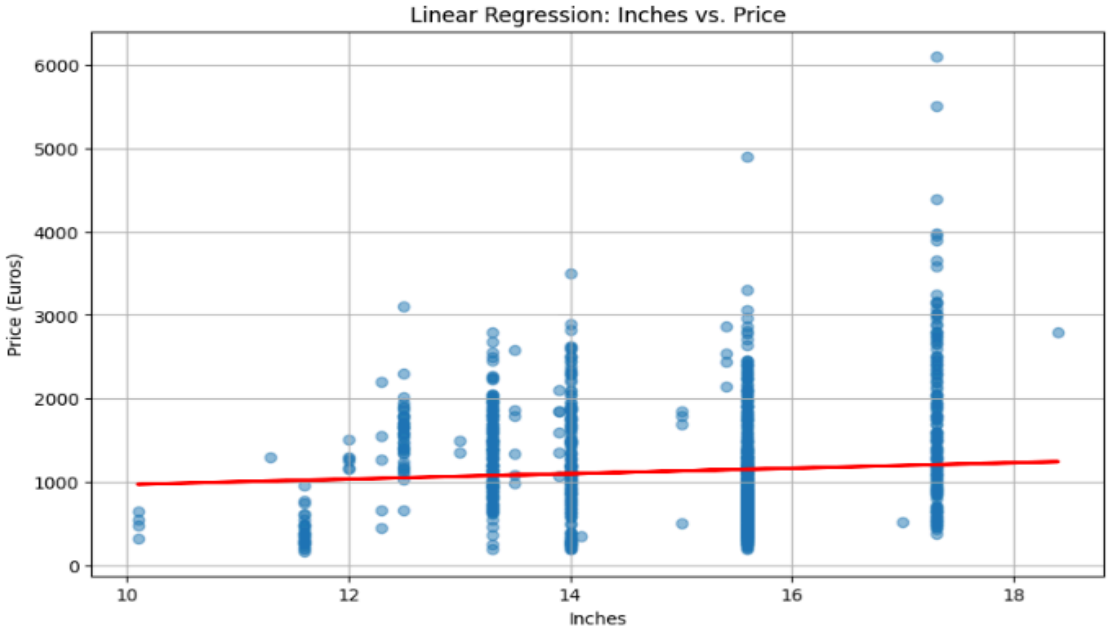
```
array([13.3, 15.6, 15.4, 14. , 12. , 11.6, 17.3, 10.1, 13.5, 12.5, 13. ,
       18.4, 13.9, 12.3, 17. , 15. , 14.1, 11.3])
```

```
laptop['Inches'].hist(bins=5)
```

```
<Axes: >
```



```
                            OLS Regression Results
==============================================================================
Dep. Variable:            Price_euros   R-squared:                       0.004
Model:                            OLS   Adj. R-squared:                  0.004
Method:                 Least Squares   F-statistic:                     5.673
Date:                Thu, 16 Jan 2025   Prob (F-statistic):             0.0174
Time:                        16:54:20   Log-Likelihood:                -10160.
No. Observations:                1275   AIC:                         2.032e+04
Df Residuals:                    1273   BIC:                         2.033e+04
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const        644.4346    206.880      3.115      0.002     238.571    1050.299
Inches        32.6524     13.709      2.382      0.017       5.757      59.547
==============================================================================
Omnibus:                      379.872   Durbin-Watson:                   2.016
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             1278.594
Skew:                           1.447   Prob(JB):                    2.27e-278
Kurtosis:                       6.961   Cond. No.                         160.
==============================================================================
```

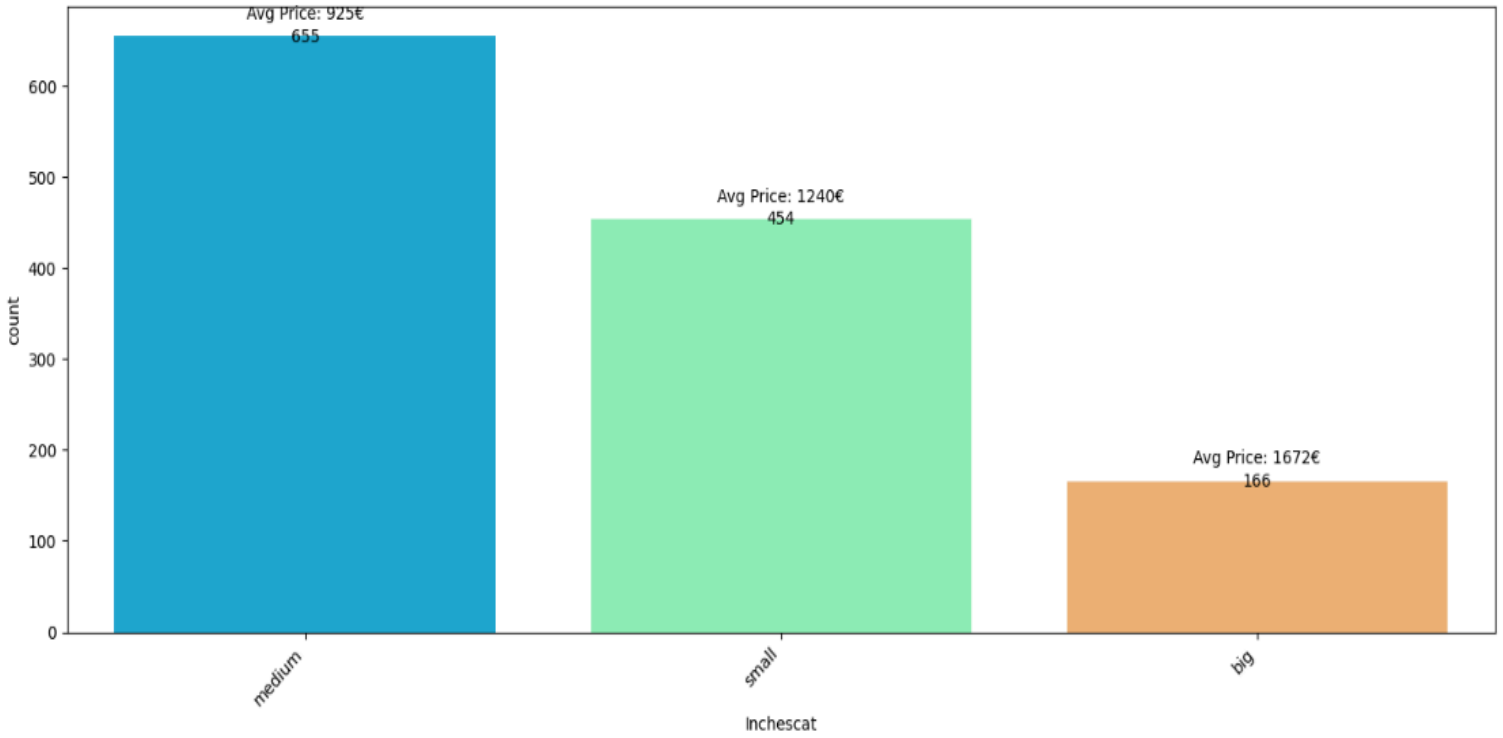



```
corralation=laptop['Inches'].corr(laptop['Price_euros'])
print(f"The correlation between Inches and Price is: {corralation}")
```

```
The correlation between Inches and Price is: 0.06660794107004478
```

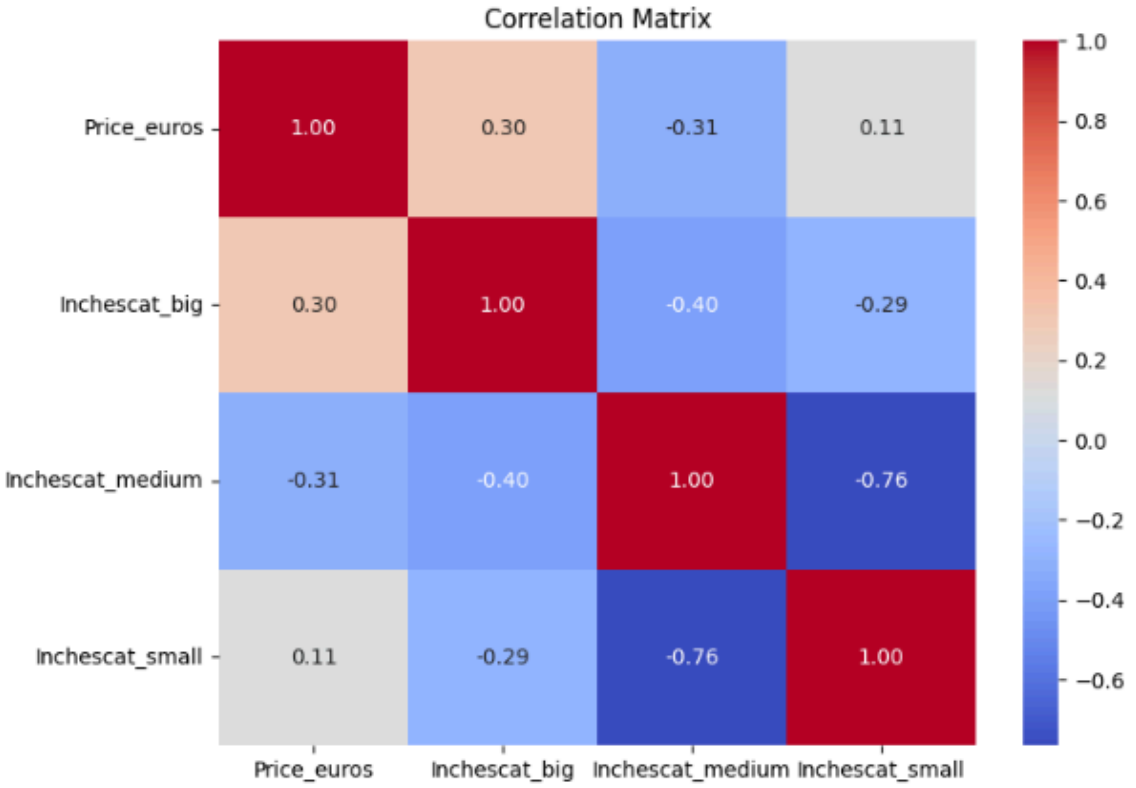
Εικονα 173 : γραμμική παλινδρόμηση μεταξύ τιμής και μεγέθους οθόνης

```
laptopd['Inchescat']='a'
laptopd.loc[(laptopd['Inches'] >= 10) & (laptopd['Inches'] <= 14.1), 'Inchescat'] = 'small'
laptopd.loc[(laptopd['Inches'] >= 14.2) & (laptopd['Inches'] <= 15.6), 'Inchescat'] = 'medium'
laptopd.loc[laptopd['Inches'] > 15.6, 'Inchescat'] = 'big'
```



Inchescat	Max Price	Average Price	Min Price	Count
big	6099.0	1672.9	379.0	166
small	3499.0	1240.7	174.0	454
medium	4899.0	925.3	199.0	655

```
dummy_columns = pd.get_dummies(laptopd['Inchescat'], prefix='Inchescat', dtype=int)
laptopd = pd.concat([laptopd, dummy_columns], axis=1)
laptopd.head()
```



Εικονες 174:heatmap pivot table και countplot του μεγέθους της οθόνης

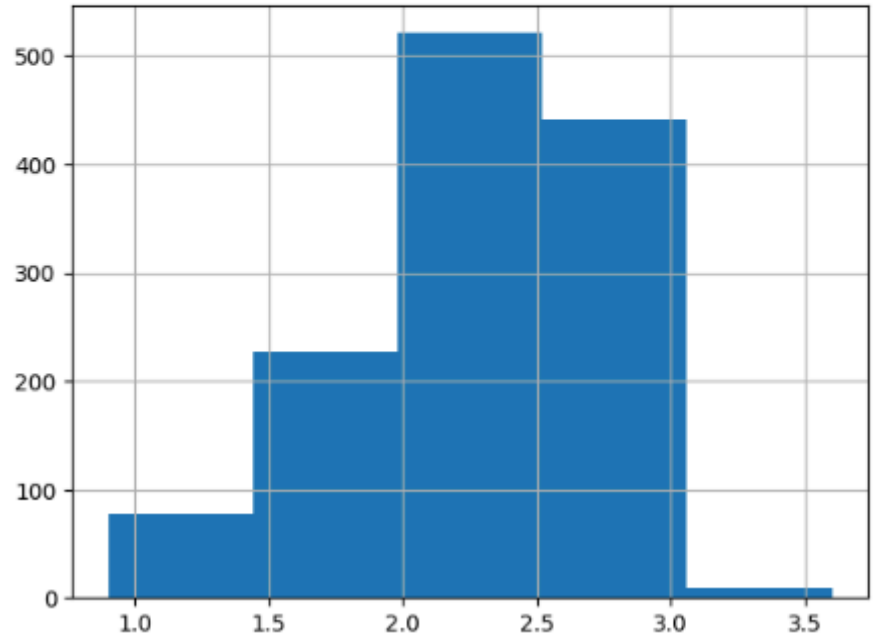
Η επόμενη αριθμητική μεταβλητή είναι η συχνότητα των επεξεργαστών του εκάστοτε λαπτοπ μετρημένη σε ghz. Και για αυτήν την μεταβλητή δημιουργούμε εάν ιστόγραμμα για να την διαχωρίσουμε σε τρεις κατηγορίες η πρώτη αποτελείτε από επεξεργαστές με συχνότητα από 0,9 έως 2 ghz η δεύτερη από 2,01 έως 2,5 και τρίτη από 2,51 και πάνω ghz. Αναφορικά με την παλινδρόμηση το r^2 είναι ίσο με 0,18 δηλαδή μόνο το 18% του συνόλου της μεταβλητότητας της τιμής ερμηνεύεται από την συγκεκριμένη μεταβλητή το p- value είναι ίσο με μηδέν αρά η μεταβλητή είναι στατιστικά σημαντική και ο συντελεστής β1 είναι θετικός ίσος με 596 δηλαδή μια αύξηση κατά μια μονάδα ghz στην συχνότητα του επεξεργαστή αυξάνει την τιμή κατά 596 ευρώ , για αυτό το λόγο και τα gaming laptop είναι πιο ακριβά από τα υπόλοιπα.

```
unique_ram_values = laptop['CPU_freq'].unique()
unique_ram_values.sort()
print(unique_ram_values)
```

```
[0.9  1.   1.1  1.2  1.3  1.44 1.5  1.6  1.8  1.9  1.92 2.   2.1  2.2
 2.3  2.4  2.5  2.6  2.7  2.8  2.9  3.   3.1  3.2  3.6 ]
```

```
laptop['CPU_freq'].hist(bins=5)
```

```
<Axes: >
```

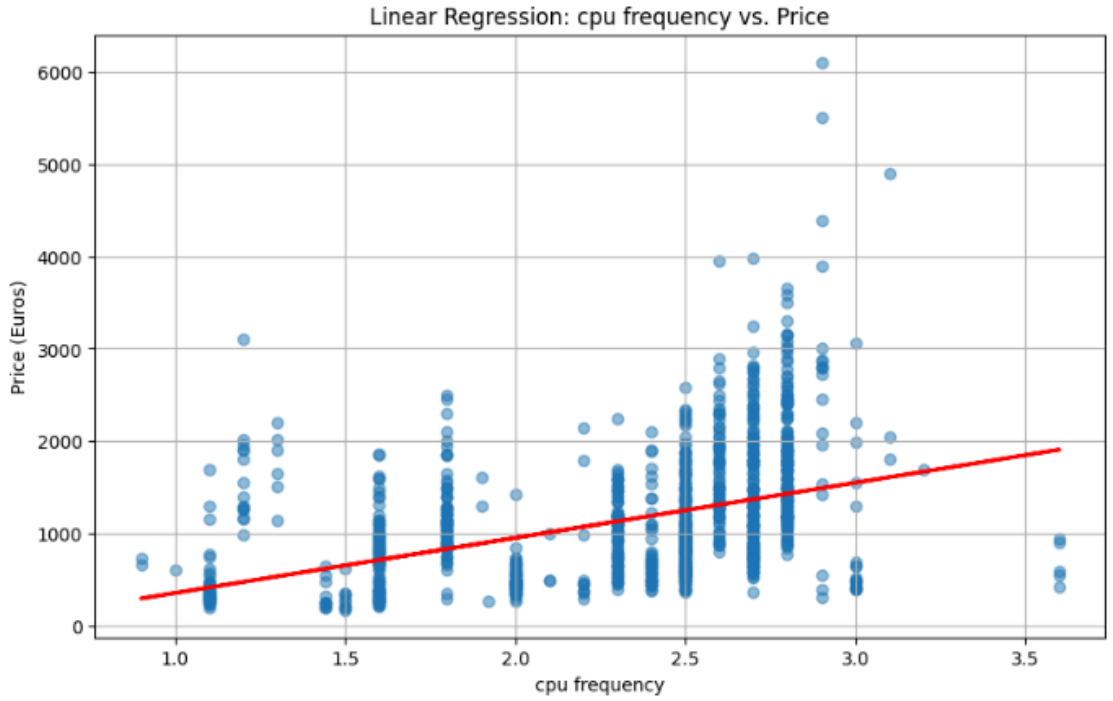


OLS Regression Results

Dep. Variable:	Price_euros	R-squared:	0.184
Model:	OLS	Adj. R-squared:	0.183
Method:	Least Squares	F-statistic:	286.9
Date:	Thu, 16 Jan 2025	Prob (F-statistic):	3.44e-58
Time:	19:14:57	Log-Likelihood:	-10033.
No. Observations:	1275	AIC:	2.007e+04
Df Residuals:	1273	BIC:	2.008e+04
Df Model:	1		
Covariance Type:	nonrobust		

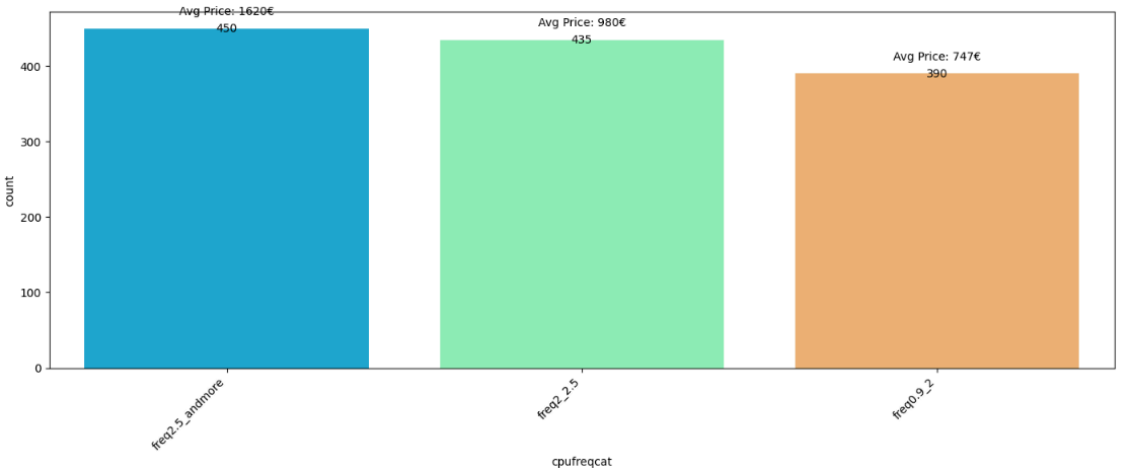
	coef	std err	t	P>|t|	[0.025	0.975]
const	-238.6299	83.015	-2.875	0.004	-401.491	-75.768
CPU_freq	596.4441	35.215	16.937	0.000	527.359	665.529

Omnibus:	402.894	Durbin-Watson:	1.998
Prob(Omnibus):	0.000	Jarque-Bera (JB):	1609.534
Skew:	1.472	Prob(JB):	0.00
Kurtosis:	7.651	Cond. No.	12.9



Εικονα 174: γραμμική παλινδρόμηση μεταξύ τιμής και συχνότητας επεξεργαστή

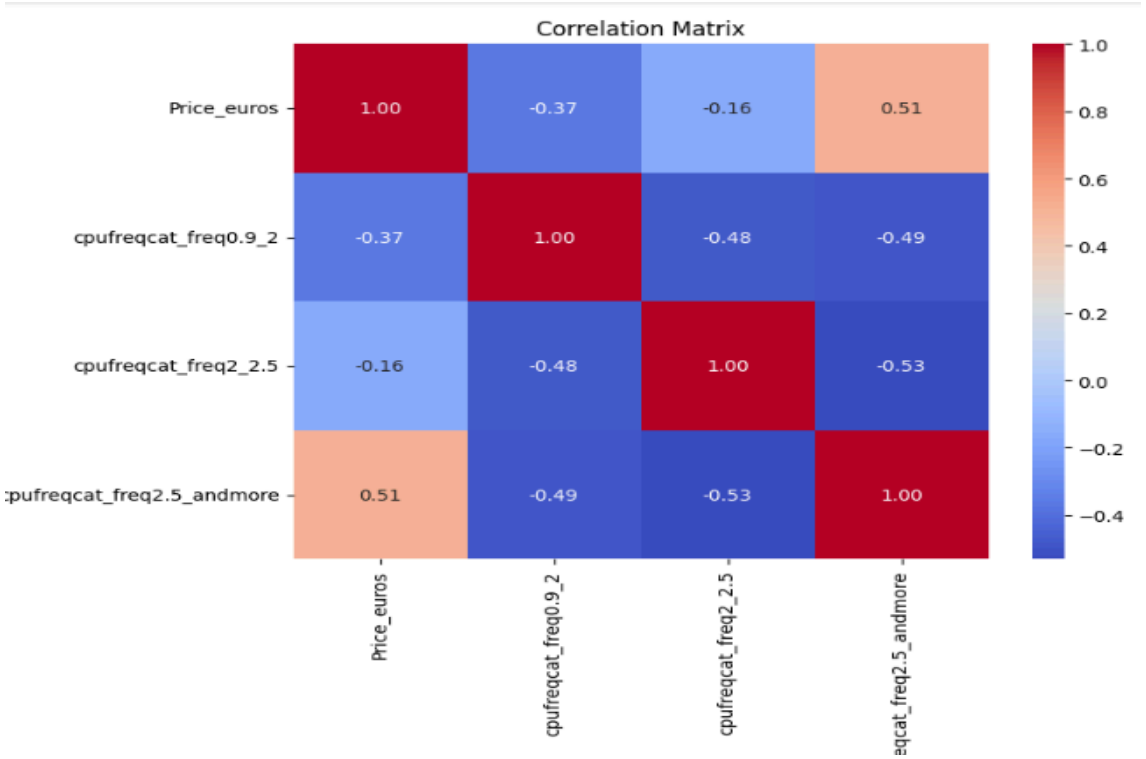
```
laptopd['cpufreqcat']='a'
laptopd.loc[(laptopd['CPU_freq'] >= 0.9) & (laptopd['CPU_freq'] <= 2), 'cpufreqcat'] = 'freq0.9_2'
laptopd.loc[(laptopd['CPU_freq'] >2) & (laptopd['CPU_freq'] <= 2.5), 'cpufreqcat'] = 'freq2_2.5'
laptopd.loc[(laptopd['CPU_freq'] >= 16) & (laptopd['CPU_freq'] <= 18), 'cpufreqcat'] = 'medium'
laptopd.loc[laptopd['CPU_freq'] > 2.5, 'cpufreqcat'] = 'freq2.5_andmore'
```



cpufreqcat	Max Price	Average Price	Min Price	Count
freq2.5_andmore	6099.0	1620.3	309.0	450
freq2_2.5	2589.0	980.0	299.0	435
freq0.9_2	3100.0	747.8	174.0	390

```
dummy_columns = pd.get_dummies(laptopd['cpufreqcat'], prefix='cpufreqcat', dtype=int)
laptopd = pd.concat([laptopd, dummy_columns], axis=1)
laptopd.head()
```

Αναφορικά με τον πίνακα συσχέτισης η τιμή έχει θετική συσχέτιση μόνο με τα λαπτοπ που ο επεξεργαστής τους είναι χρονισμένος πάνω από τα 2,5 ghz καθώς οι μικρότεροι χρονισμοί έχουν αρνητική συσχέτιση με την τιμή των λαπτοπ.



Εικονα 175: countplot pivot table , heatmap της μεταβλητής συχνότητα επεξεργαστή

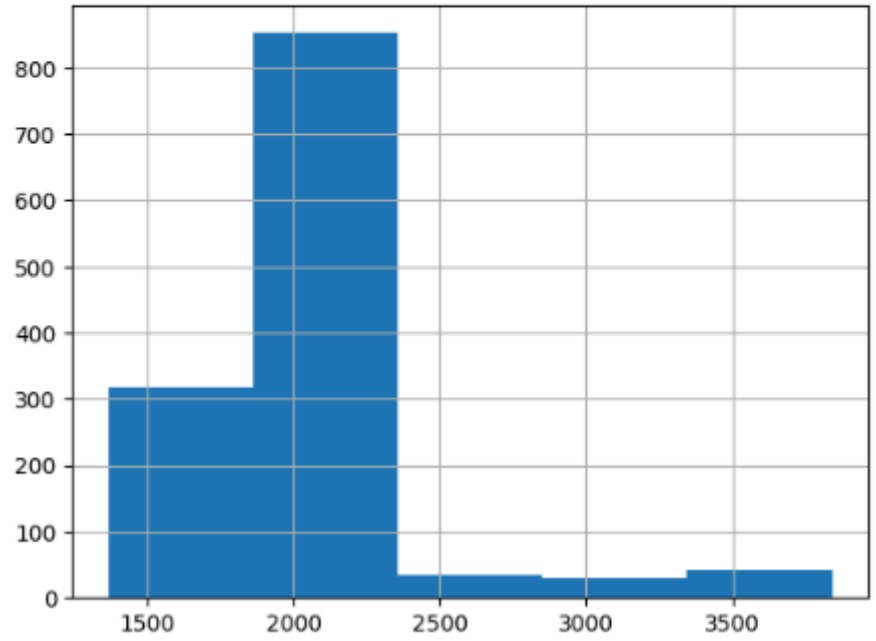
Συνεχίζουμε με την αριθμητική μεταβλητή της ανάλυσης του πλάτους της οθόνης σε pixels , και αυτή την μεταβλητή για την μετατρέψουμε σε ψευδομεταβλητες την διαχωρίζουμε σε τρεις υπό ομάδες με βάση το ιστόγραμμα στην πρώτη ομάδα ανήκουν τα λατομώ με ανάλυση πλάτους οθόνης από τα 1366 έως τα 1600 pixels η δεύτερη ομάδα αποτελείται από τα 1920 έως τα 2304 pixels και στην τρίτη ανήκουν τα λαπτοπ με μεγαλύτερη από 2304 pixels ανάλυση. Σχετικά με την γραμμική παλινδρόμηση το r^2 είναι ίσο με 0,3 δηλαδή μόνο το 30% της μεταβλητότητας της τιμής των λαπτοπ ερμηνεύεται από την ερμηνευτική μεταβλητή, Επίσης το p-value είναι ίσο με 0 αρά η μεταβλητή είναι στατιστικά σημαντική επιπλέον το β1 είναι θετικό ίσο με 0,78 δηλαδή μια αύξηση κατά ένα 1 pixel στην οριζόντια διαγώνιο αυξάνει την τιμή των λαπτοπ κατά 0,78 λεπτά του ευρώ και επίσης ο συντελεστής συσχέτισης είναι ίσος με 0,5 αρά υπάρχει ξεκάθαρη θετική συσχέτιση μεταξύ των δυο μεταβλητών.

```
unique_ram_values = laptop['ScreenW'].unique()
unique_ram_values.sort()
print(unique_ram_values)
```

```
[1366 1440 1600 1920 2160 2256 2304 2400 2560 2736 2880 3200 3840]
```

```
laptop['ScreenW'].hist(bins=5)
```

```
<Axes: >
```



OLS Regression Results

Dep. Variable:	Price_euros	R-squared:	0.305
Model:	OLS	Adj. R-squared:	0.305
Method:	Least Squares	F-statistic:	559.3
Date:	Thu, 16 Jan 2025	Prob (F-statistic):	8.55e-103
Time:	19:25:14	Log-Likelihood:	-9930.5
No. Observations:	1275	AIC:	1.986e+04
Df Residuals:	1273	BIC:	1.988e+04
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	-356.1125	65.138	-5.467	0.000	-483.902	-228.323
ScreenW	0.7848	0.033	23.650	0.000	0.720	0.850

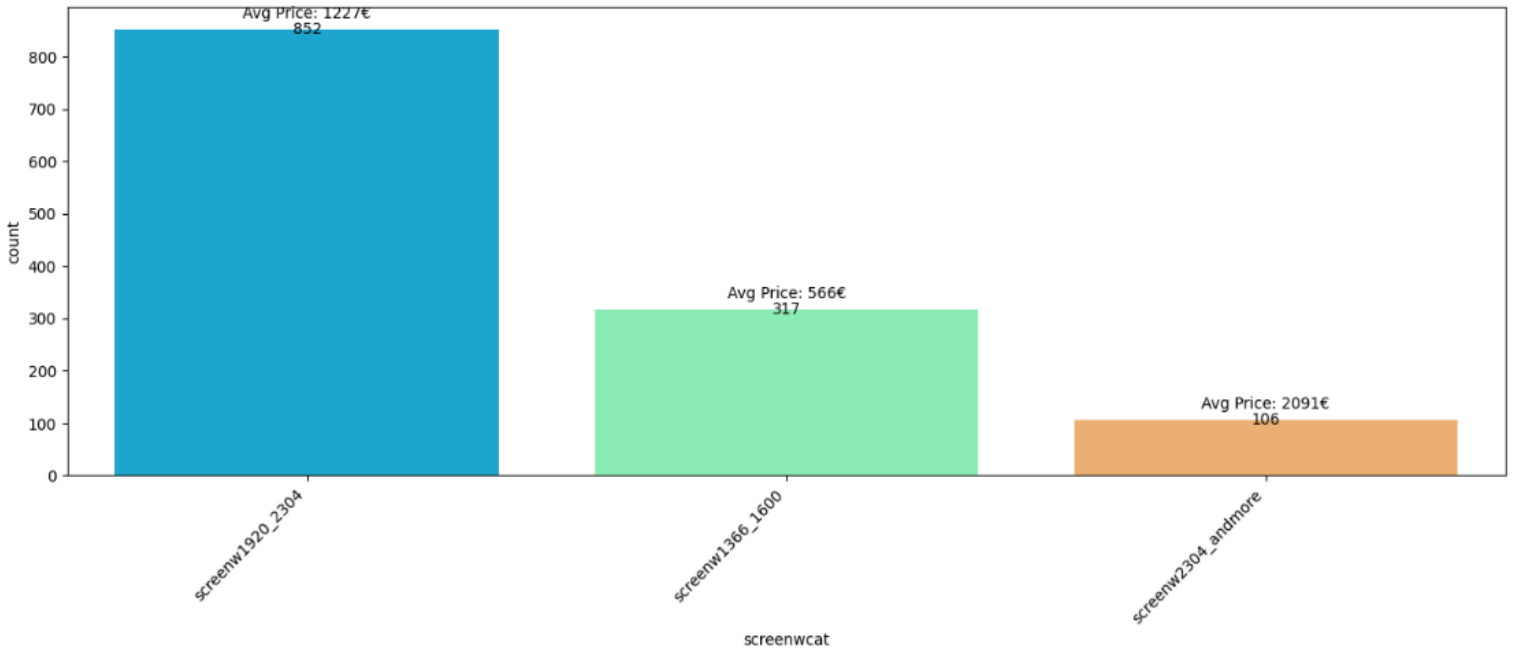
Omnibus:	361.243	Durbin-Watson:	2.019
Prob(Omnibus):	0.000	Jarque-Bera (JB):	1181.513
Skew:	1.384	Prob(JB):	2.74e-257
Kurtosis:	6.819	Cond. No.	7.81e+03


```
corralation=laptop['ScreenW'].corr(laptop['Price_euros'])
corralation
print(f"The correlation between Screen Width and Price is: {corralation}")
```

```
The correlation between Screen Width and Price is: 0.5524905658784381
```

Εικονα 176: γραμμική παλινδρόμηση μεταξύ πλάτους οθόνης και τιμής

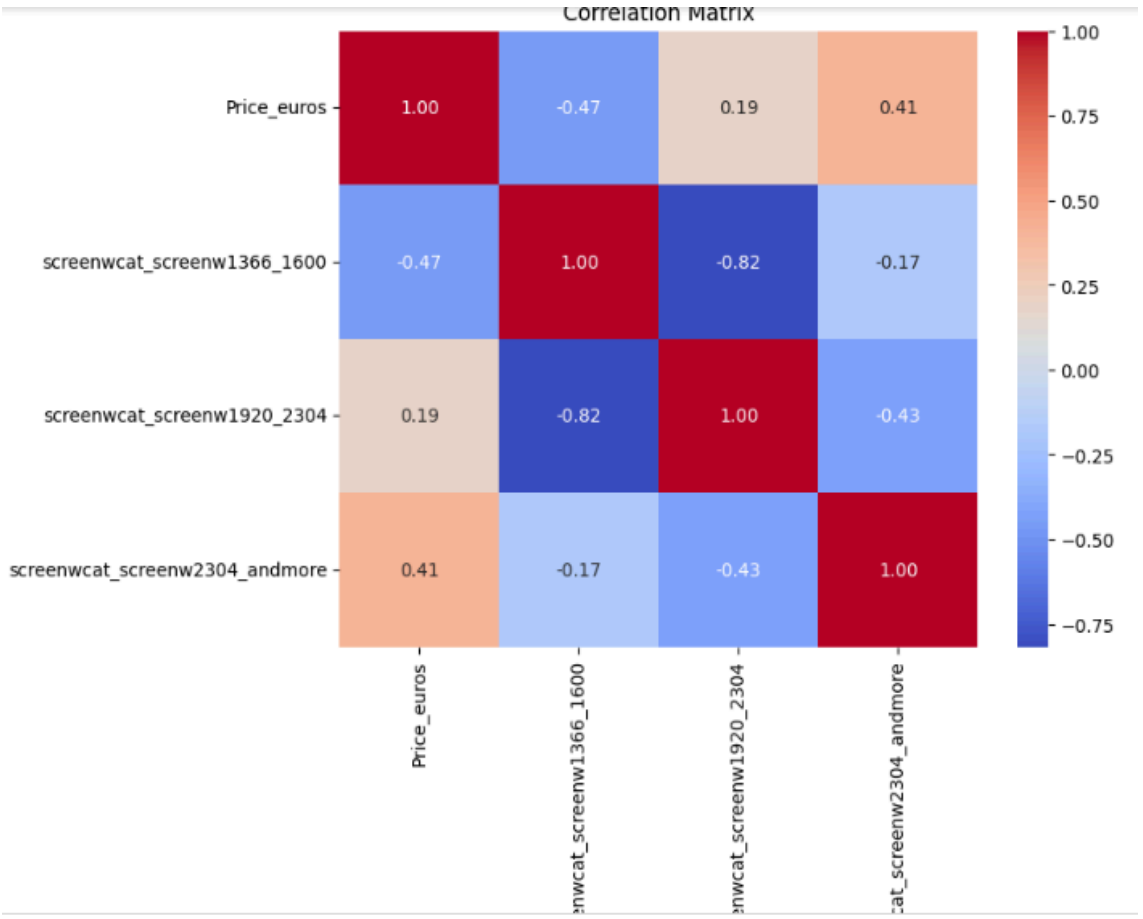
```
laptopd['screenwcat']='a'
laptopd.loc[(laptopd['ScreenW'] >= 1366) & (laptopd['ScreenW'] <= 1600), 'screenwcat'] = 'screenw1366_1600'
laptopd.loc[(laptopd['ScreenW'] >= 1920) & (laptopd['ScreenW'] <= 2304), 'screenwcat'] = 'screenw1920_2304'
laptopd.loc[laptopd['ScreenW'] > 2304, 'screenwcat'] = 'screenw2304_andmore'
```



screenwcat	Max Price	Average Price	Min Price	Count
screenw2304_andmore	6099.0	2091.6	449.0	106
screenw1920_2304	4389.0	1227.6	196.0	852
screenw1366_1600	1895.0	566.2	174.0	317

```
dummy_columns = pd.get_dummies(laptopd['screenwcat'], prefix='screenwcat', dtype=int)
laptopd = pd.concat([laptopd, dummy_columns], axis=1)

laptopd.head()
```

Όσον αφορά τον πίνακα συσχέτισης που δημιουργήθηκε με τις ψευδομεταβλητες βλέπουμε ότι τα λαπτοπ με ανάλυση από 1920 και άνω pixel στην οριζόντια διαγώνιο έχουν θετική συσχέτιση με την τιμή ενώ λαπτοπ με λιγότερα από 1920 pixel αρνητική συσχέτιση με την τιμή αρά τα καταστήματα που εμπορεύονται λαπτοπ θα περί να εστιάσουν στην πώληση λαπτοπ που έχουν υψηλή ανάλυση οθόνης.



Εικονα 177: countplot pivot table , heatmap της μεταβλητής πλατους οθονης

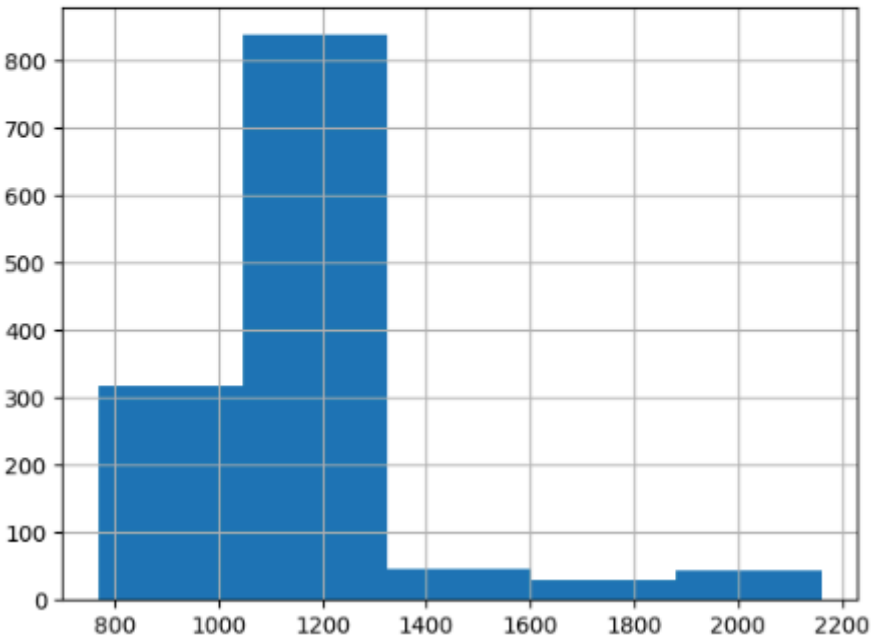
Την ιδιά ακριβός μεθοδολογία ακολουθούμε και για την μεταβλητή της ανάλυσης της οθόνης κατά ύψος μετρημένη σε pixels. Για άλλη μια φορά δημιουργούμε το ιστόγραμμα και διαχωρίζουμε τις αριθμητικές τιμές της μεταβλητής σε τέσσερις ποιοτικές τιμές πιο συγκεκριμένα η πρώτη ομάδα αποτελείται από λαπτοπ με ανάλυση από 768 έως 900 pixels στην καθετή διαγώνιο της οθόνης η δεύτερη ομάδα είναι αποτελείται από λαπτοπ με ανάλυση από 1080 έως 1200 pixels η τρίτη από 1440 έως 1600 pixels και η τελευταία περιλαβαίνει τα λαπτοπ με ανάλυση άνω των 1600 pixels. Και αυτή η μεταβλητή όπως και η προηγουμένη έχει r^2 ίσο με 0,3 και το p-value της μεταβλητής είναι ίσο με μηδέν ο συντελεστής β1 είναι ίσος με 1,35 δηλαδή μια αύξηση της ανάλυσης κατά ένα pixel στην καθετή διαγώνιο αυξάνει την τιμή του λαπτοπ κατά 1,35 ευρώ δηλαδή υπάρχει θετική συσχέτιση μεταξύ της ερμηνευτικής και της εξαρτημένης μεταβλητής η όποια ερμηνεύεται και από το συντελεστή συσχέτισης οπού είναι ίσος με 0,548.

```
unique_ram_values = laptop['ScreenH'].unique()
unique_ram_values.sort()
print(unique_ram_values)
```

```
[ 768  900 1080 1200 1440 1504 1600 1800 1824 2160]
```

```
laptop['ScreenH'].hist(bins=5)
```

```
<Axes: >
```

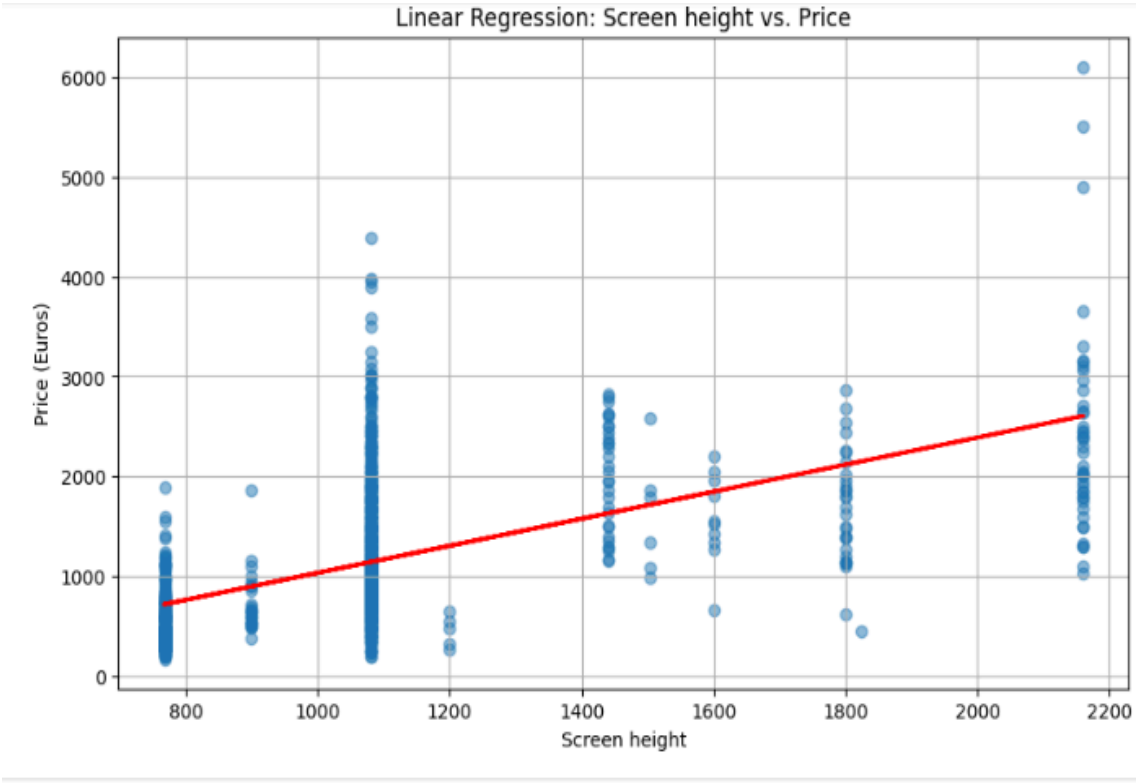


OLS Regression Results

Dep. Variable:	Price_euros	R-squared:	0.301
Model:	OLS	Adj. R-squared:	0.300
Method:	Least Squares	F-statistic:	547.9
Date:	Thu, 16 Jan 2025	Prob (F-statistic):	4.62e-101
Time:	19:29:28	Log-Likelihood:	-9934.5
No. Observations:	1275	AIC:	1.987e+04
Df Residuals:	1273	BIC:	1.988e+04
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	-319.1137	64.255	-4.966	0.000	-445.171	-193.057
ScreenH	1.3540	0.058	23.407	0.000	1.241	1.468

Omnibus:	367.892	Durbin-Watson:	2.010
Prob(Omnibus):	0.000	Jarque-Bera (JB):	1229.454
Skew:	1.402	Prob(JB):	1.07e-267
Kurtosis:	6.909	Cond. No.	4.35e+03

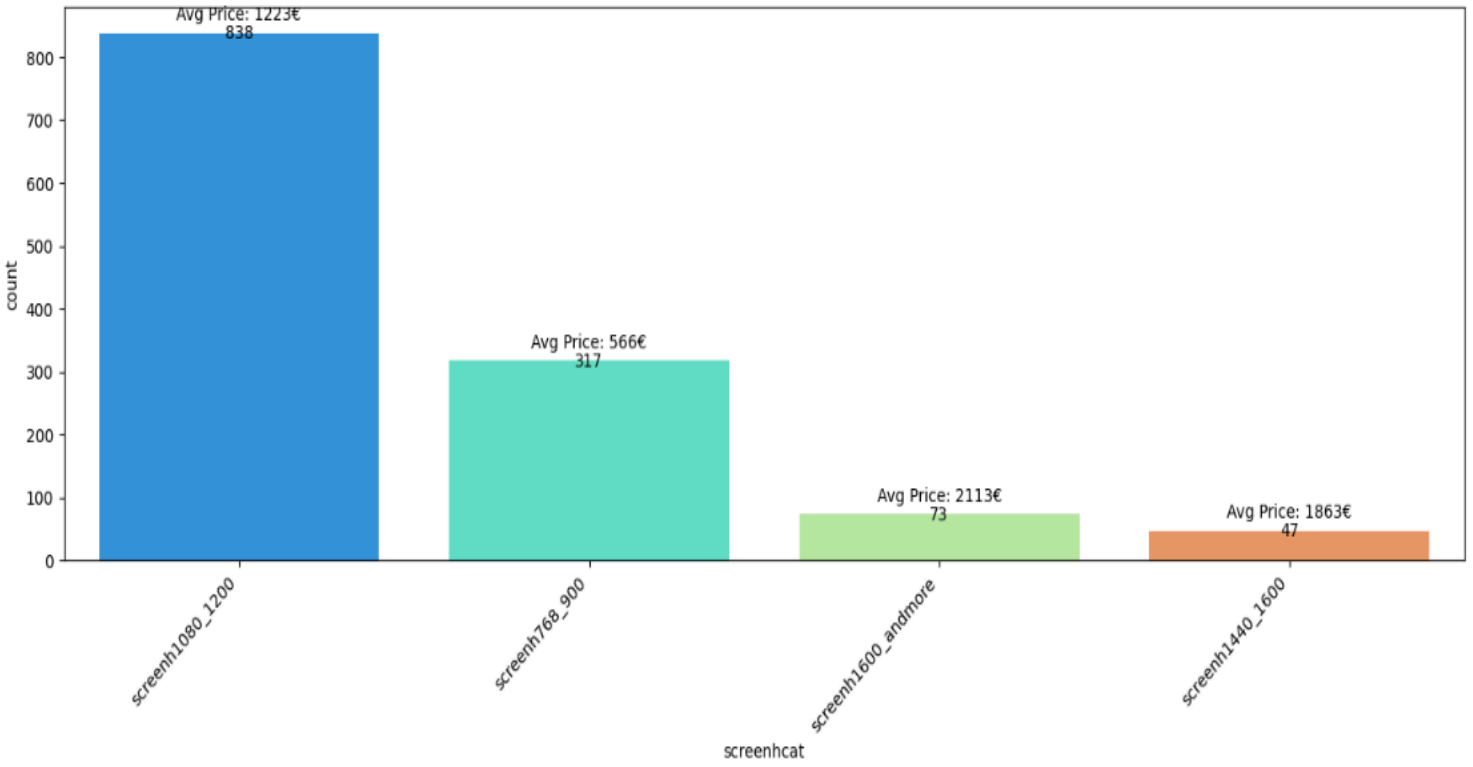



```
corralation=laptop['ScreenH'].corr(laptop['Price_euros'])
corralation
print(f"The correlation between Screen height and Price is: {corralation}")
```

```
The correlation between Screen height and Price is: 0.5485291571162705
```

Εικονα 178: γραμμική παλινδρόμηση μεταξύ ύψους οθόνης και τιμής

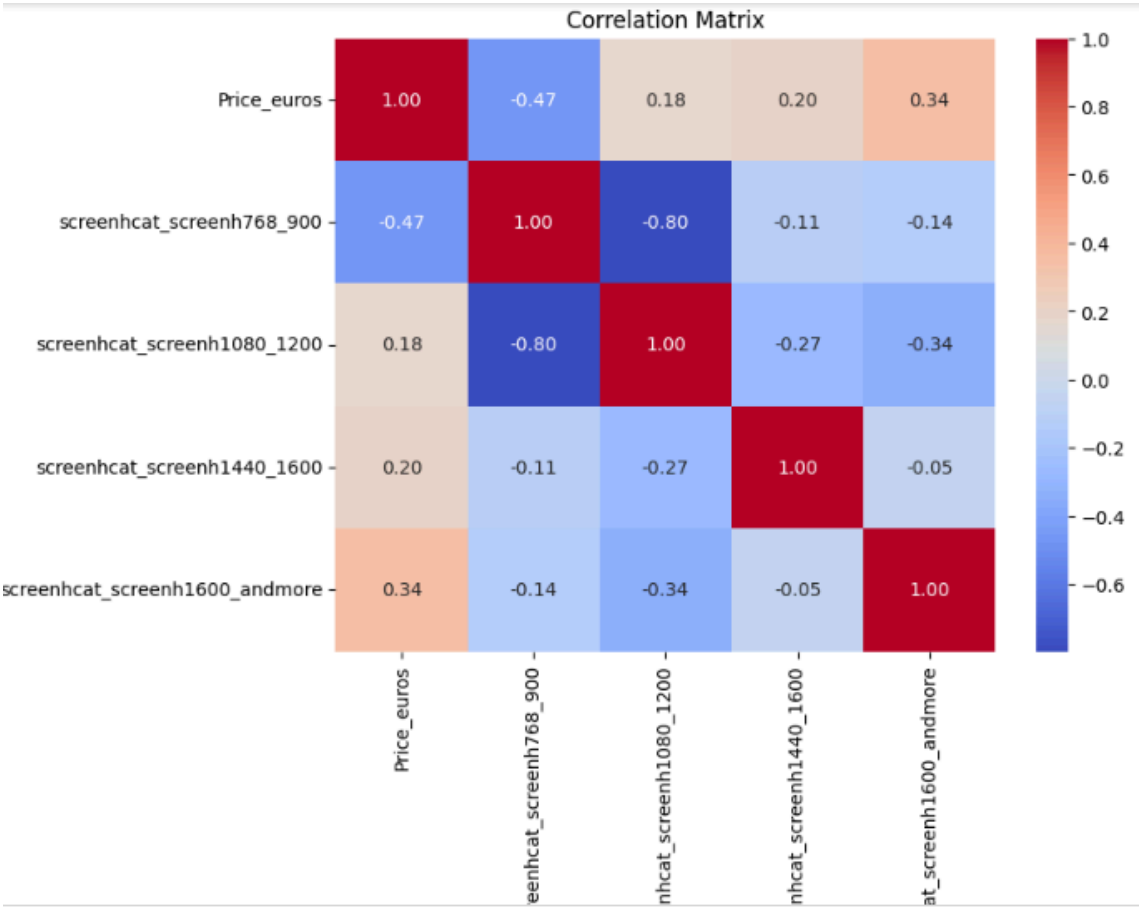
```
laptopd['screenhcat']='a'
laptopd.loc[(laptopd['ScreenH'] >= 768) & (laptopd['ScreenH'] <= 900), 'screenhcat'] = 'screenh768_900'
laptopd.loc[(laptopd['ScreenH'] >= 1080) & (laptopd['ScreenH'] <= 1200), 'screenhcat'] = 'screenh1080_1200'
laptopd.loc[(laptopd['ScreenH'] >= 1440) & (laptopd['ScreenH'] <= 1600), 'screenhcat'] = 'screenh1440_1600'
laptopd.loc[laptopd['ScreenH'] > 1600, 'screenhcat'] = 'screenh1600_andmore'
```



screenhcat	Max Price	Average Price	Min Price	Count
screenh1600_andmore	6099.0	2114.0	449.0	73
screenh1440_1600	2824.0	1863.7	659.0	47
screenh1080_1200	4389.0	1224.0	196.0	838
screenh768_900	1895.0	566.2	174.0	317

```
dummy_columns = pd.get_dummies(laptopd['screenhcat'], prefix='screenhcat', dtype=int)

laptopd = pd.concat([laptopd, dummy_columns], axis=1)

laptopd.head()
```

Αναφορικά με τον πίνακα συσχετίσεων παρατηρούμε ότι υπάρχει θετική συσχέτιση με την τιμή των λαπτοπ όταν η ανάλυση της οθόνης στην καθετή διαγώνιο είναι από 1080 pixels και πάνω ενώ αντιθέτως αν είναι μικρότερη τότε η τιμή έχει αρνητική συσχέτιση με την τιμή , το ίδιο ακριβός φαινόμενο παρατηρήσαμε και με την μεταβλητή της ανάλυσης της οθόνης στην οριζόντια διαγώνιο οπού η τιμή αυξανόταν όταν η ανάλυση ήταν και αυτή υψηλή.



Εικονα 179: countplot pivot table , heatmap της μεταβλητής ύψους οθόνης

Οι δυο τελευταίες αριθμητικές μεταβλητές οπού θα αναλύσουμε και θα μετατρέψουμε σε ποιοτικές και κατά επέκταση σε ψευδομεταβλητες είναι ο πρωτεύον και ο δευτερεύον αποθηκευτικός χώρος των λαπτοπ μετρημένα σε gb. Θα ξεκινήσουμε με τον πρωτεύον αποθηκευτικό χώρο (να αναφέρουμε πριν ξεκινήσουμε ότι η συγκεκριμένη μεταβλητή αφορά αποκλειστικά το μέγεθος της μνήμης του υπολογιστή όχι τον τύπο και την ταχύτητα προσπέλασης των δεδομένων του σκληρού δίσκου τα συγκεκριμένα δεδομένά αναλυθήκαν στην μεταβλητή primarystoragetype και secondarystoragetype αντίστοιχα) για τον οποίο δημιουργούμε ένα ιστόγραμμα για να μας διευκολύνει στον διαχωρισμό των ψευδομεταβλητων σε τέσσερεις κατηγορίες στην πρώτη κατηγορία μπαίνουν τα λαπτοπ με πρωτεύον αποθηκευτικό χώρο από 8 έως 128 gb στην δεύτερη κατηγορία τα λαπτοπ με αποθηκευτικό χώρο από 180 έως 256 gb στην τρίτη αυτά που έχουν από 500 έως 512 gb και τέλος στην τέταρτη κατηγορία ανήκουν τα λαπτοπ με αποθηκευτικό χώρο μεγαλύτερο των 512 gb. Αναφορικά με την γραμμική παλινδρόμηση το r^2 είναι κοντά στο 0 δηλαδή η συγκεκριμένη μεταβλητή ερμηνεύει μόνο το 1,6% της μεταβλητότητας της τιμής των λαπτοπ. Ο συντελεστής β1 είναι σχεδόν 0 δηλαδή αύξηση του αποθηκευτικού χώρου δεν

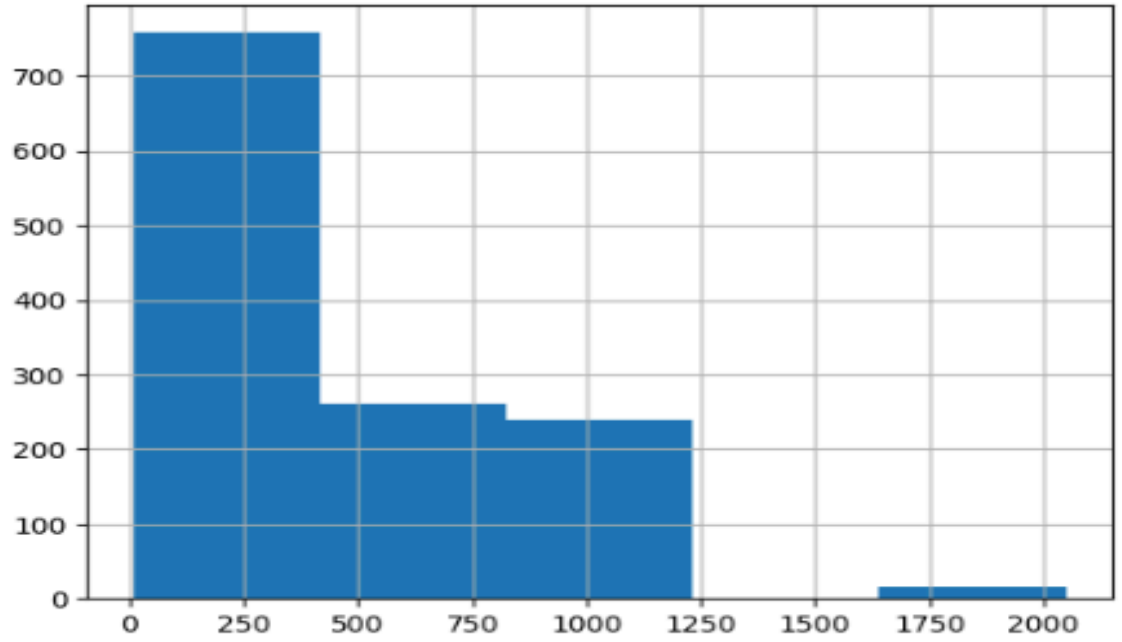
οδηγεί σε αύξηση της τιμής, αρά αυτό που μπορούμε να συμπεράνουμε με βάση την παρακάτω ανάλυση είναι ότι ίσως σημαντικότερο ρολό στην αύξηση της τιμής <<παίζει>> ο τύπος και η ταχύτητα του αποθηκευτικού χώρου και όχι το μέγεθος του καθώς ο χώρος μπορεί να αυξηθεί τουλάχιστον στους υπολογιστές με windows με την αγορά ενός εξωτερικού δίσκου (δηλαδή υπάρχουν υποκατάστατα ωθούν την τιμή του ενσωματωμένου αποθηκευτικού χώρου προς τα κάτω, επίσης σε αρκετά λαπτοπ μπορεί να αφαιρεθεί ο σκληρός δίσκος και να αντικατασταθεί με έναν SSD).

```
unique_ram_values = laptop['PrimaryStorage'].unique()
unique_ram_values.sort()
print(unique_ram_values)
```

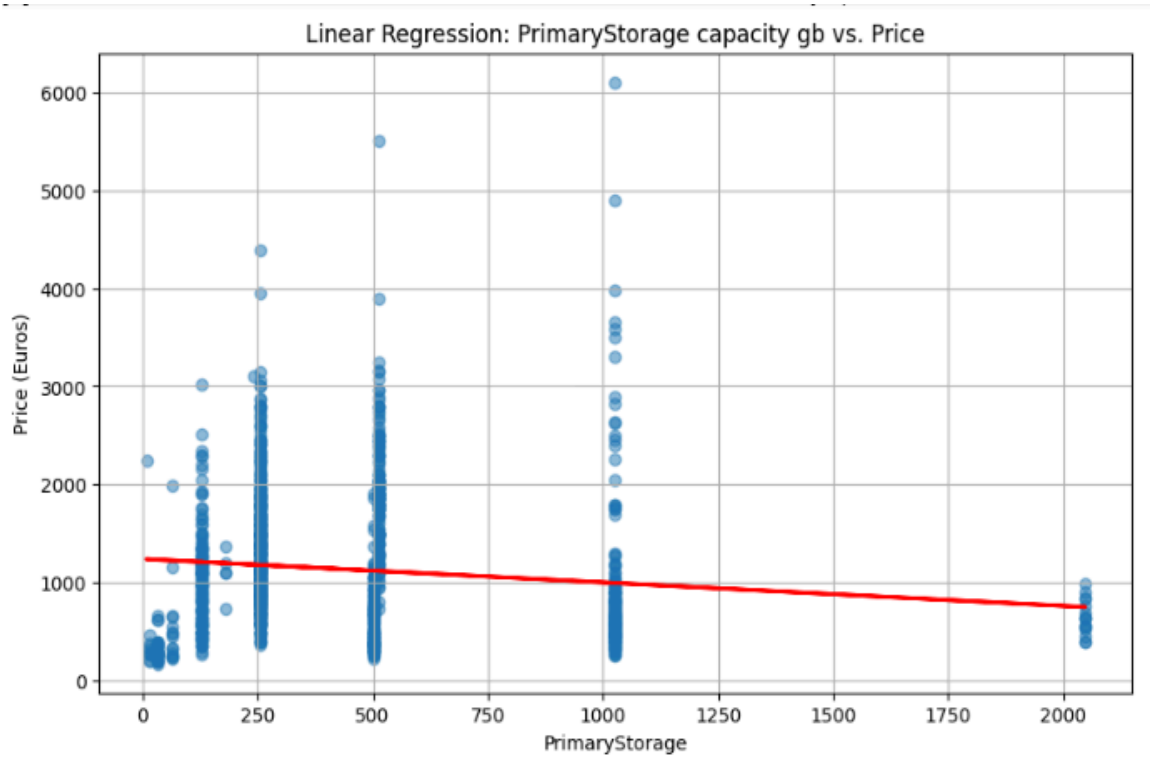
```
[   8   16   32   64  128  180  240  256  500  508  512 1024 2048]
```

```
laptop['PrimaryStorage'].hist(bins=5)
```

```
<Axes: >
```



```
                            OLS Regression Results
==============================================================================
Dep. Variable:            Price_euros   R-squared:                       0.016
Model:                            OLS   Adj. R-squared:                  0.015
Method:                 Least Squares   F-statistic:                     20.13
Date:                Thu, 16 Jan 2025   Prob (F-statistic):           7.88e-06
Time:                        20:06:31   Log-Likelihood:                -10153.
No. Observations:                1275   AIC:                         2.031e+04
Df Residuals:                    1273   BIC:                         2.032e+04
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const       1241.2975     30.676     40.465      0.000    1181.117    1301.478
PrimaryStorage    -0.2392      0.053     -4.487      0.000      -0.344      -0.135
==============================================================================
Omnibus:                      440.203   Durbin-Watson:                   2.037
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             1885.821
Skew:                           1.595   Prob(JB):                         0.00
Kurtosis:                       8.032   Cond. No.                         906.
==============================================================================
```

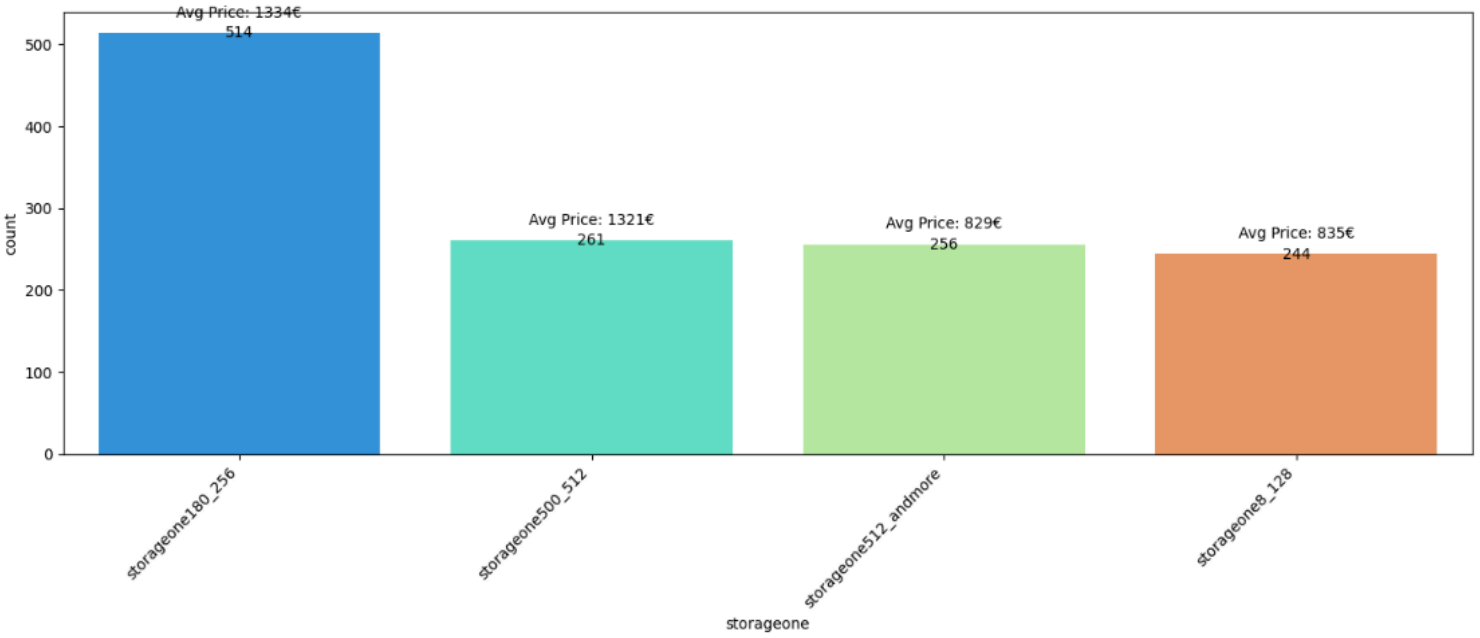



```
corralation=laptop['PrimaryStorage'].corr(laptop['Price_euros'])
corralation
print(f"The correlation between PrimaryStorage capacity gb and Price is: {corralation}")
```

```
The correlation between PrimaryStorage capacity gb and Price is: -0.12477515194071405
```

Εικονα 180:Γραμμικη παλινδρόμηση μεταξύ τιμής και primary storage capacity

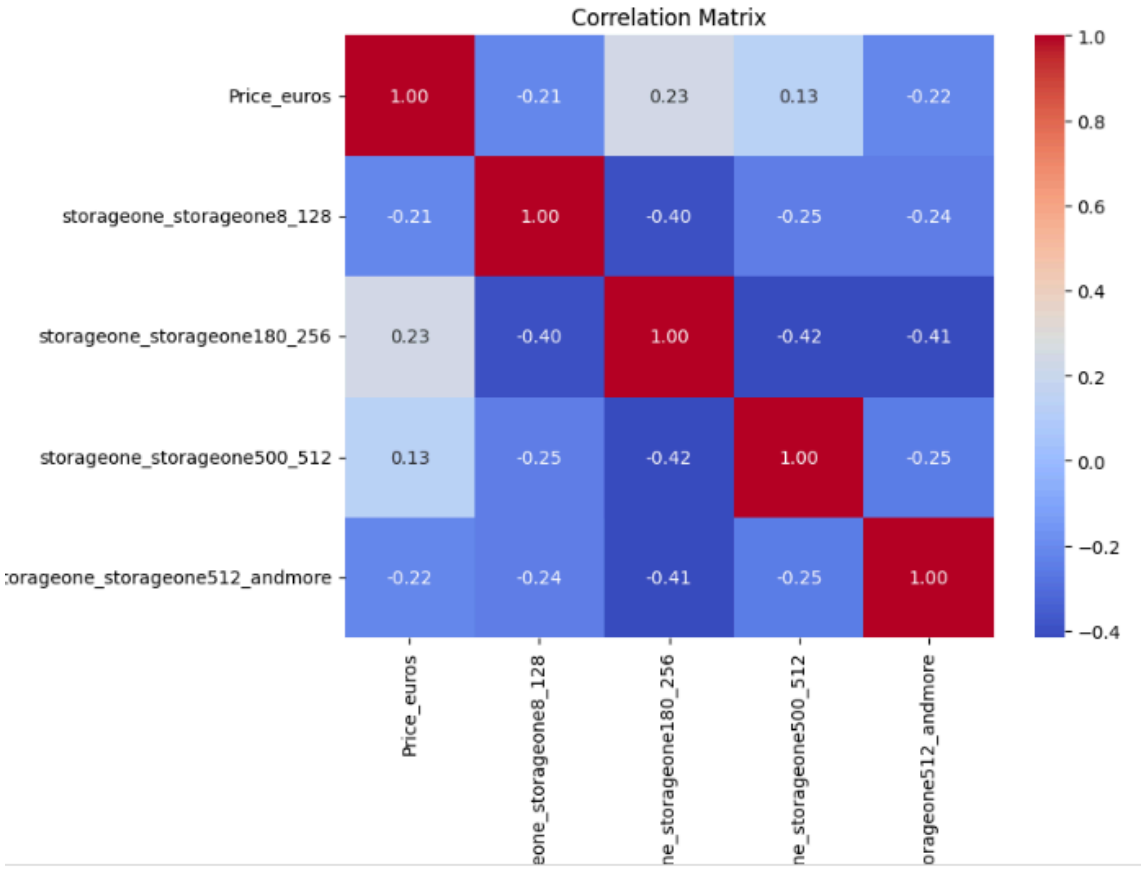
```
laptopd['storageone']='a'
laptopd.loc[(laptopd['PrimaryStorage'] >= 8) & (laptopd['PrimaryStorage'] <= 128), 'storageone'] = 'storageone8_128'
laptopd.loc[(laptopd['PrimaryStorage'] >= 180) & (laptopd['PrimaryStorage'] <= 256), 'storageone'] = 'storageone180_256'
laptopd.loc[(laptopd['PrimaryStorage'] >= 500) & (laptopd['PrimaryStorage'] <= 512), 'storageone'] = 'storageone500_512'
laptopd.loc[laptopd['PrimaryStorage'] > 512, 'storageone'] = 'storageone512_andmore'
```



storageone	Max Price	Average Price	Min Price	Count
storageone180_256	4389.0	1334.5	369.0	514
storageone500_512	5499.0	1321.2	224.0	261
storageone8_128	3012.8	835.8	174.0	244
storageone512_andmore	6099.0	829.7	252.4	256

```
dummy_columns = pd.get_dummies(laptopd['storageone'], prefix='storageone', dtype=int)
laptopd = pd.concat([laptopd, dummy_columns], axis=1)
```

Αναφορικά με τον πίνακα συσχετίσεων των ψευδομεταβλητων παρατηρούμε ότι υπάρχει θετική συσχέτιση με την τιμή όταν ο πρωτεύον αποθηκευτικός χώρος είναι μεταξύ των 180 και 512 gb ενώ αντιθέτως όταν ο αποθηκευτικός χώρος είναι πολύ μικρός δηλαδή από τα 8 έως τα 128 gb ή πολύ μεγάλος πάνω από 512 gb τότε η τιμή έχει αρνητική συσχέτιση.



Εικονα 181: countplot pivot table , heatmap της μεταβλητής primary storage capacity

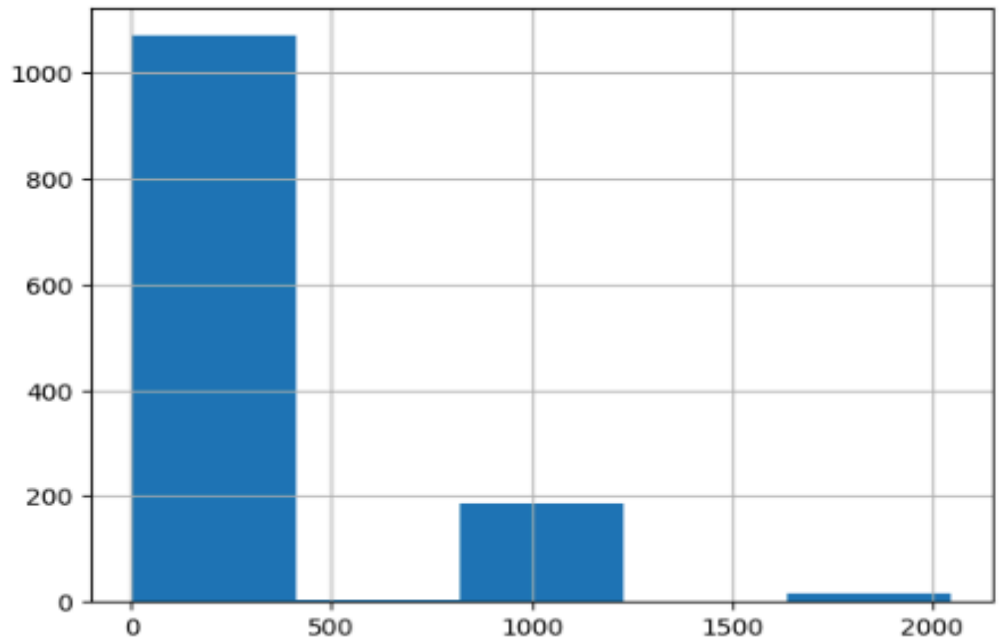
Και τέλος ολοκληρώνουμε τις μετατροπές και την ανάλυση των αριθμητικών μεταβλητών με την αριθμητική μεταβλητή του δευτερεύον αποθηκευτικού χώρου μετρημένο σε gb. Για άλλη μια φορά δημιουργούμε το ιστόγραμμα και εντάσσουμε τις τιμές της μεταβλητής σε τρεις κατηγορίες η πρώτη είναι τα λαπτοπ που έχουν δευτερεύον αποθηκευτικό χώρο ίσο με μηδέν στην δεύτερη κατηγορία ανήκουν τα λαπτοπ με αποθηκευτικό χώρο από 256 έως 512 gb και η τρίτη περιλαμβάνει τα λαπτοπ με αποθηκευτικό χώρο μεγαλύτερο από 512 gb. Στην συνεχεία παρατηρούμε τα αποτελέσματα της γραμμικής παλινδρόμησης οπού και αυτή έχει πολύ μικρό r^2 και ο συντελεστής β1 της ερμηνευτικής μεταβλητής είναι σχεδόν ίσος με μηδέν δηλαδή μια αύξηση του δευτερεύον αποθηκευτικού χώρου κατά 1 gb οδηγεί σε αύξηση μόνο 0,46 ευρώ στην τιμή των λαπτοπ. Βέβαια ο συντελεστής συσχέτισης είναι ίσος με 0,29 δηλαδή υπάρχει θετική συσχέτιση μεταξύ τιμής και του δευτερεύον αποθηκευτικού χώρου.

```
unique_ram_values = laptop['SecondaryStorage'].unique()
unique_ram_values.sort()
print(unique_ram_values)
```

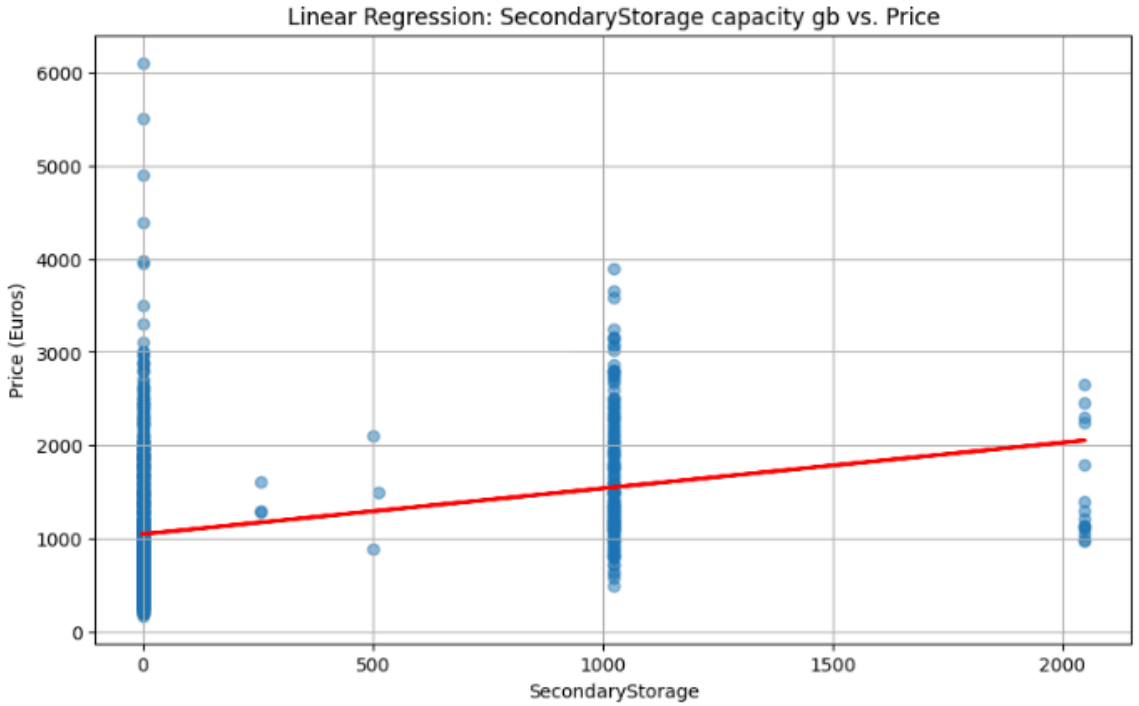
```
[   0  256  500  512 1024 2048]
```

```
laptop['SecondaryStorage'].hist(bins=5)
```

```
<Axes: >
```




```
                            OLS Regression Results
==============================================================================
Dep. Variable:            Price_euros   R-squared:                       0.085
Model:                            OLS   Adj. R-squared:                  0.084
Method:                 Least Squares   F-statistic:                     118.0
Date:                Thu, 16 Jan 2025   Prob (F-statistic):           2.43e-26
Time:                        20:15:16   Log-Likelihood:                -10106.
No. Observations:                1275   AIC:                         2.022e+04
Df Residuals:                    1273   BIC:                         2.023e+04
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                     coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const          1048.5923     20.396     51.411      0.000    1008.578    1088.606
SecondaryStorage     0.4906      0.045     10.861      0.000       0.402       0.579
==============================================================================
Omnibus:                      464.994   Durbin-Watson:                   1.971
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             2195.730
Skew:                           1.657   Prob(JB):                         0.00
Kurtosis:                       8.509   Cond. No.                         490.
==============================================================================
```

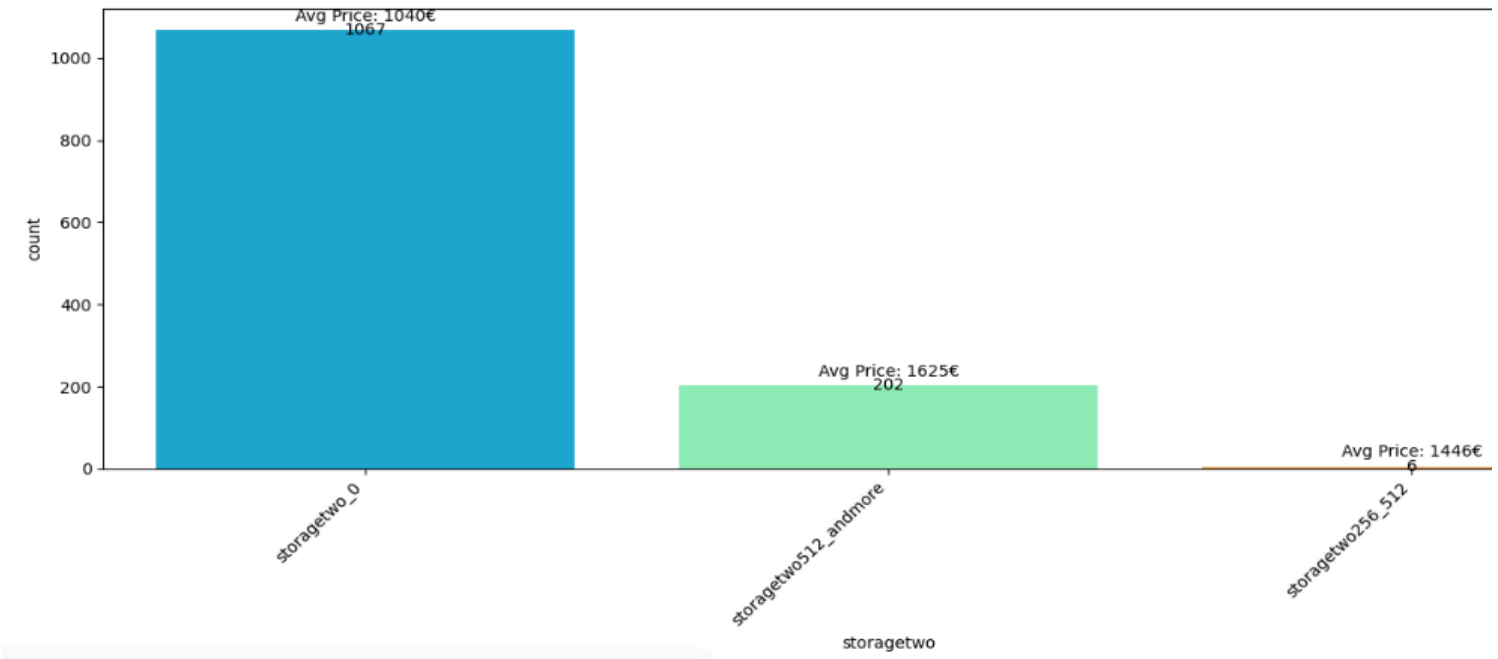



```
corralation=laptop['SecondaryStorage'].corr(laptop['Price_euros'])
print(f"The correlation between SecondaryStorage capacity gb and Price is: {corralation}")
```

```
The correlation between SecondaryStorage capacity gb and Price is: 0.29120695010730563
```

Εικονα 182: γραμμική παλινδρόμηση μεταξύ τιμής και secondary storage capacity

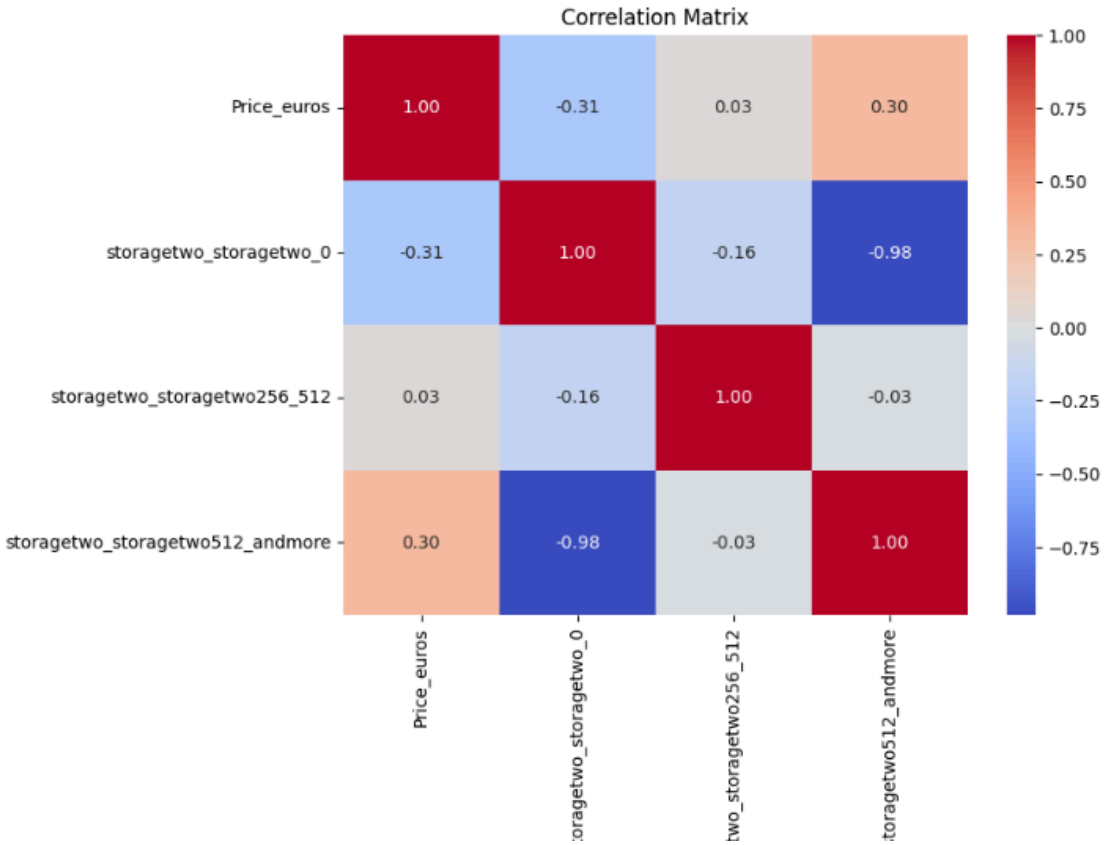
```
laptopd['storagetwo']='a'
laptopd.loc[(laptopd['SecondaryStorage'] == 0) , 'storagetwo'] = 'storagetwo_0'
laptopd.loc[(laptopd['SecondaryStorage'] >= 256) & (laptopd['SecondaryStorage'] <= 512), 'storagetwo'] = 'storagetwo256_512'
laptopd.loc[laptopd['SecondaryStorage'] > 512, 'storagetwo'] = 'storagetwo512_andmore'
```



storagetwo	Max Price	Average Price	Min Price	Count
storagetwo512_andmore	3890.0	1625.3	499.0	202
storagetwo256_512	2103.3	1446.4	891.0	6
storagetwo_0	6099.0	1040.4	174.0	1067

```
dummy_columns = pd.get_dummies(laptopd['storagetwo'], prefix='storagetwo', dtype=int)

laptopd = pd.concat([laptopd, dummy_columns], axis=1)
```

Όσον αφορά το πίνακα συσχετίσεων παρατηρούμε ότι υπάρχει αρνητική συσχέτιση μεταξύ τιμής και των λαπτοπ που δεν έχουν δεύτερο αποθηκευτικό χώρο , η τιμή είναι ανεξάρτητη με τον δευτερεύον αποθηκευτικό χώρο μεταξύ των 256 και 512 gb ενώ υπάρχει θετική συσχέτιση μεταξύ τιμής και λαπτοπ οπού έχουν αποθηκευτικό χώρο μεγαλύτερο από 512 gb.



Εικονα 183: countplot pivot table , heatmap της μεταβλητής secondary storage capacity

Έχοντας αναλύσει όλες τις μεταβλητές είτε αριθμητικές είτε ποιοτικές είμαστε σε θέση να χρησιμοποιήσουμε τις μεθόδους μηχανικής μάθησης για να μοντελοποιησουμε την τιμή των λαπτοπ. Αρχικά δημιουργούμε ένα αντίγραφο των δεδομένων οπού οι αριθμητικές μεταβλητές έχουν μετατραπεί σε ψευδομεταβλητες όπως είδαμε παραπάνω (laptopd), στην συνεχεία διαγράφουμε μια ψευδομεταβλητη από κάθε μια μεταβλητή από την όποια δημιουργήθηκε έτσι ώστε να αποφευχθεί το dummy variable trap, στην συνεχεία διαγράφουμε και τις αριθμητικές μεταβλητές τις οποίες μετατρέψαμε σε ψευδομεταβλητες.

```
laptopd2=laptopd.copy()
```

```
laptopd2.columns
```

```
Index(['Company', 'Product', 'TypeName', 'Inches', 'Ram', 'OS', 'Weight',
       'Price_euros', 'Screen', 'ScreenW', 'ScreenH', 'Touchscreen',
       'IPSpanel', 'RetinaDisplay', 'CPU_company', 'CPU_freq', 'CPU_model',
       'PrimaryStorage', 'SecondaryStorage', 'PrimaryStorageType',
       'SecondaryStorageType', 'GPU_company', 'GPU_model', 'ramcat',
       'ramcat_ram16_more', 'ramcat_ram2_7gb', 'ramcat_ram8_15gb', 'weightcat',
       'weightcat_weight0.7_1.5', 'weightcat_weight1.51_2.5',
       'weightcat_weight2.5_more', 'Inchescat', 'Inchescat_big',
       'Inchescat_medium', 'Inchescat_small', 'cpufreqcat',
       'cpufreqcat_freq0.9_2', 'cpufreqcat_freq2.5_andmore',
       'cpufreqcat_freq2_2.5', 'screenwcat', 'screenwcat_screenw1366_1600',
       'screenwcat_screenw1920_2304', 'screenwcat_screenw2304_andmore',
       'screenhcat', 'screenhcat_screenh1080_1200',
       'screenhcat_screenh1440_1600', 'screenhcat_screenh1600_andmore',
       'screenhcat_screenh768_900', 'storageone',
       'storageone_storageone180_256', 'storageone_storageone500_512',
       'storageone_storageone512_andmore', 'storageone_storageone8_128',
       'storagetwo', 'storagetwo_storagetwo256_512',
       'storagetwo_storagetwo512_andmore', 'storagetwo_storagetwo_0'],
      dtype='object')
```

```
laptopd2.drop(columns=['Product','Inches', 'Ram','Weight','ScreenW', 'ScreenH', 'CPU_freq','PrimaryStorage', 'SecondaryStorage','ramcat',
                       'ramcat_ram2_7gb','weightcat','weightcat_weight0.7_1.5','Inchescat','Inchescat_small','screenwcat',
                       'screenwcat_screenw1366_1600','screenhcat','screenhcat_screenh768_900','storageone', 'storageone_storageone8_128',
                       'storagetwo','storagetwo_storagetwo512_andmore','cpufreqcat', 'cpufreqcat_freq0.9_2'],inplace=True)
```

Οι μεταβλητές οπού θα χρησιμοποιούν είναι οι παρακάτω.

```
laptopd2.columns
```

```
Index(['Price_euros', 'ramcat_ram16_more', 'ramcat_ram8_15gb',
       'weightcat_weight1.51_2.5', 'weightcat_weight2.5_more', 'Inchescat_big',
       'Inchescat_medium', 'cpufreqcat_freq2.5_andmore',
       'cpufreqcat_freq2_2.5', 'screenwcat_screenw1920_2304',
       ...
       'GPU_model_HD Graphics 515', 'GPU_model_HD Graphics 520',
       'GPU_model_HD Graphics 615', 'GPU_model_HD Graphics 620',
       'GPU_model_Others', 'GPU_model_Radeon 520', 'GPU_model_Radeon 530',
       'GPU_model_Radeon R5 M430', 'GPU_model_Radeon R7 M445',
       'GPU_model_UHD Graphics 620'],
      dtype='object', length=111)
```

Εικονα 184: μετατροπή των δεδομένων

Στην συνεχεία όπως έχουμε είδη δει στο κεφάλαιο 6 και 7 κάνουμε import την βιβλιοθήκη sklearn οπού θα μας επιτρέψει να μοντελοποιησουμε τα δεδομένα μας με την πολυγαμική παλινδρόμηση την support vector regression και το decision tree regression, ταυτόχρονα κάνουμε import και μερικά μετρά αξιολογήσεις των μοντέλων μηχανικής μάθησης όπως το rmse το mse κλπ. Εν συνεχεία χωρίζουμε τα δεδομένα σε train και test set, πιο συγκεκριμένα τα δεδομένα για την εκπαίδευση του μοντέλου αποτελούν το 80% του συνόλου των εγγράφων ενώ το υπόλοιπο 20% των δεδομένων δια κρατούνται για τον έλεγχο της αποδοτικότητας του μοντέλου (αυτό το σύνολο των εγγράφων δεν έχουν

μοντελοποιηθει από τους αλγορίθμους μηχανικής μάθησης κατά την εκπαίδευση τους). Το random state είναι ίσο με 42 (ένα τυχαίο νούμερο) ώστε τα δεδομένα για το test και train σετ να είναι ιδιά σε όλους τους αλγορίθμους. Στην συνεχεία θέτουμε ως εξαρτημένη μεταβλητή την τιμή των λαπτοπ και ως ερμηνευτικές μεταβλητές όλες τις υπόλοιπες.

```
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
import statsmodels.api as sm

X = laptopd2.drop(columns=['Price_euros'])
y = laptopd2['Price_euros']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

X_train_sm = sm.add_constant(X_train)
model = sm.OLS(y_train, X_train_sm).fit()

print(model.summary())
X_test_sm = sm.add_constant(X_test)
y_pred = model.predict(X_test_sm)
mse = mean_squared_error(y_test, y_pred)
rmse = np.sqrt(mse)
r2 = r2_score(y_test, y_pred)
print("\nModel Evaluation Metrics:")
print(f"Mean Squared Error (MSE): {mse:.2f}")
print(f"Root Mean Squared Error (RMSE): {rmse:.2f}")
print(f"R-squared (R²): {r2:.2f}")
```

Εικονα 185: πολύ γραμμική παλινδρόμηση

Το αποτέλεσμα του αλγορίθμου οδηγεί σε ένα r^2 ίσο με 0,82 αρά οι μεταβληθέντες ερμηνεύουν το 82% της μεταβλητότητας της τιμής των λαπτοπ. Το mse του που προκύπτει μεταξύ του y_test(τα δεδομένα των τιμών των λαπτοπ που διακρατιθηκαν για την αξιολόγηση του αλγορίθμου) και του y_pred(τις τιμές των λαπτοπ που πρόβλεψε το μοντέλο με βάση τις ερμηνευτικές τιμές του x_test) είναι είναι ίσο με 117.296 ενώ το rmse είναι ίσο με 342,49 και τέλος το r^2 μεταξύ των δυο αυτών μεταβλητών είναι ίσο με 0,76. Όσον αφορά τις στατιστικά σημαντικές μεταβλητές είναι όλες εκείνες οπού έχουν μικρότερο p-value από το 0,05 αν υποθέσουμε ότι το α είναι ίσο με 0,05 δηλαδή για επίπεδο σημαντικότητας 95% μερικές από αυτές είναι η μνήμη ραμ από 16 gb και πάνω το βάρος των λαπτοπ το μέγεθος της οθόνης κλπ. Προφανώς έχοντας όλες εκείνες τις τιμές β για κάθε μια ερμηνευτική μεταβλητή ο αναλυτής μπορεί να εισάγει τις δίκες του τιμές και να πάρει ως έξοδο του αλγορίθμου την τιμή του λαπτοπ , βέβαια στην περίπτωση οπού τα νέα λαπτοπ δεν έχουν επεξεργαστές ή κάρτες γραφικών που να αντιστοιχούν με αυτές

τις ερμηνευτικές μεταβλητές ,θα πρέπει να κατασκευαστεί νέο μοντέλο ή να γίνουν προσαρμογές στο είδη υπάρχον.

```
                            OLS Regression Results
==============================================================================
Dep. Variable:            Price_euros   R-squared:                       0.821
Model:                            OLS   Adj. R-squared:                  0.803
Method:                 Least Squares   F-statistic:                     45.79
Date:                Sat, 18 Jan 2025   Prob (F-statistic):          1.19e-285
Time:                        14:33:22   Log-Likelihood:                -7249.6
No. Observations:                1020   AIC:                         1.469e+04
Df Residuals:                     926   BIC:                         1.515e+04
Df Model:                          93
Covariance Type:            nonrobust
==============================================================================================================
                                     coef    std err          t      P>|t|      [0.025      0.975]
--------------------------------------------------------------------------------------------------------------
const                          -4.945e+14   4.75e+14     -1.041      0.298   -1.43e+15    4.38e+14
ramcat_ram16_more                 524.0685     54.164      9.676      0.000     417.770     630.367
ramcat_ram8_15gb                  165.8137     33.508      4.949      0.000     100.054     231.574
weightcat_weight1.51_2.5         -120.5269     43.231     -2.788      0.005    -205.369     -35.685
weightcat_weight2.5_more         -227.2058     68.583     -3.313      0.001    -361.801     -92.610
Inchescat_big                     325.2646     64.001      5.082      0.000     199.660     450.869
Inchescat_medium                  -49.9824     39.197     -1.275      0.203    -126.907      26.942
cpufreqcat_freq2.5_andmore        513.2055     88.128      5.823      0.000     340.252     686.159
cpufreqcat_freq2_2.5              126.5588     82.415      1.536      0.125     -35.184     288.302
screenwcat_screenw1920_2304      6.297e+12   6.05e+12      1.041      0.298   -5.58e+12    1.82e+13
screenwcat_screenw2304_andmore   6.297e+12   6.05e+12      1.041      0.298   -5.58e+12    1.82e+13
screenhcat_screenh1080_1200      -6.297e+12   6.05e+12     -1.041      0.298   -1.82e+13    5.58e+12
screenhcat_screenh1440_1600      -6.297e+12   6.05e+12     -1.041      0.298   -1.82e+13    5.58e+12
screenhcat_screenh1600_andmore   -6.297e+12   6.05e+12     -1.041      0.298   -1.82e+13    5.58e+12
storageone_storageone180_256      135.2557     37.621      3.595      0.000      61.423     209.089
storageone_storageone500_512      327.1842     53.709      6.092      0.000     221.779     432.589
storageone_storageone512_andmore  316.8175     65.768      4.817      0.000     187.746     445.889
storagetwo_storagetwo256_512      -234.0651    179.507     -1.304      0.193    -586.353     118.223
storagetwo_storagetwo_0           1.709e+14   1.64e+14      1.041      0.298   -1.51e+14    4.93e+14
Company_Acer                     -1.186e+13   1.14e+13     -1.041      0.298   -3.42e+13    1.05e+13
Company_Apple                     7.423e+13   7.13e+13      1.041      0.298   -6.58e+13    2.14e+14
Company_Asus                     -1.186e+13   1.14e+13     -1.041      0.298   -3.42e+13    1.05e+13
Company_Chuwi                    -1.186e+13   1.14e+13     -1.041      0.298   -3.42e+13    1.05e+13
Company_Dell                     -1.186e+13   1.14e+13     -1.041      0.298   -3.42e+13    1.05e+13
GPU_model_GeForce 930MX          -1.24e+13   1.19e+13     -1.041      0.298   -3.58e+13    1.1e+13
GPU_model_GeForce 940MX          -1.24e+13   1.19e+13     -1.041      0.298   -3.58e+13    1.1e+13
GPU_model_GeForce GTX 1050       -1.24e+13   1.19e+13     -1.041      0.298   -3.58e+13    1.1e+13
GPU_model_GeForce GTX 1050 Ti    -1.24e+13   1.19e+13     -1.041      0.298   -3.58e+13    1.1e+13
GPU_model_GeForce GTX 1060       -1.24e+13   1.19e+13     -1.041      0.298   -3.58e+13    1.1e+13
GPU_model_GeForce GTX 1070       -1.24e+13   1.19e+13     -1.041      0.298   -3.58e+13    1.1e+13
GPU_model_GeForce MX150          -1.24e+13   1.19e+13     -1.041      0.298   -3.58e+13    1.1e+13
GPU_model_HD Graphics            -1.24e+13   1.19e+13     -1.041      0.298   -3.58e+13    1.1e+13
GPU_model_HD Graphics 400        -1.24e+13   1.19e+13     -1.041      0.298   -3.58e+13    1.1e+13
GPU_model_HD Graphics 500        -1.24e+13   1.19e+13     -1.041      0.298   -3.58e+13    1.1e+13
GPU_model_HD Graphics 515        -1.24e+13   1.19e+13     -1.041      0.298   -3.58e+13    1.1e+13
GPU_model_HD Graphics 520        -1.24e+13   1.19e+13     -1.041      0.298   -3.58e+13    1.1e+13
GPU_model_HD Graphics 615        -1.24e+13   1.19e+13     -1.041      0.298   -3.58e+13    1.1e+13
GPU_model_HD Graphics 620        -1.24e+13   1.19e+13     -1.041      0.298   -3.58e+13    1.1e+13
GPU_model_Others                 -1.24e+13   1.19e+13     -1.041      0.298   -3.58e+13    1.1e+13
GPU_model_Radeon 520             -1.24e+13   1.19e+13     -1.041      0.298   -3.58e+13    1.1e+13
GPU_model_Radeon 530             -1.24e+13   1.19e+13     -1.041      0.298   -3.58e+13    1.1e+13
GPU_model_Radeon R5 M430         -1.24e+13   1.19e+13     -1.041      0.298   -3.58e+13    1.1e+13
GPU_model_Radeon R7 M445         -1.24e+13   1.19e+13     -1.041      0.298   -3.58e+13    1.1e+13
GPU_model_UHD Graphics 620       -1.24e+13   1.19e+13     -1.041      0.298   -3.58e+13    1.1e+13
==============================================================================
Omnibus:                      315.458   Durbin-Watson:                   1.875
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             3260.970
Skew:                           1.109   Prob(JB):                         0.00
Kurtosis:                      11.474   Cond. No.                     1.06e+16
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The smallest eigenvalue is 9.99e-29. This might indicate that there are
strong multicollinearity problems or that the design matrix is singular.

Model Evaluation Metrics:
Mean Squared Error (MSE): 117296.06
Root Mean Squared Error (RMSE): 342.49
R-squared (R²): 0.76
```

Εικονα 186: αποτελέσματα πολύ γραμμικής παλινδρόμησης

Ο επόμενος αλγόριθμος μηχανικής μάθησης είναι το support vector regression για άλλη μια φορά θέτουμε ως εξαρτημένη μεταβλητή την τιμή των λαπτοπ και ως ερμηνευτικές όλες τις υπόλοιπες ψευδομεταβλητες και ποιοτικές μεταβλητές. Όπως βλέπουμε ο συγκεκριμένος αλγόριθμος με Kernal=rbf c=100 και epsilon=0,1 έχει μικρότερο και mse=94797 και rmse=307,89 και μεγαλύτερο r^2=0,81 από την multi linear regression.(τα δεδομένα του τεστ και εκπαίδευσης όπως αναφέραμε είναι τα ιδιά και στους δυο αλγορίθμους μέσο του random state=42)

```
X = laptopd2.drop(columns=['Price_euros'])
y = laptopd2['Price_euros']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

scaler_X = StandardScaler()
scaler_y = StandardScaler()

X_train_scaled = scaler_X.fit_transform(X_train)
X_test_scaled = scaler_X.transform(X_test)
y_train_scaled = scaler_y.fit_transform(y_train.values.reshape(-1, 1)).ravel()
y_test_scaled = scaler_y.transform(y_test.values.reshape(-1, 1)).ravel()

svr = SVR(kernel='rbf', C=100, epsilon=0.1)
svr.fit(X_train_scaled, y_train_scaled)

y_pred_scaled = svr.predict(X_test_scaled)
y_pred = scaler_y.inverse_transform(y_pred_scaled.reshape(-1, 1)).flatten()
y_test = scaler_y.inverse_transform(y_test_scaled.reshape(-1, 1)).flatten()

mse = mean_squared_error(y_test, y_pred)
rmse = np.sqrt(mse)
r2 = r2_score(y_test, y_pred)

print("\nModel Evaluation Metrics (SVR):")
print(f"Mean Squared Error (MSE): {mse:.2f}")
print(f"Root Mean Squared Error (RMSE): {rmse:.2f}")
print(f"R-squared (R²): {r2:.2f}")
```

```
Model Evaluation Metrics (SVR):
Mean Squared Error (MSE): 94797.71
Root Mean Squared Error (RMSE): 307.89
R-squared (R²): 0.81
```

Εικονα 187: svr

Και τέλος ο αλγόριθμος οπού δίνει τα καλυτέρα αποτελέσματα στο συγκεκριμένο σύνολο δεδομένων είναι η παλινδρόμηση των τυχαίων δασών με n_estimators=100 καθώς το mse=90.669 και το rmse=301,1 είναι μικρότερα από εκείνα του svr στο ίδιο σύνολο δεδομένων (train και test) και το ίδιο ισχύει για και για το r^2=0,82 το οποίο είναι λίγο μεγαλύτερο. Επίσης βλέπουμε ότι η ποιο σημαντική ερμηνευτική μεταβλητή για την απόφαση του αλγορίθμου είναι η μεταβλητή της μνήμης ραμ παραπάνω από 16 gb.

```
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error, r2_score

X = laptopd2.drop(columns=['Price_euros'])
y = laptopd2['Price_euros']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

rf_model = RandomForestRegressor(n_estimators=100, random_state=42)
rf_model.fit(X_train, y_train)

y_pred = rf_model.predict(X_test)

mse = mean_squared_error(y_test, y_pred)
rmse = np.sqrt(mse)
r2 = r2_score(y_test, y_pred)

print("\nModel Evaluation Metrics (Random Forest):")
print(f"Mean Squared Error (MSE): {mse:.2f}")
print(f"Root Mean Squared Error (RMSE): {rmse:.2f}")
print(f"R-squared (R²): {r2:.2f}")

importances = rf_model.feature_importances_
feature_names = X.columns
feature_importance_df = pd.DataFrame({'Feature': feature_names, 'Importance': importances})
feature_importance_df = feature_importance_df.sort_values(by='Importance', ascending=False)

print("\nFeature Importances:")
print(feature_importance_df)
```

```
Model Evaluation Metrics (Random Forest):
Mean Squared Error (MSE): 90669.07
Root Mean Squared Error (RMSE): 301.11
R-squared (R²): 0.82
```

```
Feature Importances:
                  Feature  Importance
0      ramcat_ram16_more    0.356529
1       ramcat_ram8_15gb    0.159270
39     TypeName_Notebook    0.073668
77      CPU_model_Others    0.027163
5       Inchescat_medium    0.020558
..                   ...         ...
36        Company_Xiaomi    0.000011
21         Company_Chuwi    0.000008
86       GPU_company_ARM    0.000006
30      Company_Mediacom    0.000006
60  CPU_company_Samsung    0.000004

[110 rows x 2 columns]
```

Εικονα 187: random forest regressor

Τέλος ολοκληρώνουμε το κεφάλαιο της τιμολόγησης των λαπτοπ με τους τρεις αλγορίθμους μηχανικής μάθησης που είδαμε παραπάνω μόνο που αυτή την φορά χρησιμοποιούμε της αριθμητικές μεταβλητές στην αρχική τους μορφή και όχι σαν ψευδομεταβλητες, προφανώς τις ποιοτικές μεταβλητές τις μετατρέπουμε για άλλη μια φορά σε ψευδομεταβλητες (καθώς τώρα χρησιμοποιούμε το dataset laptop2). Και στην

συνεχεία διαγράφουμε μια ψευδομεταβλητη από κάθε μια ποιοτική μεταβλητή για να αποφύγουμε το dummy trap. Ο κώδικας εκτέλεσης των αλγορίθμων είναι ακριβός ίδιος με πριν οπότε απλά θα αναφερθούμε στα αποτελέσματα του καθενός.

```
laptop2=laptop.copy()
```

```
laptop2.columns
```

```
Index(['Company', 'Product', 'TypeName', 'Inches', 'Ram', 'OS', 'Weight',
       'Price_euros', 'Screen', 'ScreenW', 'ScreenH', 'Touchscreen',
       'IPSpanel', 'RetinaDisplay', 'CPU_company', 'CPU_freq', 'CPU_model',
       'PrimaryStorage', 'SecondaryStorage', 'PrimaryStorageType',
       'SecondaryStorageType', 'GPU_company', 'GPU_model'],
      dtype='object')
```

```
dummy_columns = pd.get_dummies(laptop2['Company'], prefix='Company', dtype=int)

laptop2 = pd.concat([laptop2, dummy_columns], axis=1)
```

```
top2.drop(columns=['Product','Company','OS','Screen','Touchscreen','IPSpanel','RetinaDisplay','CPU_company','CPU_model','PrimaryStorageType','SecondaryStorageType','GPU_company','GPU_model'],i
```

Εικονα 188: μετατροπή δεδομένων

Ο πρώτος αλγόριθμος μηχανικής μάθησης είναι η πολυγραμμικη παλινδρόμηση με mse=111246 rmse=335 και r^2=0,77 μεταξύ των y_test και y_pred τα όποια είναι καλυτέρα από εκείνα του αντίστοιχου αλγορίθμου με την μετατροπή των αριθμητικών μεταβλητών . όσον αφορά το r^2 στο train set αυτό είναι ίσο με 0,819 αναφορικά με την στατιστική σημαντικότητα των ερμηνευτικών μεταβλητών βλέπουμε ότι μεταβλητές όπως η μνήμη ραμ η συχνότητα του επεξεργαστή το αν το λαπτοπ είναι μάρκας apple , razer το αν είναι gaming κλπ. έχουν p-value πολύ κοντά στο μηδέν οπότε είναι στατιστικά σημαντικές.

```python
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
import statsmodels.api as sm

X = laptop2.drop(columns=['Price_euros'])
y = laptop2['Price_euros']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
X_train_sm = sm.add_constant(X_train)
model = sm.OLS(y_train, X_train_sm).fit()

print(model.summary())

X_test_sm = sm.add_constant(X_test)
y_pred = model.predict(X_test_sm)
mse = mean_squared_error(y_test, y_pred)
rmse = np.sqrt(mse)
r2 = r2_score(y_test, y_pred)
print("\nModel Evaluation Metrics:")
print(f"Mean Squared Error (MSE): {mse:.2f}")
print(f"Root Mean Squared Error (RMSE): {rmse:.2f}")
print(f"R-squared (R²): {r2:.2f}")
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:            Price_euros   R-squared:                       0.819
Model:                            OLS   Adj. R-squared:                  0.803
Method:                 Least Squares   F-statistic:                     49.75
Date:                Sat, 18 Jan 2025   Prob (F-statistic):          7.77e-290
Time:                        15:05:37   Log-Likelihood:                -7256.1
No. Observations:                1020   AIC:                         1.468e+04
Df Residuals:                     934   BIC:                         1.511e+04
Df Model:                          85
Covariance Type:            nonrobust
=========================================================================================================
                                            coef    std err          t      P>|t|      [0.025      0.975]
---------------------------------------------------------------------------------------------------------
const                                  -521.3802     98.381     -5.300      0.000    -714.454    -328.307
Inches                                   12.6333     16.569      0.762      0.446     -19.884      45.151
Ram                                      52.5781      3.113     16.890      0.000      46.469      58.687
Weight                                   58.1312     39.172      1.484      0.138     -18.745     135.007
ScreenW                                   1.1103      0.867      1.281      0.201      -0.591       2.811
ScreenH                                  -0.9187      1.518     -0.605      0.545      -3.898       2.061
CPU_freq                                368.9044     60.015      6.147      0.000     251.125     486.683
PrimaryStorage                           -0.0003      0.051     -0.007      0.995      -0.100       0.099
SecondaryStorage                          0.0271      0.103      0.264      0.792      -0.175       0.229
Company_Acer                           -232.9317     81.436     -2.860      0.004    -392.751     -73.112
Company_Apple                             4.4160     76.843      0.057      0.954    -146.389     155.221
Company_Asus                           -165.4304     78.318     -2.112      0.035    -319.130     -11.731
Company_Chuwi                          -296.3101    202.025     -1.467      0.143    -692.786     100.166
Company_Dell                           -112.9963     75.751     -1.492      0.136    -261.657      35.665
Company_Fujitsu                        -247.3919    187.322     -1.321      0.187    -615.013     120.229
Company_Google                          -34.7934    422.651     -0.082      0.934    -864.248     794.662
Company_HP                               -0.1651     75.639     -0.002      0.998    -148.607     148.276
Company_Huawei                           44.4321    410.958      0.108      0.914    -762.075     850.940
Company_LG                              585.7687    225.463      2.598      0.010     143.296    1028.241
Company_Lenovo                          -85.5997     75.017     -1.141      0.254    -232.821      61.622
Company_MSI                             -17.1940     92.506     -0.186      0.853    -198.738     164.350
Company_Mediacom                        -215.2395    206.474     -1.042      0.297    -620.445     189.966
Company_Microsoft                       -135.0199    338.513     -0.399      0.690    -799.353     529.314
Company_Razer                            912.1328    159.426      5.721      0.000     599.258    1225.007
```

```
GPU_model_GeForce GTX 1070          478.4093     91.842      5.209      0.000     298.169     658.649
GPU_model_GeForce MX150            -245.4927    113.652     -2.160      0.031    -468.535     -22.451
GPU_model_HD Graphics                -2.5382     84.542     -0.030      0.976    -168.453     163.376
GPU_model_HD Graphics 400           -98.8400    134.749     -0.734      0.463    -363.286     165.607
GPU_model_HD Graphics 500            23.7512    161.247      0.147      0.883    -292.696     340.199
GPU_model_HD Graphics 515           593.0136    121.595      4.877      0.000     354.383     831.645
GPU_model_HD Graphics 520           177.7537     73.357      2.423      0.016      33.791     321.717
GPU_model_HD Graphics 615           407.3483    120.654      3.376      0.001     170.564     644.133
GPU_model_HD Graphics 620           194.6188     69.812      2.788      0.005      57.612     331.625
GPU_model_Others                     -9.0416     34.394     -0.263      0.793     -76.540      58.457
GPU_model_Radeon 520                -266.5978     95.746     -2.784      0.005    -454.500     -78.696
GPU_model_Radeon 530                -222.9877     78.115     -2.855      0.004    -376.289     -69.686
GPU_model_Radeon R5 M430            -103.8524     88.890     -1.168      0.243    -278.300      70.595
GPU_model_Radeon R7 M445            -231.4651    105.784     -2.188      0.029    -439.067     -23.863
GPU_model_UHD Graphics 620           85.4798     91.772      0.931      0.352     -94.623     265.583
TypeName_2 in 1 Convertible          -7.0910     57.176     -0.124      0.901    -119.299     105.117
TypeName_Gaming                    -424.0940     58.791     -7.214      0.000    -539.471    -308.717
TypeName_Notebook                  -296.9008     36.061     -8.233      0.000    -367.671    -226.131
TypeName_Others                     146.7529     50.497      2.906      0.004      47.653     245.853
TypeName_Ultrabook                   59.9526     40.333      1.486      0.138     -19.202     139.107
==============================================================================
Omnibus:                      208.310   Durbin-Watson:                   1.870
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             1508.301
Skew:                           0.732   Prob(JB):                         0.00
Kurtosis:                       8.775   Cond. No.                     8.20e+16
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The smallest eigenvalue is 8.08e-25. This might indicate that there are
strong multicollinearity problems or that the design matrix is singular.

Model Evaluation Metrics:
Mean Squared Error (MSE): 112446.73
Root Mean Squared Error (RMSE): 335.33
R-squared (R²): 0.77
```

Εικονα 188:Εφαρμογη και αποτελεσματα γρμμικης παλινδρομησης

Ο αλγόριθμος που ακολουθεί είναι το support vector regression και όπως βλέπουμε οι τιμές των mse=89285 rmse=298,8 r^2=0,82 είναι καλύτερες και από τις αντίστοιχες της multi linear regression με τις αριθμητικές μεταβλητές στην αρχική τους μορφή και από αυτές του ιδίου μοντέλου με μόνο ποιοτικές μεταβλητές του συνόλου δεδομένων laptopd.

```python
from sklearn.svm import SVR
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import mean_squared_error, r2_score

X = laptop2.drop(columns=['Price_euros'])  s
y = laptop2['Price_euros']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

scaler_X = StandardScaler()
scaler_y = StandardScaler()

X_train_scaled = scaler_X.fit_transform(X_train)
X_test_scaled = scaler_X.transform(X_test)
y_train_scaled = scaler_y.fit_transform(y_train.values.reshape(-1, 1)).ravel()
y_test_scaled = scaler_y.transform(y_test.values.reshape(-1, 1)).ravel()
svr = SVR(kernel='rbf', C=100, epsilon=0.1)
svr.fit(X_train_scaled, y_train_scaled)
y_pred_scaled = svr.predict(X_test_scaled)
y_pred = scaler_y.inverse_transform(y_pred_scaled.reshape(-1, 1)).flatten()
y_test = scaler_y.inverse_transform(y_test_scaled.reshape(-1, 1)).flatten()
mse = mean_squared_error(y_test, y_pred)
rmse = np.sqrt(mse)
r2 = r2_score(y_test, y_pred)
print("\nModel Evaluation Metrics (SVR):")
print(f"Mean Squared Error (MSE): {mse:.2f}")
print(f"Root Mean Squared Error (RMSE): {rmse:.2f}")
print(f"R-squared (R²): {r2:.2f}")
```

```
Model Evaluation Metrics (SVR):
Mean Squared Error (MSE): 89285.24
Root Mean Squared Error (RMSE): 298.81
R-squared (R²): 0.82
```

Εικονα 189: svr

Και τέλος ο αλγόριθμος που δίνει τα καλυτέρα αποτελέσματα είναι η παλινδρόμηση των τυχαίων δασών με τις αριθμητικές μεταβλητές στην αρχική τους μορφή καθώς το mse=67427 και το rmse=259 είναι μικρότερα από όλους τους υπολοίπους αλγορίθμους που δοκιμαστήκαν και το ίδιο ισχύει και για το r^2=0,86 το οποίο είναι το μεγαλύτερο. Αναφορικά με τις πιο σημαντικές ερμηνευτικές μεταβλητές του μοντέλου αυτές είναι το μέγεθος της μνήμης ραμ το βάρος του λαπτοπ το μέγεθος της οθόνης το εάν είναι notebook και η συχνότητα χρονισμού του επεξεργαστή.

```
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error, r2_score

X = laptop2.drop(columns=['Price_euros'])
y = laptop2['Price_euros']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

rf_model = RandomForestRegressor(n_estimators=100, random_state=42)
rf_model.fit(X_train, y_train)

y_pred = rf_model.predict(X_test)
mse = mean_squared_error(y_test, y_pred)
rmse = np.sqrt(mse)
r2 = r2_score(y_test, y_pred)

print("\nModel Evaluation Metrics (Random Forest):")
print(f"Mean Squared Error (MSE): {mse:.2f}")
print(f"Root Mean Squared Error (RMSE): {rmse:.2f}")
print(f"R-squared (R²): {r2:.2f}")

importances = rf_model.feature_importances_
feature_names = X.columns
feature_importance_df = pd.DataFrame({'Feature': feature_names, 'Importance': importances})
feature_importance_df = feature_importance_df.sort_values(by='Importance', ascending=False)

print("\nFeature Importances:")
print(feature_importance_df)
```

```
Model Evaluation Metrics (Random Forest):
Mean Squared Error (MSE): 67427.34
Root Mean Squared Error (RMSE): 259.67
R-squared (R²): 0.86

Feature Importances:
                 Feature  Importance
1                    Ram    0.555625
2                 Weight    0.079157
97    TypeName_Notebook    0.067130
5               CPU_freq    0.052699
0                 Inches    0.025374
..                   ...         ...
45  CPU_company_Samsung    0.000007
11        Company_Chuwi    0.000003
14       Company_Google    0.000003
71      GPU_company_ARM    0.000002
20     Company_Mediacom    0.000001

[100 rows x 2 columns]
```

Εικονα 190: random forest regressor

Αρά ο αλγόριθμος που προτείνεται για την μελλοντική τιμολόγηση νέων φορητών υπολογιστών είναι αυτός των τυχαίων δασών.

7. ΕΦΑΡΜΟΓΗ ΠΡΟΒΛΕΨΗΣ ΠΩΛΗΣΕΩΝ ΚΑΙ ΚΟΣΤΟΥΣ ΠΩΛΗΘΕΝΤΩΝ ΤΗΣ ΕΤΑΙΡΙΑΣ ΠΛΑΙΣΙΟ COMPUTERS ΑΕ

7.1 ΣΥΝΤΟΜΗ ΠΕΡΙΓΡΑΦΗ ΕΤΑΙΡΙΑΣ

Το πλαίσιο είναι μια ελληνική εταιρία λιανικής πώλησης ηλεκτρικών συσκευών όπως υπολογιστές κινητά τηλεφώνα λευκές συσκευές αλλά και ειδών γραφείου που ιδρύθηκε από το Γεώργιο Γεράρδο το 1969 . Το πρώτο κατάστημα της εταιρίας είχε μέγεθος 14 τετραγωνικά μετρά και βρίσκονταν στην οδό Στουρνάρι στο κέντρο της Αθήνας . Τα πρώτα προϊόντα τα όποια πουλούσε το κατάστημα ήταν είδη γραφείου όπως για παράδειγμά ειδικοί χάρακες χρήσιμοι για φοιτητές πανεπιστήμιων θετικών επιστήμων όπως το πολύτεχνοί το οποίο βρισκόταν σε πολύ μικρή απόσταση από το κατάστημα αυξάνοντας με αυτό τον τρόπο τις αρχικές πωλήσεις του καταστήματος μέσο της γεωγραφικής γειτνίασης. Το πλαίσιο μέχρι την συγχώνευση των media markt των Saturn και των multirama με τα public ήταν ηγέτης στο κλάδο λιανικών πωλήσεων προϊόντων τεχνολογίας και σε αυτό το βοήθησαν πολλοί παράγοντες ένας από αυτούς είναι ότι διαθέτει 24 καταστήματα σε μεγάλες πόλεις της Ελλάδας όπως η Αθηνά η θεσσαλονικεί η πατρα η Λάρισα τα Γιάννενα το ηράκλειο Κρήτης και τα Χανιά εξυπηρετώντας με αυτό τον τρόπο μεγάλο πλήθος καταναλωτών αλλά μειώνοντας με αυτό τον τρόπο και τα μεταφορικά κόστη καθώς το μέγεθος του εκάστοτε καταστήματος εξυπηρετεί στο να χρησιμοποίειται το κατάστημα και ως αποθήκη για παράδειγμά μια παραγγελία στην Αλεξανδρούπολη μπορεί να εξυπηρετηθεί από τα καταστήματα στην θεσσαλονικεί και όχι απευθείας από το κέντρο διανομής 22.000 τετραγωνικών μέτρων στην μαγουλά Αττικής. Επιπλέον η εταιρία διαθέτει και ένα κατάστημα στην Σόφια την πρωτεύουσα της Βουλγαρίας αυξάνοντας με αυτό τον τρόπο της πωλήσεις του ομίλου. Ένας άλλος παράγοντας που βοήθησε στην ανάπτυξη της εταιρίας είναι το πολύπροιοντικο μοντέλο της καθώς το πλαίσιο διαθέτει πάνω από 25.000 διαφορετικά προϊόντα τεχνολογίας λευκών συσκευών και ειδών γραφείου αλλά και συνδυάζει τα παραπάνω με διάφορες συνοδευτικές υπηρεσίες όπως επέκταση εγγυήσεων των προϊόντων επισκευή προϊόντων την ασφάλιση τους αλλά ακόμα και την συναρμολόγηση desktop υπολογιστών δηλαδή ο καταναλωτής μπορεί να αγοράσει από το κατάστημα τα διαφορά μέρη ενός υπολογιστή όπως αυτά που είδαμε στο dataset price laptop του προηγουμένου κεφαλαίου δηλαδή κάρτα γραφικών επεξεργαστή κλπ. και να τα συναρμολογήσουν εντός του καταστήματος

,επιπρόσθετα σε πολλές κατηγορίες προϊόντων όπως τους φορητούς υπολογιστές και τις τηλεοράσεις διαθέτει την ιδιοκτήτη της μάρκα turbox . Ένας άλλος παράγοντας που ωθεί σε υψηλότερο επίπεδο τις πωλήσεις είναι η πελατοκεντρικη φιλοσοφία της εταιρίας , συγκεκριμένα ο όμιλός εστιάζει στην ικανοποίηση των αναγκών των πελατών το οποίο επιτυγχάνεται και από τα πολλά σημεία πώλησης,όχι μόνο από τα φυσικά καταστήματα αλλά και το διαδίκτυο και στην συνέχεια μέσο της άμεσης παράδοσης των προϊόντων στην τοποθεσία του πελάτη αλλά και μέσα από το τηλεφωνικό κέντρο στο οποίο υπάρχει τμήμα που εστιάζει αποκλειστικά και μόνο στις πωλήσεις b2b οι οποίες αποτελούν το 52,3% του συνόλου των πωλήσεων. Πέρα από τα παραπάνω η εταιρία έχει στρατηγικές και μακροχρόνιες σχέσεις με τους προμηθευτές της και πλήρως ανεπτυγμένη εφοδιαστική αλυσίδα με ιδιόκτητο στόλο αλλά και υπερσύγχρονο κέντρο διανομής στην μαγουλά , όσον αφορά τους εργαζομένους μέχρι την σύνταξη της οικονομικής έκθεσης του 2023 το πλαίσιο απασχολούσε 1531 εργαζομένους οπού το 53% αυτών είναι κάτω των 30 ετών δίνοντας με αυτό τον τρόπο ευκαιρία σε νέους ανθρώπους να ενταχθούν στον εργασιακό τομέα και να εξελιχθούν επαγγελματικά.

Στην συνεχεία θα αναλύσουμε του κίνδυνους που συντρέχει ο όμιλός. Ο πρώτος κίνδυνος που αναλύεται στην οικονομική έκθεση του 2023 είναι ο κίνδυνος επιτοκίου ο οποίος μπορεί να προκαλέσει έξοδα αξίας 140.000 ευρώ σε περίπτωση που τα επιτόκια αυξηθούν κατά 1% προς τα πάνω και 140.000 ευρώ αύξηση των ιδίων κεφαλαίων σε περίπτωση που τα επιτόκια κινηθούν μια ποσοστιαία μονάδα προς τα κάτω. Αυτό είναι αποτέλεσμα των ομολογιακών δάνειων κυμαινομένων επιτοκίων αξίας 3.500.000 από την εθνική τράπεζα και 5.000.000 ευρώ αντίστοιχου δανείου από την Eurobank. Στην συνεχεία γίνεται αναφορά στον πιστωτικό κίνδυνο δηλαδή την πιθανότητα κάποιος από τους πελάτες της εταιρίας να αθετήσει τις υποχρεώσεις του προς αυτήν δηλαδή ένας πελάτης να έχει αγοράσει προϊόντα επι πίστωση πχ λαπτοπ και την περίοδο της αποπληρωμής του χρέους του προς την εταιρία να μην έχει χρήματα(και αλλά χρεόγραφα όπως γραμμάτια εισπρακτέα) να την αποπληρώσει, η εταιρία αναφέρει ότι ο συγκεκριμένος κίνδυνος είναι μικρός καθώς έχει επιτύχει διασπορά του πελατολογίου της και πιο συγκεκριμένα οι λιανικές πωλήσεις γίνονται της μετρητοίς ή με χρήση πιστωτικής ή και χρεωστικών καρτών , όσον αφορά της b2b πωλήσεις πριν διαθέσει τα προϊόντα στους πελάτες εξετάζει την πιστοληπτική τους ικανότητα ενδελεχώς δημιουργώντας για τον καθένα σενάρια και προβλέψεις. Πιο αναλυτικά οι απαιτήσεις του ομίλου έχουν αξία ύψους 50.276.000 οπού από αυτές μόνο οι 1.894.000 είναι επισφαλής.

Αναφορικά με τον συναλλαγματικό κίνδυνο και αυτός βρίσκεται σε χαμηλά επίπεδα καθώς η εταιρία δεν έχει δανειστεί από τράπεζες σε ξένο νόμισμα ενώ ταυτόχρονά έχει μόνο 1,1 εκατομμύρια ευρώ ταμιακά διαθέσιμα σε δολάρια ενώ ταυτόχρονα το κατάστημα της στην Βουλγαρία δεν υπόκειται σε αυτόν τον κίνδυνο καθώς τα Λέβα έχουν σταθερή ισοτιμία με τα ευρώ. Εν συνεχεία και ο κίνδυνος ρευστότητας δηλαδή η πιθανότητα η εταιρία να <<ξεμείνει>> από ρευστά διαθέσιμα και κατά συνέπεια να μην μπορεί να αποπληρώσει τις υποχρεώσεις τις στις τράπεζες και τους προμηθευτές που θα οδηγούσε στον τερματίσω της λειτουργίας της εταιρία καθώς οι προμηθευτές δεν θα πουλούσαν τα εμπορεύματα τους επι πίστωση είναι μικρός και σχεδόν μηδαμινός καθώς η εταιρία διαθέτει ταμειακά διαθέσιμα που μπορούν να αποπληρώσουν το σύνολο του δανεισμού της και ταυτόχρονα λόγο της διαχρονικής μεγέθυνσης της εταιρίας έχει ως αποτέλεσμα οι τράπεζες να την εμπιστεύονται και κατά συνέπεια να έχει άμεση πρόσβαση σε ρευστότητα.

Οι δυο σημαντικότεροι κίνδυνοι οπού συντρέχει η εταιρία είναι η ένταση του ανταγωνισμού και ο κίνδυνος αποθεμάτων και προμηθευτών. Όσον αφορά τον πρώτο η εταιρία αναφέρει ότι ο ανταγωνισμός είναι πολύ υψηλός καθώς υπάρχουν πολλές μικρές εγχώριες εταιρίες στον τομέα πωλήσεων ηλεκτρικών συσκευών όπως για παράδειγμά τα κινητά τηλεφώνα που έχει ως αποτέλεσμα το περιθώριο κέρδους για την εταιρία να μειώνεται , παρόλα αυτά η εταιρία περιορίζει αυτόν τον κίνδυνο μέσο της μεγάλης γκάμας προϊόντων από ηλεκτρικές συσκευές έως είδη γραφείου αλλά και μέσο της υψηλής ρευστότητας που διαθέτει καθώς ανεξάρτητος της κίνησής της αγοράς πτωτική ή ανοδική το πλαίσιο μπορεί να προμηθευτεί νέα προϊόντα σε μεγάλες ποσότητες με αποτέλεσμα να αγοράζει σε χαμηλότερες τιμές σε σχέση με τους ανταγωνιστές τις οι οποίοι αγοράζουν σε μικρές ποσότητες και κάνουν χρήση της just in time τεχνικής έχοντας χαμηλά αποθέματα αλλά και ταυτόχρονα υψηλοτέρους χρόνους παραδόσεις των προϊόντων στους πελάτες τους ,αυτό έχει ως αποτέλεσμα διαχρονικά το πλαίσιο να επιτυγχάνει από τα μεγαλύτερα περιθώρια απόδοσης του κλάδου, προφανώς εάν η ελληνική οικονομία μπει σε κρίση στο βραχυχρόνιο μέλλον και οι καταναλωτικές δαπάνες μειωθούν αυτό θα οδηγήσει σε ανακατανομή του μεριδίου αγοράς και σε μείωση των πωλήσεων και κατά επέκταση των κερδών της εταιρίας, οπότε και ο συγκεκριμένος κίνδυνος είναι πολύ σημαντικός όπως και ο κίνδυνος αποθεμάτων και προμηθευτών. Αναφορικά με τον δεύτερο είναι και αυτός αρκετά υψηλός όχι λόγο κλοπής, ζημίων ή καταστροφής των εμπορευμάτων αλλά κυρίως λόγο του γεγονότος ότι η εταιρία εμπορεύεται προϊόντα υψηλής τεχνολογίας με αποτέλεσμα αυτά να απαξιώνονται μέσα σε μικρό χρονικό διάστημα από την ημέρα της

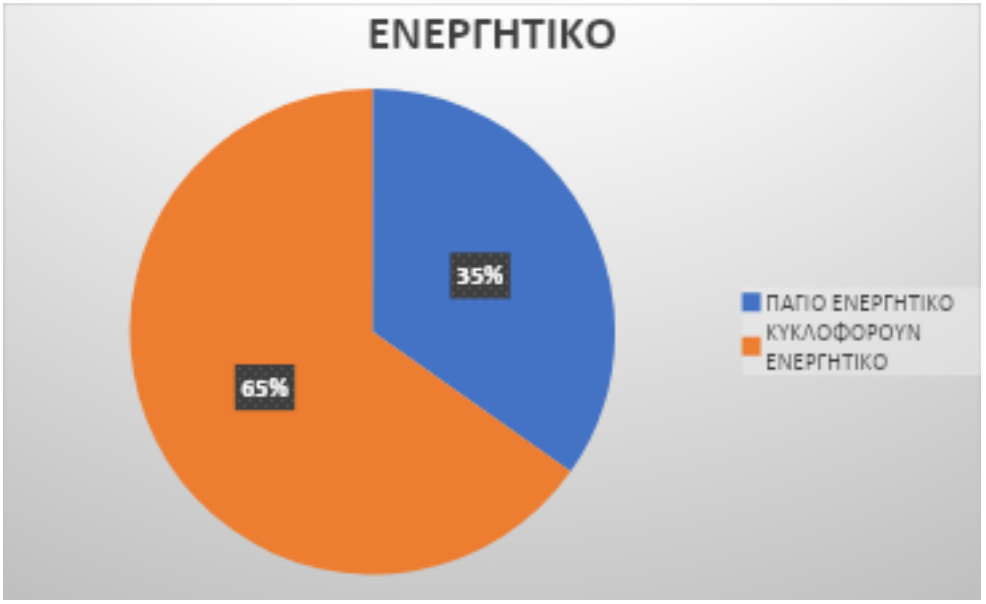
κυκλοφορίας τους για παράδειγμά κάθε χρόνο βγαίνει καινούργιο iPhone κινήτο τηλέφωνο με αποτέλεσμα το προηγούμενο που κυκλοφόρησε ένα χρόνο πριν να χάνει ένα μεγάλο μέρος της αξίας του το ίδιο ακριβός φαινόμενο υπάρχει και με τα υπόλοιπα προϊόντα τεχνολογίας όπως κάρτες γραφικών επεξεργαστές λαπτοπ τηλεοράσεις κλπ., βέβαια η εταιρία φροντίζει να υπολογίζει τις αντίστοιχες προβλέψεις απομείωσης της αξίας των προϊόντων και με αυτό τον τρόπο οι ισολογισμοί να απεικονίζουν την πραγματική αξία των αποθεμάτων για παράδειγμά το έτος 2023 η αξία τον αποθεμάτων ανήλθε τα 68.781.000 ευρώ και οι προβλέψεις για την απαξίωση τους ανήλθαν τα 7.183.000 ευρώ , το ποσοστό απαξίωσης όπως και ο αριθμοδείκτης ταχύτητας κυκλοφορίας αποθεμάτων που θα αναλύσουμε στην επόμενη ενότητα παρέμειναν στα ιδιά επίπεδα με το 2022. Σχετικά με την δύναμη των προμηθευτών να μπορούν να επηρεάζουν τις τιμές των προϊόντων ή να μην αποστέλλουν τις παραγγελίες ή την πιθανότητα να έχουν πάρει χρηματική προκαταβολή και να μην τηρήσουν την συμφωνία είναι σχετικά μικρή καθώς η εταιρία ελέγχει την πιστοληπτική ικανότητα των προμηθευτών της όπως είδαμε παραπάνω αλλά και συνεργάζεται με πόλους προμηθευτές ταυτόχρονα και μόνο ένας από αυτούς την προμηθεύει το 10% της αξίας του συνόλου των προμήθειων με τον οποίο έχει μακροχρόνια καλή και αδιάκοπη σχέση, και δεν υπάρχει κάποιο πρόβλημα σύγκρουσης συμφερόντων, αρά ο συγκείμενος κίνδυνος είναι μικρός παρόλα αυτά με την κρίση (πόλεμο) στην μέση ανατολή(ερυθρά θάλασσα) έχουν αυξηθεί και τα κόστη μεταφοράς και οι χρόνοι παράδοσης των προϊόντων(lead time) καθώς τα πλοία δεν μπορούν να διασχίσουν το στενό του Σουέζ και κάνουν όλο τον κύκλο από την νοτιά Αφρική(cape town) στην Ισπανία και στην συνεχεία φτάνουν στην Ελλάδα , καθώς αυτά τα προϊόντα παράγονται σε χώρες της αίσιας όπως την κίνα το Βιετνάμ κλπ. Ένας τελευταίος αλλά αρκετά σημαντικός παράγοντας που επηρεάζει τις πωλήσεις του ομίλου είναι ο υψηλός πληθωρισμός των τελευταίων μηνών στην Ελλάδα οπού περιορίζει αρκετά τις καταναλωτικές δαπάνες ,αλλά και προβλέψεις για περιορισμό της ανάπτυξης της ελληνικής οικονομίας στο βραχυχρόνιο μέλλον. Στην συνεχεία του κεφαλαίου θα υπολογίσουμε τους αριθμοδείκτες της εταιρίας με βάση τους ισολογισμούς του 2018 έως το 2023.

7.2 ΑΝΑΛΥΣΗ ΙΣΟΛΟΓΙΣΜΟΥ 2023 ΚΑΙ ΥΠΟΛΟΓΙΣΜΟΣ ΑΡΙΘΜΟΔΕΙΚΤΩΝ (2018-2023)

Τα παρακάτω δεδομένά προέρχονται από την πιο πρόσφατη οικονομική έκθεση του ομίλου αυτή του 2023 επίσης θα πρέπει να αναφέρουμε ότι τα δεδομένά αφορούν τα χρηματοοικονομικά στοιχεία του συνόλου του ομίλου για το 2023 και όχι της εταιρίας και ότι τα ποσά είναι σε χιλιάδες δηλαδή όταν εμφανίζεται ένα ποσό στα παρακάτω διαγράμματα πχ έστω αποθέματα ίσο με 10 στην πραγματικότητα είναι συνολικής αξίας ίση με 10.000 ευρώ. Αρχικά ξεκινάμε βλέποντας ότι ο ισολογισμός είναι ισοσκελισμένος δηλαδή το ενεργητικό είναι ίσο με παθητικό του ομίλου.

Όταν αναφερόμαστε στο ενεργητικό εννοούμε όλα εκείνα τα στοιχεία που ανήκουν στην εταιρεία και χρησιμοποιούνται με στόχο να επιτευχθούν οι προσδοκίες μιας επιχείρησης παραδείγματος χάρη αύξηση των πωλήσεων , αύξηση των κερδών διείσδυση σε μια αγορά κλπ. Το ενεργητικό χωρίζεται σε δυο κατηγορίες στο πάγιο ενεργητικό δηλαδή στοιχεία του ενεργητικού που η επιχείρηση σκοπεύει να χρησιμοποιήσει μακροπρόθεσμα και δεν είναι ευκολά ρευστοποιήσιμα παραδείγματα τέτοιων πόρων είναι το κέντρο διανομής του πλαίσιο στην μαγουλά , τα φορτηγά οχήματα , τα μηχανήματα όπως για παράδειγμά τα παλετοφορα , τα ιδιόκτητα καταστήματα της εταιρίας κλπ. παρόλο που κάποια από αυτά φαίνονται ευκολά ρευστοποιήσιμα παραδείγματος χάρη ένα κλαρκ ή ένα εταιρικό αυτοκίνητο δεν είναι εύκολο να πουληθούν σε σύντομο χρονικό διάστημα πχ για να βρεθούν χρήματα για την αποπληρωμή ενός ομολογιακού δανείου ή για την αγορά αποθεμάτων χωρίς να χάσουν μεγάλο μέρος της υπολειπομένης αξίας τους καθώς μπορεί

να μην υπάρχει ανταγωνιστική εταιρία που να δεχτεί να τα αγοράσει στην κανονική τους αξία , το παραπάνω είναι μια από τις πέντε δυνάμεις του Πόρτερ αναφορικά με την δυσκολία εισόδου και εξόδου από την εκάστοτε αγορά. Το δεύτερο κομμάτι του ενεργητικού είναι το κυκλοφορούν ενεργητικό το οποίο περιλαμβάνει όλα εκείνα τα στοιχεία που έχει στην κατοχή της η εταιρία και είναι ευκολά ρευστοποιήσιμα στο βραχυχρόνιο μέλλον όπως για παράδειγμά τα ταμειακά διαθέσιμα(μετρητά) διαφορά χρεόγραφα όπως γραμματεία εισπρακτέα τα αποθέματα και αλλά όπως απαιτήσεις από πελάτες(δηλαδή πωλήσεις της εταιρίας με πίστωση), βέβαια σχετικά με τα αποθέματα αν αυτά δεν πωληθούν σε σύντομο χρονικό διάστημα ειδικά όσον αφορά προϊόντα τεχνολογίας υπάρχει υψηλός κίνδυνος απαξίωσης όπως είδαμε και στην προηγουμένη ενότητα.

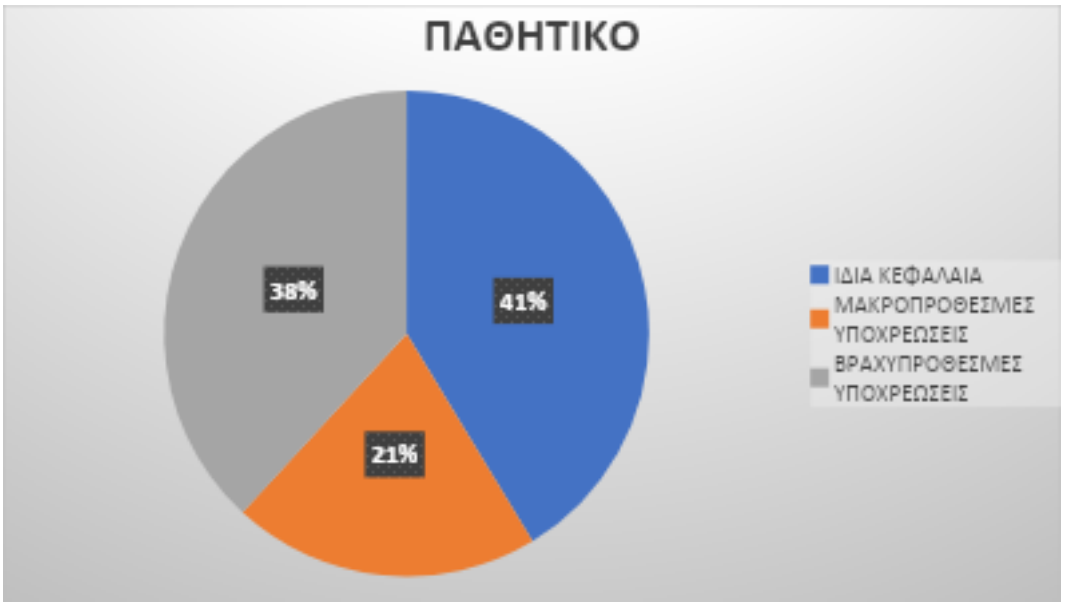


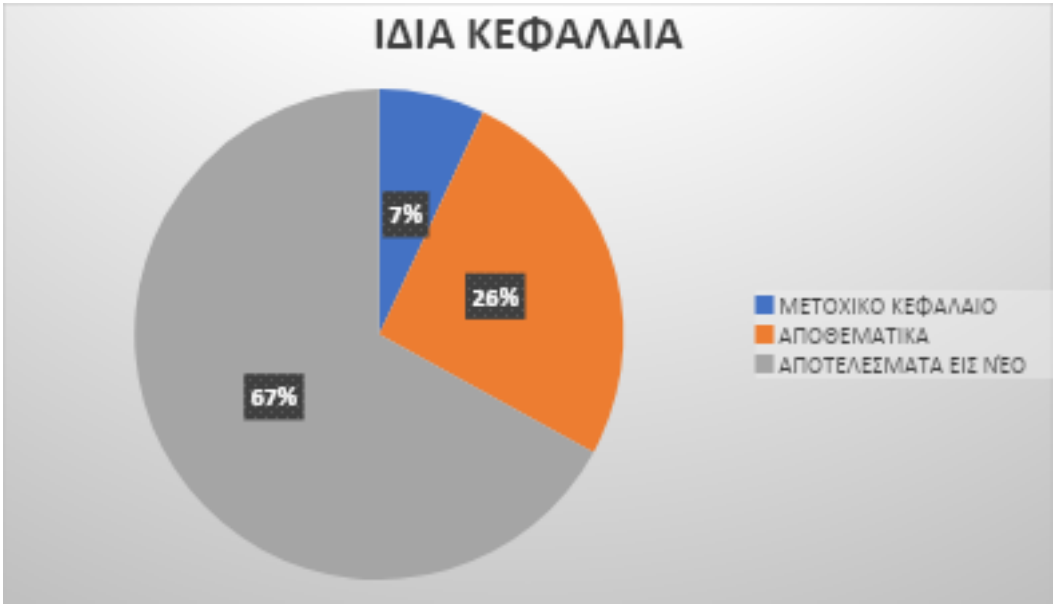
Όσο αφορά το ενεργητικό του πλαίσιο παρατηρούμε ότι το μεγαλύτερο μέρος αυτού αποτελείται από κυκλοφορούν στοιχεία που είναι πολύ λογικό για μια εταιρία λιανικών πωλήσεων καθώς η εταιρία θα πρέπει να έχει μεγάλο αριθμό αποθεμάτων έτσι ώστε να ανταποκριθεί στην διακύμανση της ζήτησης για το εκάστοτε προϊόν. Επίσης το 35% του συνόλου των επενδύσεων σε πάγιο ενεργητικό δείχνει ότι η εταιρία δεν δεσμεύει μεγάλο μέρος των πόρων της σε πάγια στοιχεία όπως οικόπεδα εγκαταστάσεις κλπ. αλλά αντί αυτού κρατεί ρευστά διαθέσιμα για να μπορεί να αγοράζει εμπορεύματα και να ανταποκρίνεται σε βραχυπρόθεσμες υποχρεώσεις , ενώ ταυτόχρονα τα περισσότερα κτήρια που στεγάζουν τα καταστήματα της δεν είναι ιδιόκτητα αλλά με μίσθωση αυτό θα

το δούμε και στο διάγραμμα αναφορικά με το πάγιο ενεργητικό οπού τα δικαιώματα χρήσης παγίων καταλαμβάνουν ένα μεγάλο ποσοστό του συνόλου του πάγιου ενεργητικού.

Όπως είπαμε και παραπονώ η εταιρεία φαίνεται να νοικιάζει με μακροπρόθεσμες συμφωνίες εάν μεγάλο μέρος των κτιρίων οπού χρησιμοποίει και με αυτό τον τρόπο να αποφεύγει κίνδυνους που πηγάζουν από την ιδιοκτησία μεγάλων εγκαταστάσεων καθώς σε περίπτωση οπού σε μια περιοχή δεν υπάρχει επαρκής ζήτηση για να καλύψει τα κόστη λειτουργίας ενός καταστήματος αυτό μπορεί να παρθεί η απόφαση να διακοπεί η λειτουργεία του πιο ευκολά και με χαμηλότερο κόστος ευκαιρίας σε σχέση με το να το είχε στην κατοχή της η εταιρεία (στην πρώτη κατηγορία της ενοικίασης αναφερόμαστε σε μεταβλητό κόστος ενώ στην δεύτερη κατηγορία με την ιδιοκτησία ενός κτιρίου σε σταθερό κόστος(μεγάλη απόσβεση) το οποίο μεταβάλει προς τα πάνω το νεκρό σημείο , γενικά μια οποιαδήποτε εταιρεία θέλει να δοκιμάσει αν ένα κατάστημα θα έχει την προσδοκώμενη ζήτηση είναι καλυτέρα αρχικά να το ενοικιάσει ενώ όταν κάτι γνωρίζει ότι δουλεύει σωστά και αποδίδει στην επιχείρηση να το αγοράσει όπως για παράδειγμά το κέντρο διανομής το οποίο είτε κλείσει είτε όχι ένα κατάστημα αυτό θα συνεχίσει την λειτουργία του καθώς από εκεί γίνεται η αρχική διανομή των προϊόντων. Όλα τα υπόλοιπα στοιχεία έχουν μικρό ποσοστό σε σχέση με το σύνολο όπως για παράδειγμά τα άυλα περιουσιακά στοιχεία (για παράδειγμα δικαιώματα χρήσης λογισμικών όπως το Microsoft office και αλλά)

Σχετικά με το δεύτερο κομμάτι του ενεργητικού το κυκλοφορούν βλέπουμε ότι η εταιρία διακρατεί ένα μεγάλο μέρος του συνόλου σε ρευστά διαθέσιμα έτσι ώστε να μπορεί να ανταποκρίνεται σε βραχυχρόνιες υποχρεώσεις όπως πληρωμές ενοικίων δόσεις δάνειων και μισθούς εργαζομένων και για να μπορεί να αγοράζει νέα εμπορεύματα. Τα αποθέματα επίσης αποτελούν μεγάλο ποσοστό του κυκλοφορούν αποθεματικό το οποίο είναι λογικό λόγο της φύσης της εταιρίας η όποια δεν λειτουργεί με διαδικασίες just in time δηλαδή μόλις έρχεται μια παραγγελία από το διαδίκτυο τότε να αγοράζει το προϊόν από τον προμηθευτή αυξάνοντας με αυτόν τον τρόπο τους χρόνους παράδοσης αλλά αντίθετος διακρατει απόθεμα ώστε να το δειγματιζει στα καταστήματα του και να εξυπηρετεί άμεσα της ανάγκες των πελατών. Στην συνεχεία θα αναλύσουμε με τους αριθμοδείκτες κατά ποσό αυτή η στρατηγική είναι αποδοτική. Όσον αφορά της απαιτήσεις της εταιρίας από τους πελάτες πρέπει και αυτές να τις αναλύσουμε περετέρω ώστε να διαπιστώσουμε εάν η εταιρεία εισπράττει γρηγορότερα τις απαιτήσεις της από τους πελάτες σε σχέση με το ποσό γρηγορά πληρώνει εκείνη τους προμηθευτές της. Στην συνεχεία γίνεται αναφορά στο παθητικό της εταιρείας για το έτος 2023. Το παθητικό αφορά τις υποχρεώσεις της επιχείρησης προς τρίτους όπως προμηθευτές τράπεζες εργαζομένους αλλά και προς τους ιδιοκτήτες της εταιρίας μέσο των ιδίων κεφαλαίων και δείχνει το τρόπο με τον οποίο αποκτήθηκαν τα στοιχεία του ενεργητικού. Αναφορικά με το πλαίσιο βλέπουμε ότι ένα μεγάλο μέρος των υποχρεώσεων είναι προς τους ιδιοκτήτες της εταιρίας που δείχνει ότι η εταιρεία βασίζεται σε δικά τις κεφάλαια για να χρηματοδοτεί τις επενδύσεις της, στην συνεχεία θα αναλύσουμε περετέρω την σχέση ιδίων προς ξένων κεφαλαίων.



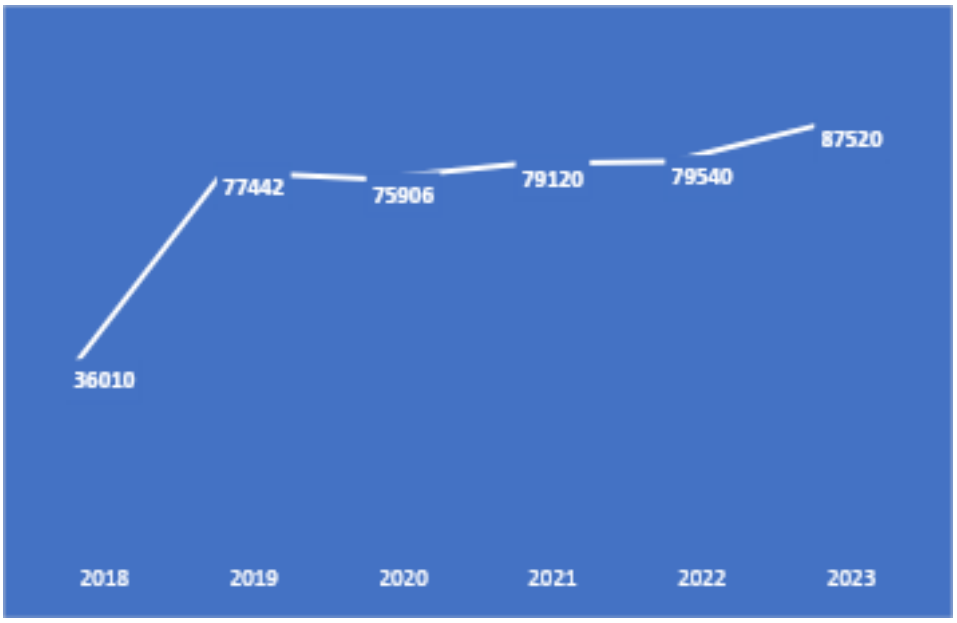


Όπως είπαμε και παραπάνω τα ιδιά κεφάλαια αποτελούνται από το μετοχικό κεφάλαιο δηλαδή τα χρήματα που έχουν εισφέρει οι μέτοχοι της εταιρίας κατά την ίδρυση της ή κατά την αύξηση του μετοχικού κεφαλαίου ανά διαστήματα οπού απαιτούνταν επενδύσεις και τα ιδιά κεφάλαια ήταν πιο φθηνός τρόπος χρηματοδότησης σε σχέση με την σύναψη τραπεζικών δάνειων και έκδοσης ομολόγων (φθηνότερο επιτόκιο δανεισμού). Τα αποθεματικά αναφέρονται σε κέρδη προηγουμένων χρήσεων τα όποια δεν αναμένεται να διανεμηθούν στους μέτοχους στο μέλλον καθώς διακρατούνται για επενδύσεις ή για περίοδούς οπού η οικονομία βρίσκεται σε ύφεση και η πορεία της εταιρείας είναι πτωτική. Ενώ από την άλλη πλευρά τα κέρδη εις νέο είναι και αυτά κέρδη από προηγούμενες

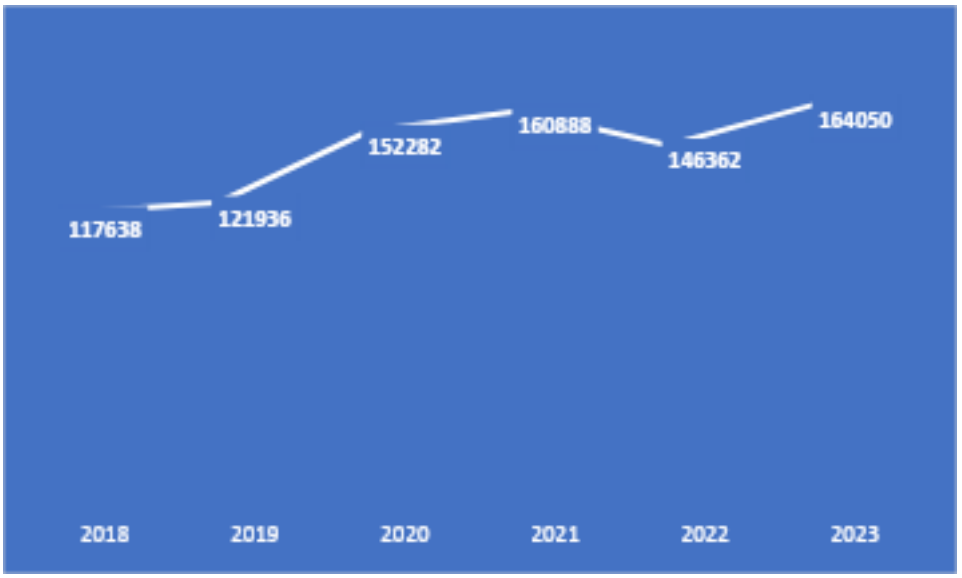
χρήσεις αλλά υπάρχει μεγάλη πιθανότητα να διανεμηθούν στους μέτοχους της εταιρίας ή να επενδυθούν στο βραχυπρόθεσμο μέλλον.

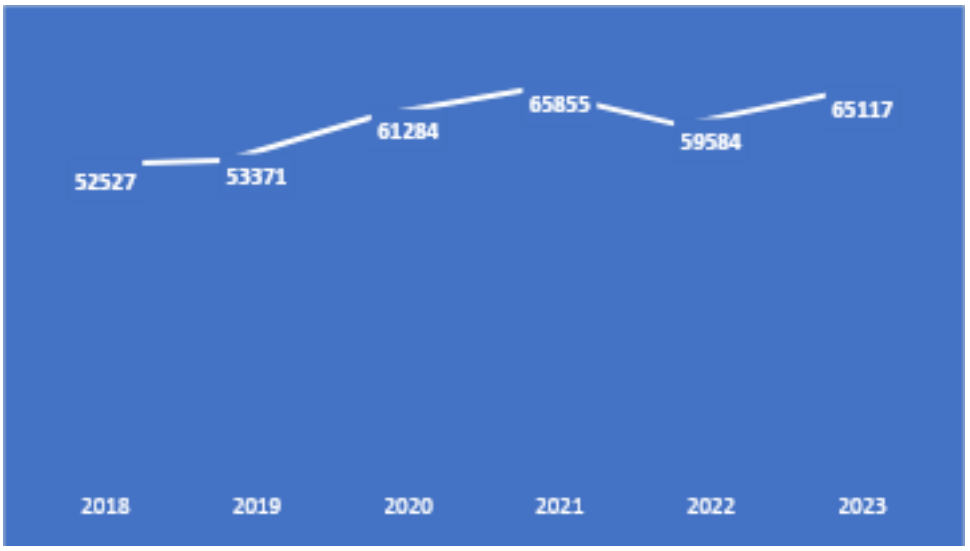
Αναφορικά με τις μακροπρόθεσμες υποχρεώσεις το μεγαλύτερο ποσοστό αυτών καταλαμβάνουν οι υποχρεώσεις μισθώσεων δηλαδή οι πληρωμές για την ενοικίαση των κτιρίων και αποθηκών για τις οποίες υπάρχουν συμβάσεις μισθώσεις χρονικού ορίζοντα από τρία έως 40 χρονιά και με ρήτρα επέκταση αυτών κατά την λήξη τους , ενώ όπως έχουμε αναφέρει η εταιρία έχει συνάψει κάποια μακροπρόθεσμα ομολογιακά δάνεια με την εθνική τράπεζα και την Eurobank.

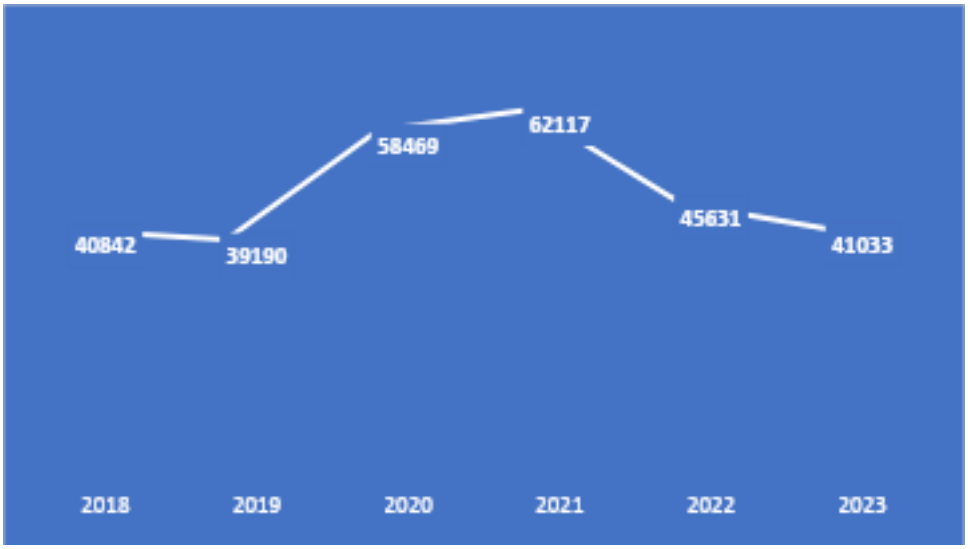
Όπως παρατηρούμε η επιχείρηση δεν αγοράζει τα εμπορεύματα της με πίστωση καθώς ο λογαριασμός προμηθευτές καταλαμβάνει μικρό ποσοστό των βραχυπροθέσμων υποχρεώσεων , αντίθετος ο λογαριασμός με την μεγαλύτερη αξία είναι οι προβλέψεις για ανομοίωση της αξίας των εμπορευμάτων και εξόδων όπως την μισθοδοσία των εργαζομένων. Στην συνεχεία πριν την ανάλυση των αριθμοδεικτών παρουσιάζονται οι αξίες κάποιον λογαριασμών του ισολογισμού (τα ποσά είναι σε χιλιάδες) και της κατάστασης αποτελεσμάτων.

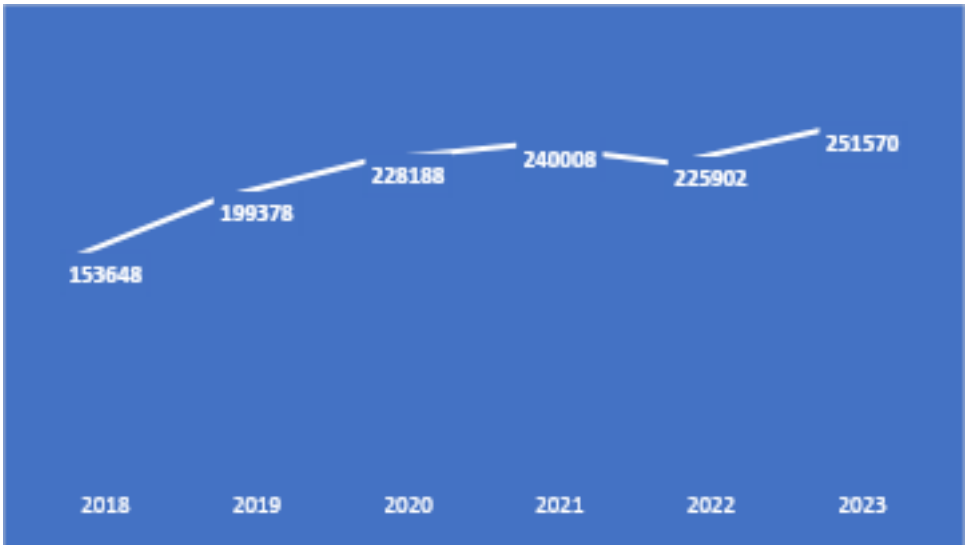


Αρχικά βλέπουμε ότι για τα έξι τελευταία χρονιά αθροιστικά το πάγιο ενεργητικό αυξάνεται, και το ίδιο συμβαίνει και με το σύνολο του κυκλοφορούν ενεργητικού αλλά και με τα υποσύνολα του όπως τα αποθέματα και τα ταμειακά διαθέσιμα , όλα τα παραπάνω οδηγούν στην αθροιστική αύξηση του ενεργητικού της εταιρίας.

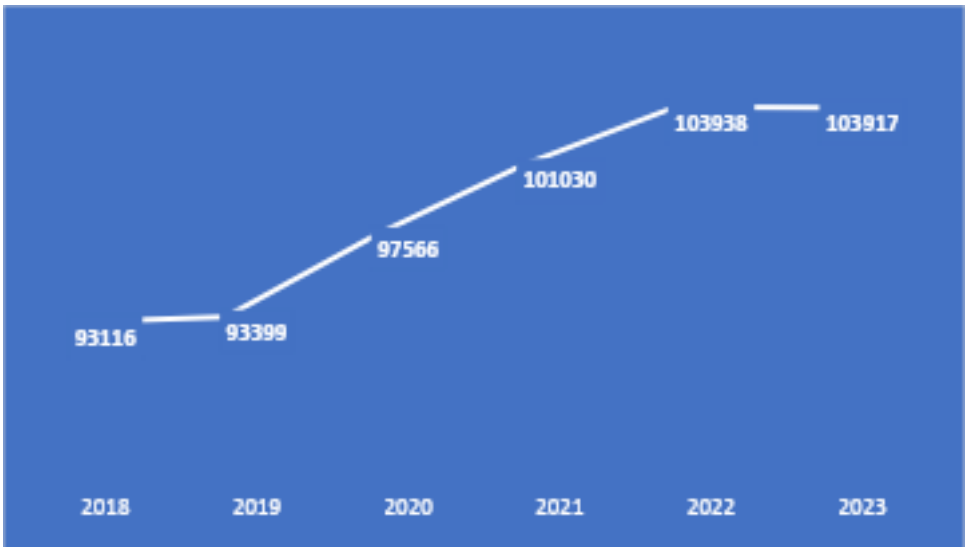


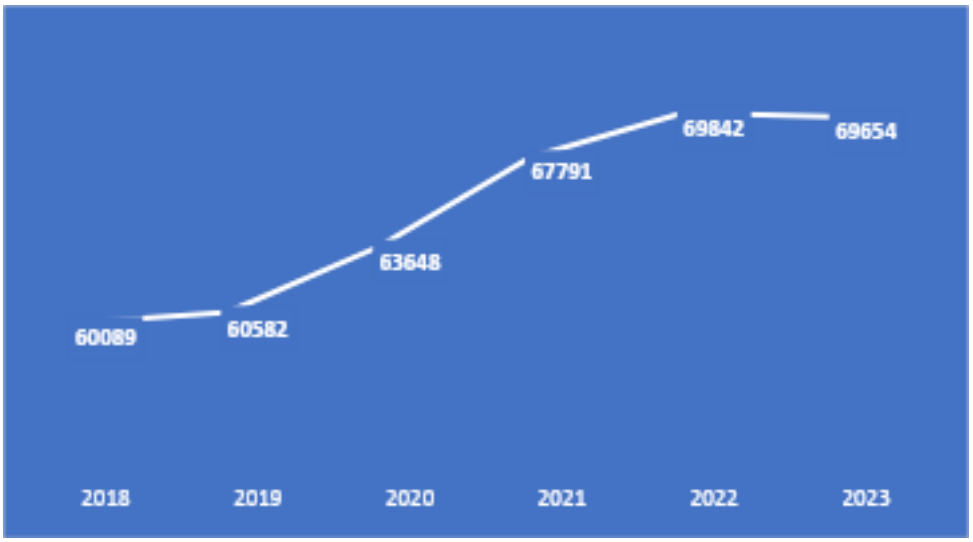

52527
53371
61284
65855
59584
65117
2018
2019
2020
2021
2022
2023


40842
39190
58469
62117
45631
41033
2018
2019
2020
2021
2022
2023


153648
199378
228188
240008
225902
251570
2018
2019
2020
2021
2022
2023

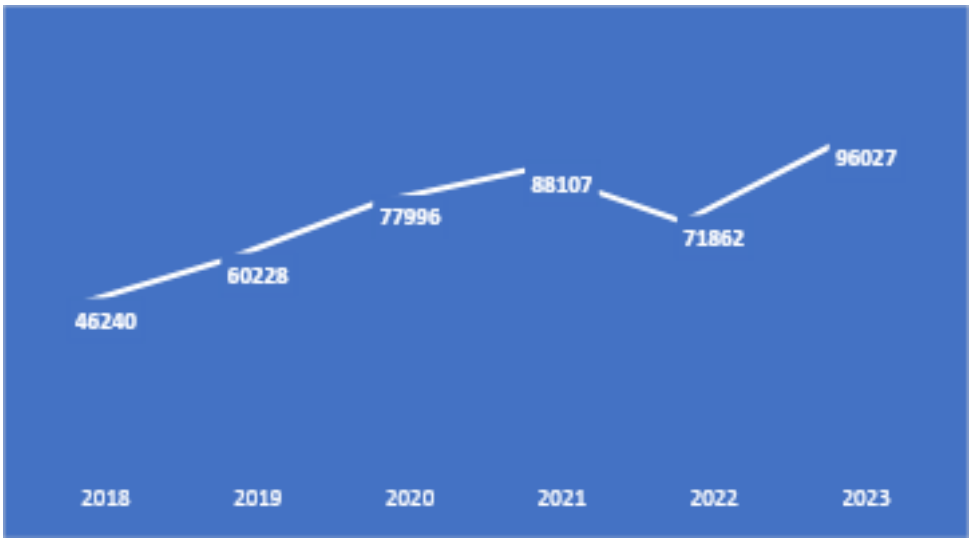
Όσον αφορά το παθητικό αρχικά παρατηρούμε ότι ο λογαριασμός ιδιά κεφάλαια αυξάνεται κυρίως λόγο της αύξησης των κερδών της επιχείρησης και κατά επέκταση της διακρατικής τους για μελλοντικές επενδύσεις.

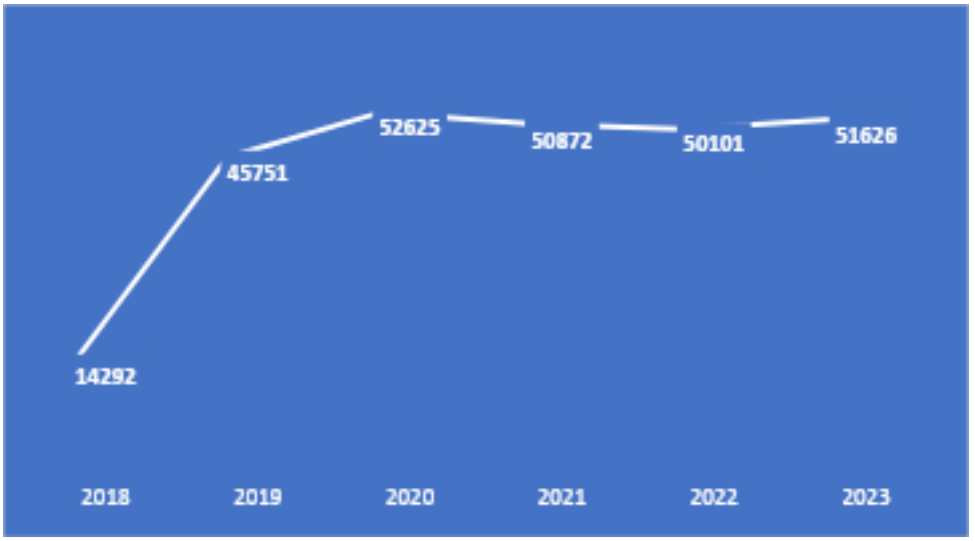


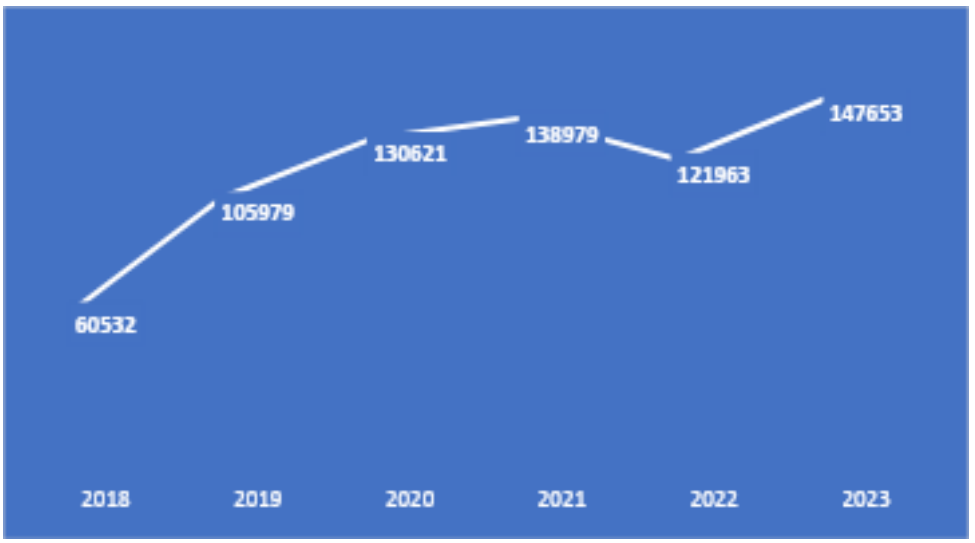


Όσο αφορά τα ξένα κεφάλαια παρατηρούμε ότι οι μακροπρόθεσμες υποχρεώσεις αυξάνονται απότομα από το 2018 προς το 2019 και στην συνεχεία για τα επόμενα έτη η διακύμανση(μεταβολή) τους είναι μικρή. από την άλλη πλευρά οι βραχυπρόθεσμες υποχρεώσεις αυξάνονται καθ' όλη την χρονοσειρα εκτός από το έτος 2022. Αποτέλεσμα όλων το παραπάνω είναι η συνολική αύξηση των ξένων κεφαλαίων και κατά επέκταση του παθητικού της εταιρίας , το οποίο είναι λογικό καθώς και το ενεργητικό της αυξήθηκε ,προφανώς στόχος της εκάστοτε εταιρίας είναι μέσο της αύξησης των επενδύσεων σε στοιχεία του ενεργητικού να αυξήσει την απόδοση τους και κατά επέκταση τις πωλήσεις

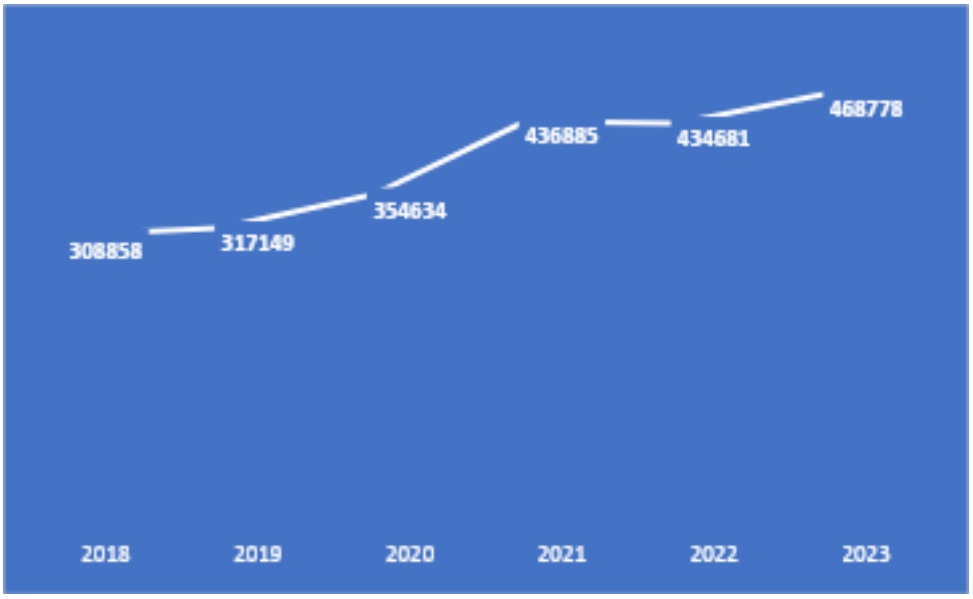
και τα κέρδη του ομίλου. Τα παραπάνω μπορούν σε κάποιο βαθμό να αναλυθούν μέσα από τους αριθμοδείκτες δραστηριότητας αλλά και απόδοσης όπως ο roa (return on assets).

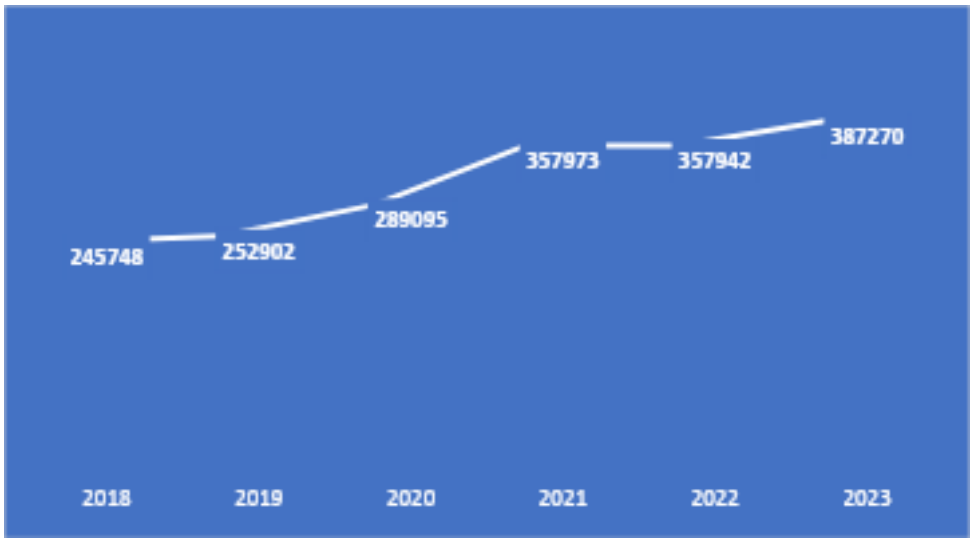


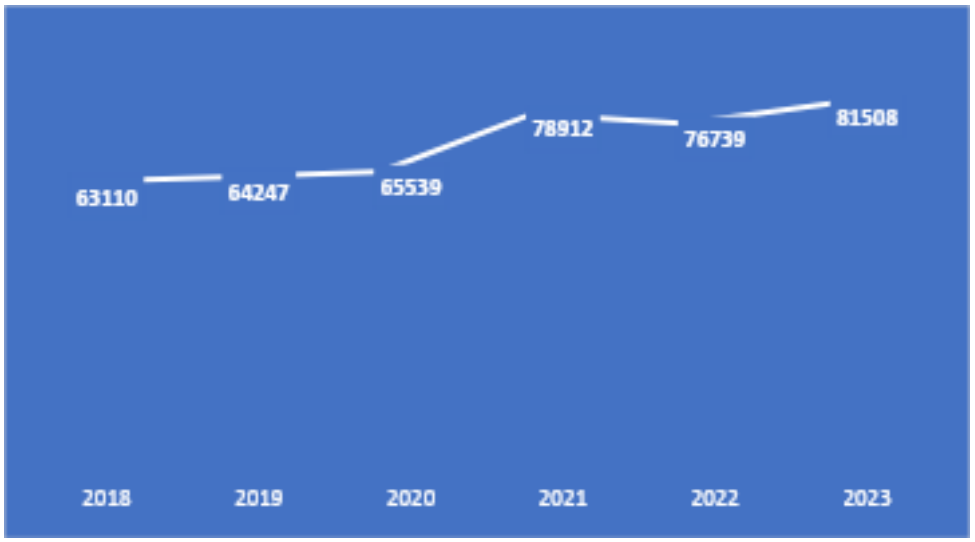


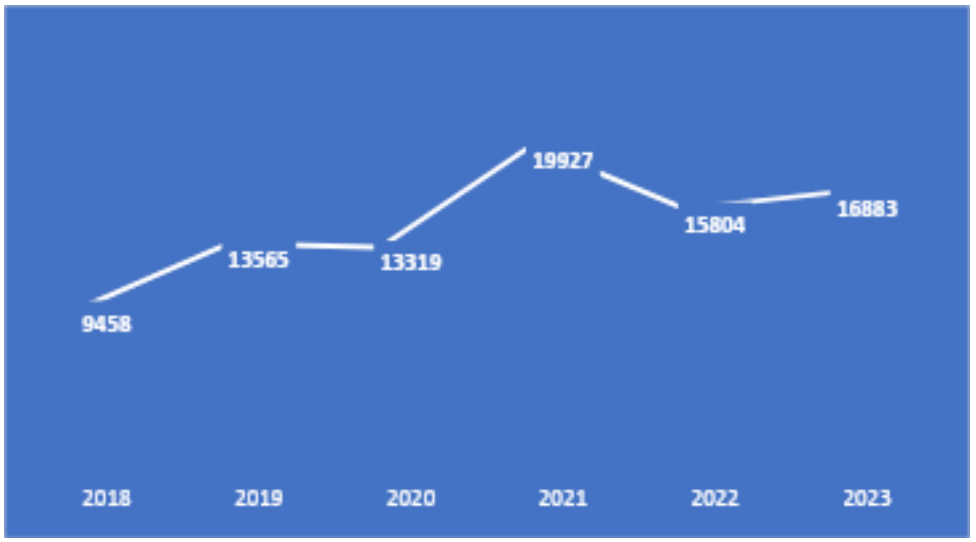


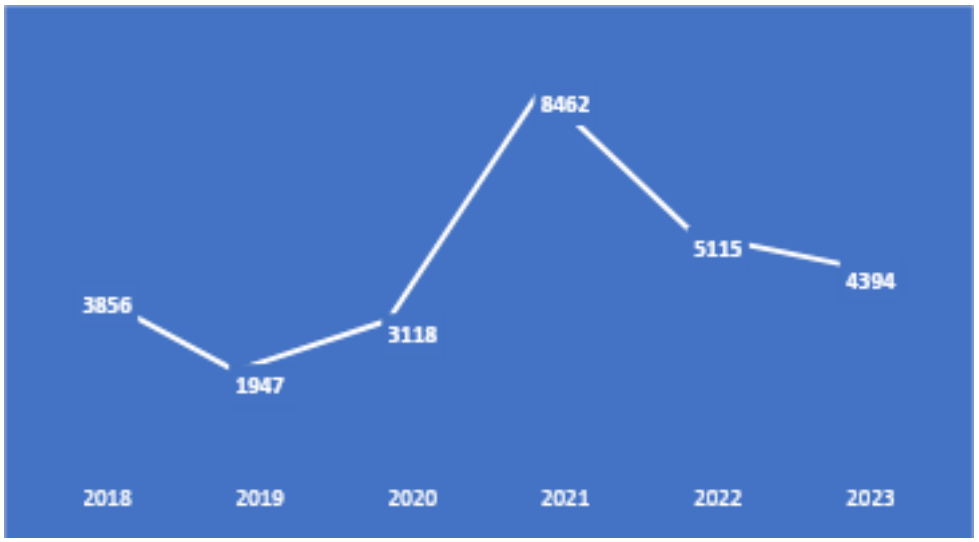
Στην συνέχεια βλέπουμε κάποια διαγράμματα στοιχείων της κατάστασης αποτελεσμάτων όπως οι πωλήσεις το κόστος πωληθεντων οπού και τα δυο αυξάνονται κατά την διάρκεια των έξι ετών τα οποία θα αναλύσουμε διεξοδικά και θα προβλέψουμε τις εξαμηνίες τους τιμές για το έτος 2024 με χρήση των μεθόδων μηχανικής μάθησης που αναλύσαμε στο κεφάλαιο 7. Πέρα από τα παραπάνω βλέπουμε ότι οι πωλήσεις έχουν σχεδόν τελειά θετική συσχέτιση με το κόστος πωληθεντων και ότι οι πωλήσεις του πλαίσιο αυξήθηκαν δραματικά το έτος 2021 λόγο της καραντίνας που είχε επιβληθεί στους κάτοικους της Ελλάδας οι οποίοι εφόσον δεν μπορούσαν να καταναλώσουν το εισόδημα τους σε εξωτερικές δραστηριότητες αυξήσαν τις δαπάνες τους σε προϊόντα τεχνολογίας και ψυχαγωγίας εντός της κατοικίας όπως οι τηλεοράσεις κονσόλες υπολογιστές και αλλά. Για αυτό και το EBIDA (δηλαδή το μικτό αποτέλεσμα μείον τα κόστη λειτουργίας μας δίνει το ebida δηλαδή τα κέρδη πριν αφαιρεθούν οι τόκοι των δάνειων οι φόροι και οι αποσβέσεις) και τα κέρδη χρήσης μετα από φόρους του 2021 ήταν μεγαλύτερα από όλα τα υπόλοιπα έτη. Όλα τα παραπάνω έχουν ως αποτέλεσμα να αυξηθούν τα διανεμηθέντα κέρδη προς του μέτοχους της εταιρίας και κατά επέκταση το μέρισμα ανά μετοχή να διαμορφωθεί για το οικονομικό έτος του 2023 ίσο με 0,2 ευρώ ανά μετοχή , το υψηλότερο που έχει διανεμηθεί σε αυτά τα τελευταία έξι χρονιά.

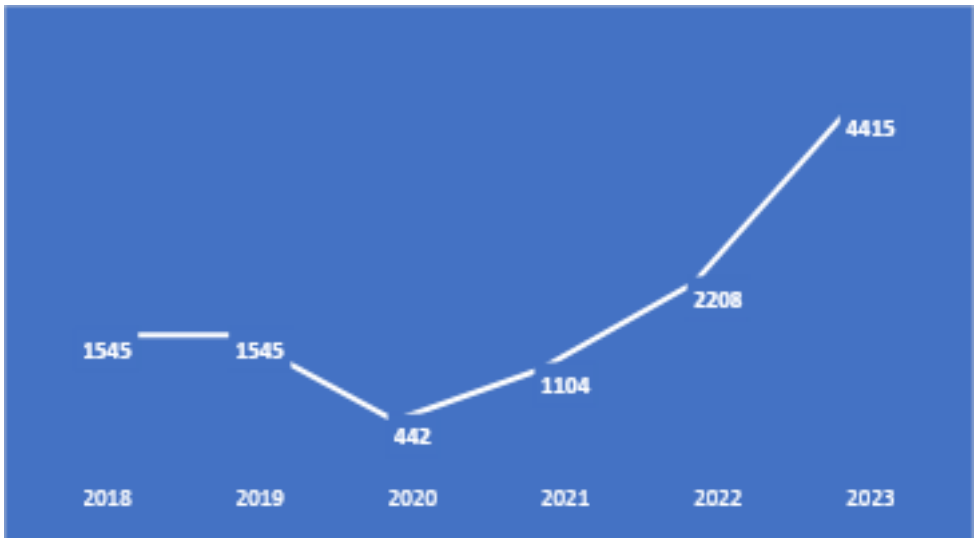



245748
252902
289095
357973
357942
387270
2018
2019
2020
2021
2022
2023

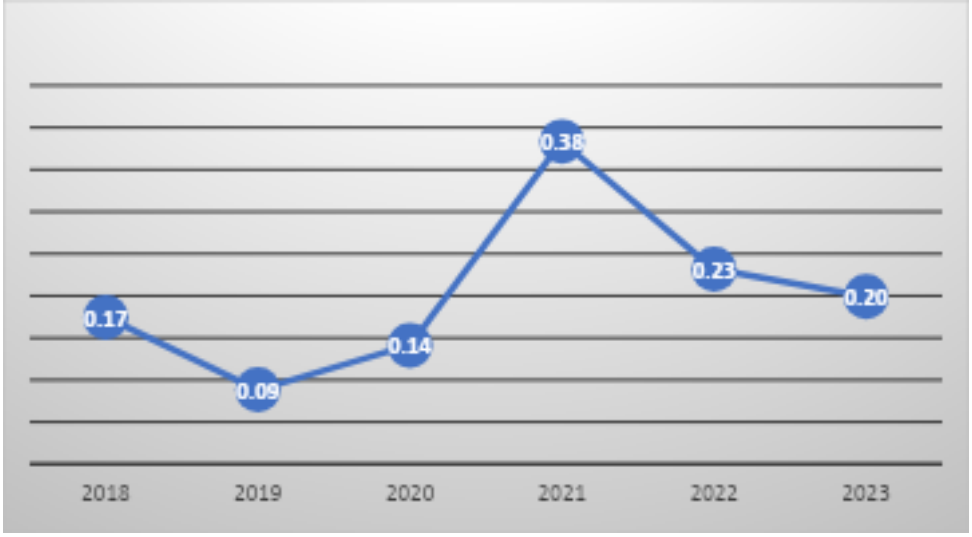

63110
64247
65539
78912
76739
81508
2018
2019
2020
2021
2022
2023


9458
13565
13319
19927
15804
16883
2018
2019
2020
2021
2022
2023


3856
1947
3118
8462
5115
4394
2018
2019
2020
2021
2022
2023


1545
1545
442
1104
2208
4415
2018
2019
2020
2021
2022
2023


ΜΕΡΙΣΜΑ
0.07
0.07
0.02
0.05
0.1
0.2
2018
2019
2020
2021
2022
2023

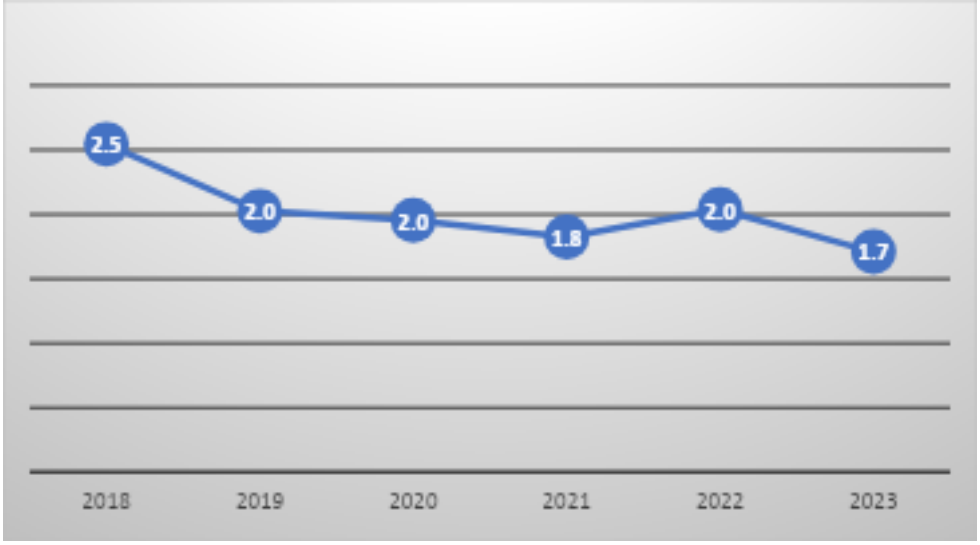


Για όλους τους παραπάνω λογαριασμούς και τα στοιχεία της κατάστασης αποτελεσμάτων έχει υπολογιστεί ο μέσος ορός, η δειγματική τυπική απόκλιση καθώς έχουμε παρατηρήσεις μόνο των τελευταίων έξι ετών ενώ η εταιρεία λειτουργεί περισσότερα από πενήντα χρονιά και στην συνέχεια υπολογίζεται το άνω όριο και κάτω όριό του διαστήματος εμπιστοσύνης με βάση τ στατιστικό με πέντε βαθμούς ελευθέριας και επίπεδο σημαντικότητας 95% δηλαδή α=5% καταλήγουμε σε τ στατιστικό ίσο με 2,57.

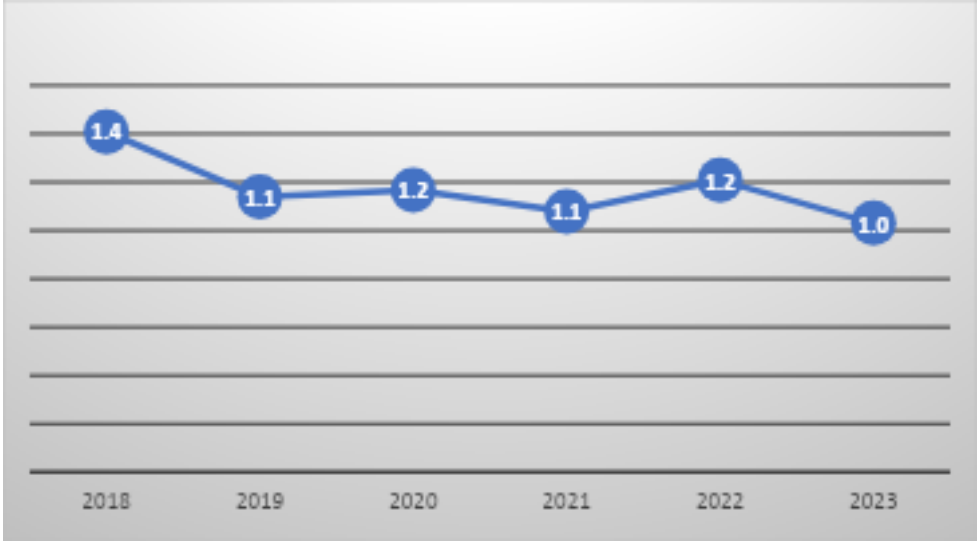
ΛΟΓΑΡΓΙΑΣΜΟΣ	ΜΕΣΟΣ ΟΡΟΣ	ΤΥΠΙΚΗ ΑΠΟΚΛΥΣΗ	ΚΑΤΩ ΟΡΙΟ	ΑΝΩ ΟΡΙΟ
ΠΑΓΙΟ ΕΝΕΡΓΗΤΙΚΟ	72589,67	18365,63	53316,12	91863,22
ΚΥΚΛΟΦΟΡΟΥΝ ΕΝΕΡΓΗΤΙΚΟ	143859,33	19713,11	123171,69	164546,98
ΑΠΟΘΕΜΑΤΑ	59623,00	5678,43	53663,85	65582,15
ΤΑΜΕΙΑΚΑ ΔΙΑΘΕΣΗΜΑ	47880,33	9917,74	37472,30	58288,37
ΣΥΝΟΛΟ ΕΝΕΡΓΗΤΙΚΟΥ	216449,00	35358,33	179342,70	253555,30
ΙΔΙΑ ΚΕΦΑΛΑΙΑ	98827,67	4910,24	93674,68	103980,65
ΑΠΟΤΕΛΕΣΜΑΤΑ ΕΙΣ ΝΈΟ	65267,67	4425,59	60623,30	69912,03
ΒΡΑΧΥΠΡΟΘΕΣΜΕΣ ΥΠΟΧΡΕΩΣΕΙΣ	73410,00	18236,95	54271,49	92548,51
ΜΑΚΡΟΠΡΟΘΕΣΜΕΣ ΥΠΟΧΡΕΩΣΕΙΣ	44211,17	14848,28	28628,85	59793,48
ΞΕΝΑ ΚΕΦΑΛΑΙΑ	117621,17	31434,47	84632,72	150609,62
ΠΩΛΗΣΕΙΣ	386830,83	68530,80	314912,17	458749,50
ΚΟΣΤΟΣ ΠΩΛΗΘΕΝΤΩΝ	315155,00	60393,00	251776,43	378533,57
ΜΙΚΤΑ ΑΠΟΤΕΛΕΣΜΑΤΑ	71675,83	8257,01	63010,63	80341,03
EBITDA	14826,00	3574,48	11074,81	18577,19
ΚΕΡΔΗ ΧΡΗΣΗΣ ΜΕΤΑ ΑΠΌ ΦΟΡΟΥΣ	4482,00	2232,82	2138,80	6825,20
ΚΕΡΔΗ ΑΝΑ ΜΕΤΟΧΗ	0,20	0,10	0,10	0,31
ΔΙΑΝΟΜΗΜΕΝΑ ΚΕΡΔΗ	1876,50	1373,19	435,43	3317,57
ΜΕΡΙΣΜΑ	0,09	0,06	0,02	0,15

ΑΡΗΘΜΟΣ ΜΕΤΟΧΩΝ	22075000,00	0,00	22075000,00	22075000,00

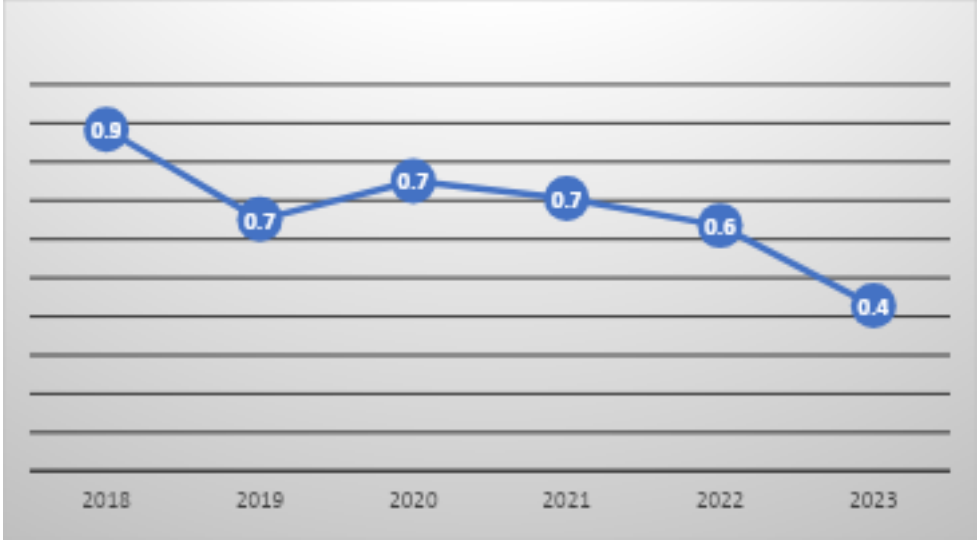
Συνεχίζουμε την ανάλυση της εταιρίας υπολογίζοντας τους αριθμοδείκτες ρευστότητας των τελευταίων έξι ετών λειτουργείας της.



Για να υπολογίσουμε τον αριθμοδείκτη γενικής ρευστότητας πρέπει να διαιρέσουμε το σύνολο του κυκλοφορούντο ενεργητικού με τις βραχυπρόθεσμες υποχρεώσεις του ομίλου. Το αποτέλεσμα μας δείχνει πόσες φορές μπορεί το κυκλοφορούν ενεργητικό να καλύψει τις βραχυπρόθεσμες υποχρεώσεις του ομίλου στην συγκεκριμένη περίπτωση για το πλαίσιο ο αριθμοδείκτης έχει πτωτική πορεία αλλά είναι διαχρονικά κοντά στο δυο το οποίο φανερώνει ότι δεν υπάρχει κάποιο έντονο πρόβλημα υψηλής αποθεματοποίησης , και αρά ότι ο όμιλος διαχειρίζεται σωστά τους πόρους του. Ο μέσος ορός που υπολογιστικέ για τον συγκεκριμένο δείκτη είναι το 2 και διάστημα εμπιστοσύνης ίσο με [1,7-2,3].



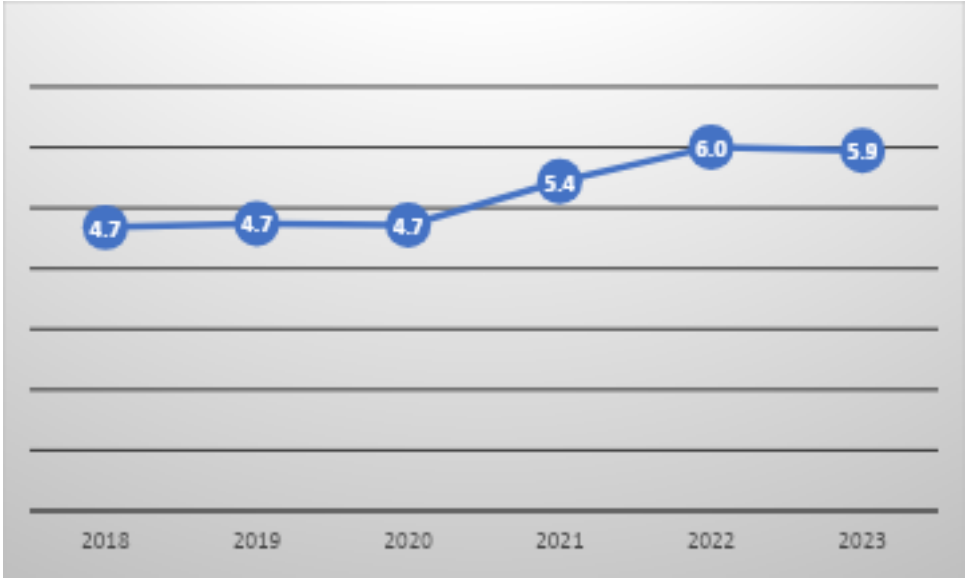
Συνεχίζουμε υπολογίζοντας τον αριθμοδείκτη ειδικής ρευστότητας ο οποίος έχει στον αριθμητή το σύνολο του κυκλοφορούν ενεργητικού μείον τα αποθέματα και στον παρονομαστή το σύνολο των βραχυπροθέσμων υποχρεώσεων. Ο συγκεκριμένος δείκτης μας επιβεβαιώνει τον χαμηλό πιστωτικό κίνδυνο του ομίλου που αναφέρθηκε στην προηγουμένη ενότητα καθώς χωρίς τα αποθέματα τα οποία σε περίπτωση ύφεσης είναι δύσκολο για την επιχείρηση να τα πουλήσει στην αξία που τα αγόρασε μπορεί να καλύψει το σύνολο των βραχυπροθέσμων υποχρεώσεων μια φορά. Ο μέσος ορός για τα έξι έτη οπού εξετάζουμε είναι 1,2 και το διάστημα εμπιστοσύνης ίσο με [1-1,3].



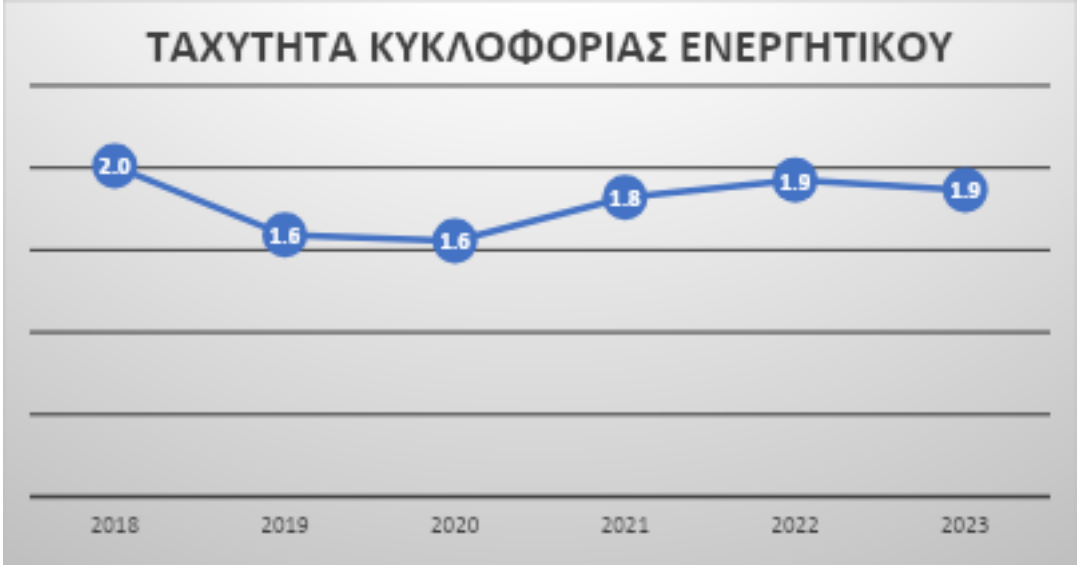
Τέλος ο πιο αυστηρός δείκτης ρευστότητας είναι αυτός της ταμειακής ρευστότητας οπού στον αριθμητή του κλάσματος υπάρχουν μόνο τα ταμειακά διαθέσιμα και αυτά διαιρούνται με το σύνολο των βραχυπροθέσμων υποχρεώσεων. Ουσιαστικά μας δείχνει πόσες φορές τα ταμειακά διαθέσιμα μπορούν να καλύψουν της βραχυπρόθεσμες

υποχρεώσεις στην περίπτωση του πλαίσιο ο αριθμοδείκτης είναι ίσος με 0,4 για το 2023 και μέσα στα έξι έτη δεν ξεπέρασε την μονάδα, βέβαια το παραπάνω δεν είναι κάτι ανησυχητικό καθώς η διακρατιση υψηλών ταμειακών διαθέσιμων επηρεάζεται από τον πληθωρισμό και γενικότερα είναι ένας αχρησιμοποίητος πόρος για την επιχείρηση. Ο μέσος ορός των τελευταίων έξι ετών είναι ίσος με 0,7 και το διάστημα εμπιστοσύνης έχει διαμορφωθεί ως εξής [0,51-0,83].

Η επόμενη κατηγορία αριθμοδεικτών είναι οι αριθμοδείκτες δραστηριότητας οι οποίοι δείχνουν στους αναλυτές της εκάστοτε εταιρίας ποσό αποτελεσματικά χρησιμοποιεί μια επιχείρηση τα στοιχεία του ενεργητικού της για την επίτευξη πωλήσεων .



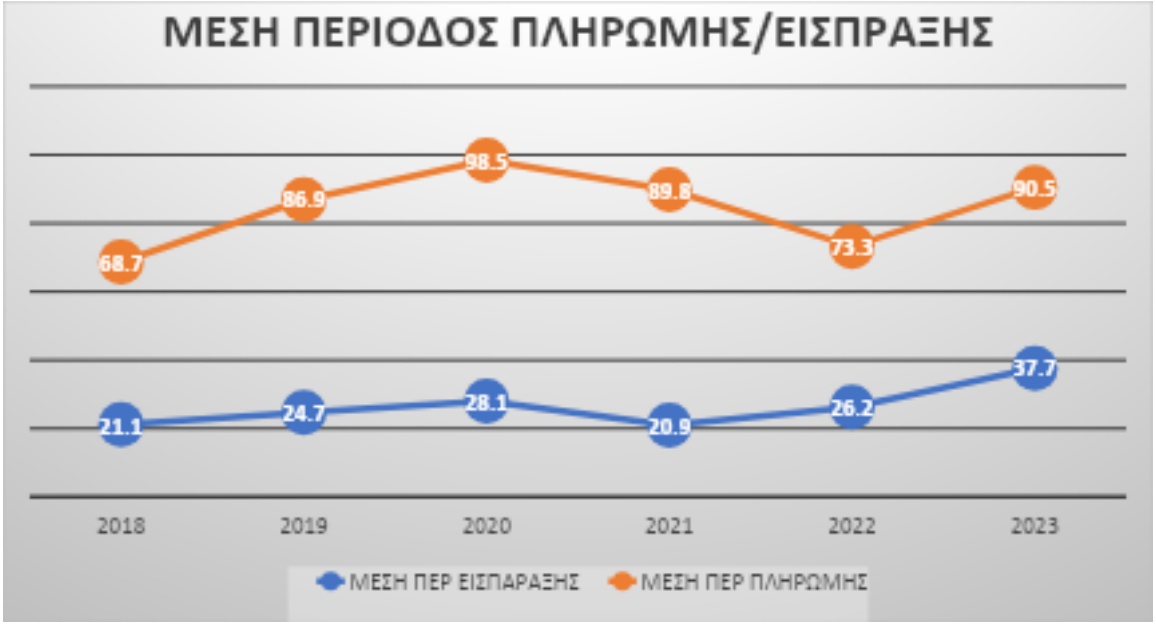
Ο πρώτος και ένας από τους σημαντικότερους δείκτες στον τομέα της εφοδιαστικής αλυσίδας ειδικά σε περιπτώσεις που δεν υπάρχει εσωτερική επιπλέον πληροφόρηση (δηλαδή αν ο αναλυτής δεν εργάζεται μέσα στην επιχείρηση και έχει στην διάθεση του δεδομένα μόνο από την οικονομική έκθεση της επιχείρησης) είναι ο δείκτης ταχύτητας αποθεμάτων ο οποίος υπολογίζεται έχοντας στον αριθμητή το κόστος πωληθέντων και στον παρανομαστη την αξία των αποθεμάτων, ο συγκεκριμένος δείκτης είναι πολύ χρήσιμος για εταιρίες λιανικών πωλήσεων όπως το πλαίσιο καθώς δείχνει πόσες φορές μέσα στο έτος ανακυκλώνεται το απόθεμα συνολικά ή διαφορετικά πόσες ημέρες(365/αριθμοδείκτη ταχύτητας αποθεμάτων) κατά μέσο ορό το σύνολο των προϊόντων μένει στα ράφια των καταστημάτων μέχρι να πουληθεί. Προφανώς θα είχε μεγαλύτερο ενδιαφέρον αυτή η ανάλυση ανά κατηγορία προϊόντων ή ακόμα ανά προϊόν (πχ ποσό γρηγορά πουλιέται το iphone16) και να γινόταν αναλύσεις όπως είδαμε στο κεφάλαιο 8 (πχ σύγκριση πωλήσεων iphone16 και Samsung s24 στο κατάστημα του πλαίσιο στα Ιωάννινα). Παρόλα αυτά ο συγκεκριμένος δείκτης δείχνει βελτίωση από το 2018 έως το 2023 καθώς το απόθεμα ανακυκλώνεται 6 φορές μέσα στο έτος δηλαδή τα προϊόντα μένουν στα <<ράφια>> κατά μέσο ορό 61 ημέρες σε αυτό έχει βοηθήσει και η ανάπτυξη της ελληνικής οικονομίας και ίσως(δεν υπάρχουν δεόμενα που να το τεκμηριώνουν) οι εκπτώσεις την περιόδου πριν τα Χριστούγεννα(black Friday κλπ.) αλλά και η καραντίνα με τον κορονοϊό(με βάση τα δεόμενα καθώς οι ημέρες πριν τον covid κυμαίνονταν στις 77 ημέρες ενώ στην διάρκεια του covid και μετα μειώθηκε κατά 10 και παραπάνω ημέρες. Προφανώς μπορεί να άλλαξε και η πολιτική της εταιρίας για παράδειγμα να εστιάζει σε προϊόντα υψηλής ζήτησης και να προμηθεύεται κωδικούς χαμηλής κινητικότητας μόνο σε περίπτωση παραγγελίας και προπληρωμής του πελάτη. Ο μέσος ορός του δείκτη για τα έξι τελευταία χρονιά είναι 70,3 ημέρες και το διάστημα εμπιστοσύνης ίσο με [61-79] ημέρες.



Ο δεύτερος αριθμοδείκτης δραστηριότητας είναι η ταχύτητα κυκλοφορίας ενεργητικού ο οποίος υπολογίζεται διαιρώντας τις πωλήσεις με το σύνολο του ενεργητικού. Το συμπέρασμα που εξάγεται είναι ποσό αποδοτικά τα στελέχη της επιχείρησης χρησιμοποιούν το ενεργητικό για την επίτευξη πωλήσεων , δηλαδή ο αριθμοδείκτης του 2023 ίσος με 1,9 μας υποδεικνύει ότι ο όμιλος πλαίσιο επιτυγχάνει πωλήσεις αξίας 1,9 ευρώ με κάθε ένα ευρώ επένδυση στο ενεργητικό της. Αρά από το παραπάνω καταλήγουμε ότι η επιχείρηση διαχειρίζεται σωστά τους πόρους, χωρίς μεγάλες αποκλίσεις με μέσο ορό 1,8 και διάστημα εμπιστοσύνης ίσο με [1,6-2].

Ο τρίτος αριθμοδείκτης αυτής της κατηγορίας είναι αυτός της ταχύτητας κυκλοφορίας παγίων ο οποίος είναι παρόμοιος με τον προηγούμενο που εξετάσαμε μόνο που σε αυτόν αντί για το σύνολο του ενεργητικό χρησιμοποιείται το σύνολο του πάγιου ενεργητικού για τον υπολογισμό του. Το 5,4 του 2023 σημαίνει ότι επενδύσεις 1 ευρώ στο πάγιο

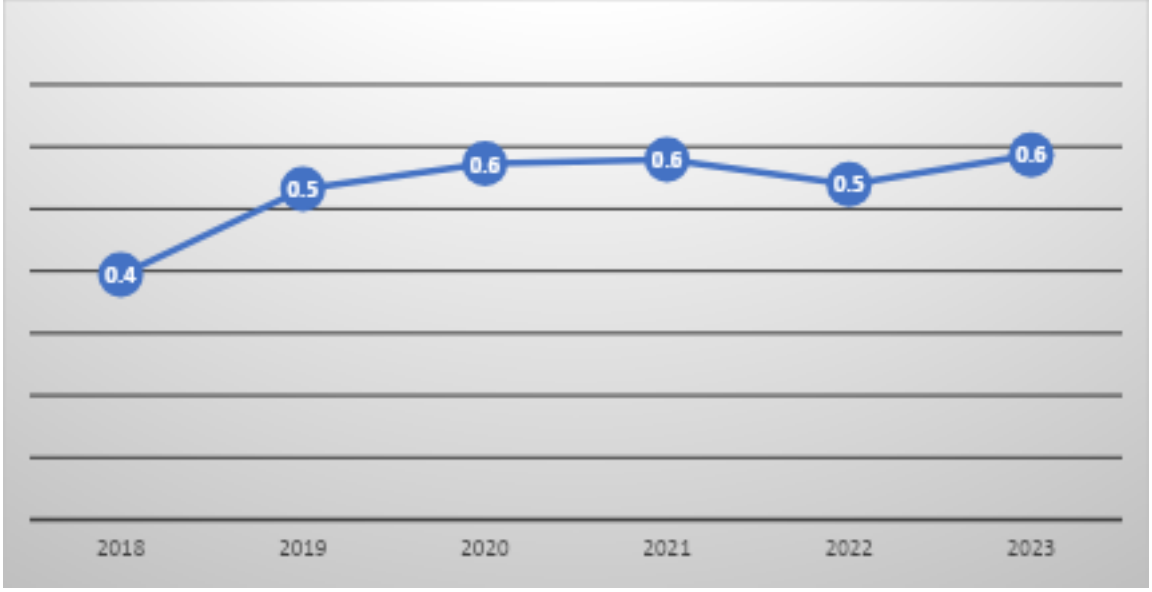
ενεργητικό της εταιρείας δημιουργούν πωλήσεις αξίας 5,4 ευρώ, δηλαδή δείχνει αυτό που είδαμε και στην πρώτη ενότητα ότι το πλαίσιο δεν κατασκευάζει ή αγοράζει κτίρια και εγκαταστάσεις για να στεγάσει τα καταστήματα του αλλά αντιθέτως τα ενοικιάζει με μακροπρόθεσμες συμφωνίες ,αλλά γενικότερα εταιρίες λιανικού εμπορίου εστιάζουν κύριος στα αποθέματα δηλαδή στο κυκλοφορούν ενεργητικό και όχι τόσο στο πάγιο. Ο μέσος ορός του είναι ίσος με 5,6 και το διάστημα εμπιστοσύνης ίσο με [4-7,2].



Συνεχίζουμε με τους αριθμοδείκτες ταχύτητας πληρωμής προμηθευτών της εταιρίας που είναι το αποτέλεσμα της διαίρεσης του συνόλου των βραχυπροθέσμων υποχρεώσεων πολλαπλασιασμένο επι 365 ημέρες έτσι ώστε το αποτέλεσμα να είναι σε ημέρες(ενώ χωρίς τον πολλαπλασιασμό το αποτέλεσμα που εξάγεται είναι σε φορές) με παρονομαστή το κόστος πωληθεντων , το αποτέλεσμα που εξάγεται είναι πόσες ημέρες θα πρέπει να περιμένει ένας προμηθευτής ή βραχυπρόθεσμος δανειστής της εταιρίας για να εισπράξει τα χρήματα που της δάνεισε ή τα χρήματα του από πωλήσεις προϊόντων με πίστωση. Για τον συγκεκριμένο δείκτη ο μέσος ορός ήταν 84,6 ημέρες ενώ το διάστημα εμπιστοσύνης που δημιουργήθηκε είναι ίσο με [72,72-96,5]. Από την άλλη πλευρά ο αριθμοδείκτης ταχύτητας είσπραξης απαιτήσεων υπολογίζεται από την διαίρεση των απαιτήσεων της εταιρίας επι 365 για να γίνει η μετατροπή του σε ημέρες δια το σύνολο των πωλήσεων, και η ερμηνεία του είναι οι ημέρες που απαιτούνται κατά μέσο ορό από την στιγμή που θα πουλήσει ένα προϊόν η εταιρία με πίστωση μέχρι να πληρωθεί (εισπράξει το σύνολο του ποσού). Όπως βλέπουμε η εταιρία εισπράττει πολύ πιο γρηγορά από ότι πληρώνει για τις υποχρεώσεις τις αρά το συμπέρασμα είναι ότι οι δανειστές της την εμπιστεύονται και τις

παρέχουν μεγάλα χρονικά διαστήματα αποπληρωμής και από την άλλη η εταιρεία όπως είδαμε και στην προηγουμένη ενότητα πουλάει τα προϊόντα της χωρίς πίστωση αλλά με μετρητά ή με χρήση πιστωτικής ή χρεωστικής κάρτας , επίσης η οικονομική έκθεση ενημερώνει τους αναλυτές προμηθευτές και ενδεχομένους επενδυτές (παρόλο που η μετοχή της εταιρίας δεν διαπραγματεύεται στο χρηματιστήριο) ότι οι απαιτήσεις από το 2022 στο 2023 αυξήθηκαν κύριος λόγο του επιδοτούμενου προγράμματος αλλάζω συσκευή και μεγάλο μέρος των πόσων δεν έχει καταβληθεί από το δημόσιο στην εταιρία αλλά προφανώς δεν υπάρχει πρόβλημα αθετήσεις των υποχρεώσεων. Σχετικά με τον μέσο ορό του δείκτη ταχύτητας είσπραξης απαιτήσεων αυτός διαμορφώθηκε στις 26,4 ημέρες και το διάστημα εμπιστοσύνης είναι ίσο με [20-33] ημέρες.

Οι επόμενοι δυο αριθμοδείκτες ανήκουν στην κατηγορία της ανάλυσης του χρέους της εταιρίας ή πιο απλά της μοχλεύσης , πιο συγκεκριμένα δείχνουν το ποσοστό στο οποίο συμμετέχει η εταιρεία με δικά της κεφάλαια και κατά ποσό με ξένα κεφάλαια τρίτων(τράπεζων, προμηθευτών κλπ.) για να χρηματοδοτήσει της επενδύσεις της.

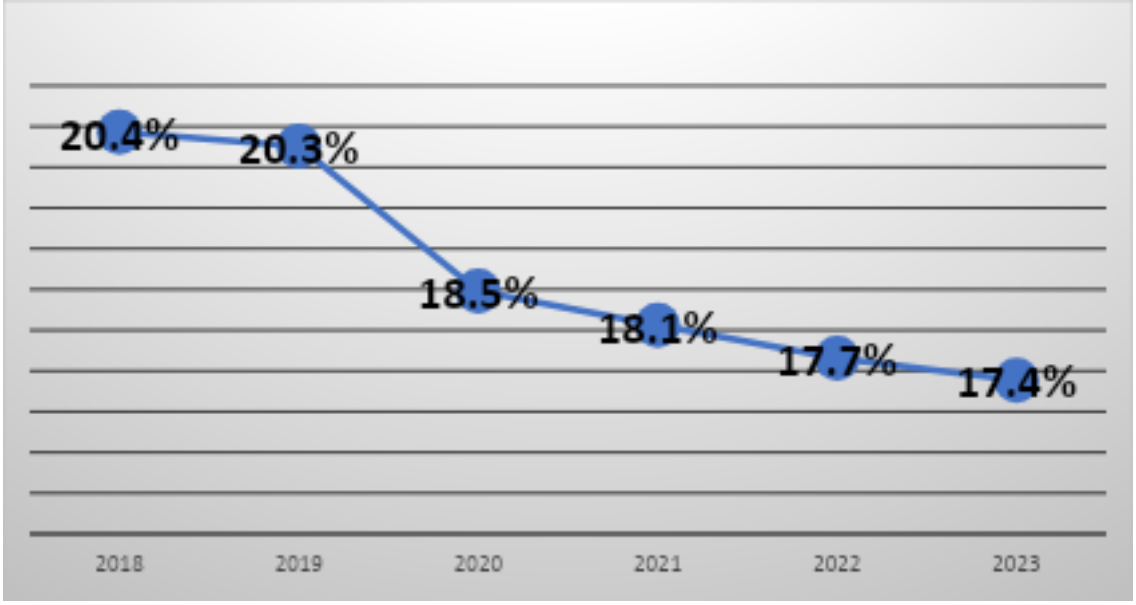


Ο πρώτος αριθμοδείκτης αυτής της κατηγορίας είναι ο δείκτης δανειακής επιβάρυνσης ο οποίος υπολογίζεται διαιρώντας τα ξένα κεφάλαια (βραχυπρόθεσμες συν μακροπρόθεσμες υποχρεώσεις) με το σύνολο των κεφαλαίων που έχουν επενδυθεί στο ενεργητικό. Ο δείκτης αυτός πρέπει να ερμηνευθεί και από την πλευρά των ιδιοκτήτων της εταιρίας οι οποίοι θέλουν το μεγαλύτερο μέρος των επενδύσεων να γίνεται με ξένα κεφάλαια και από

την άλλη πλευρά τους δανειστές οι οποίοι θέλουν το παραπάνω ποσοστό όσο το δυνατόν μικρότερο καθώς σε περίπτωση χρεοκοπίας της εταιρίας να πάρουν τα χρήματα τους πίσω μέσο των ιδίων κεφαλαίων. Ο μέσος ορός για τον υστερημένο δείκτη είναι ίσος με 0,5 δηλαδή τα στοιχεία του ενεργητικού έχουν αποκτηθεί με ισόποσες επενδύσεις των μέτοχων και των δανειστών της εταιρίας ενώ το διάστημα εμπιστοσύνης έχει διαμορφωθεί στο [0,46-0,6].

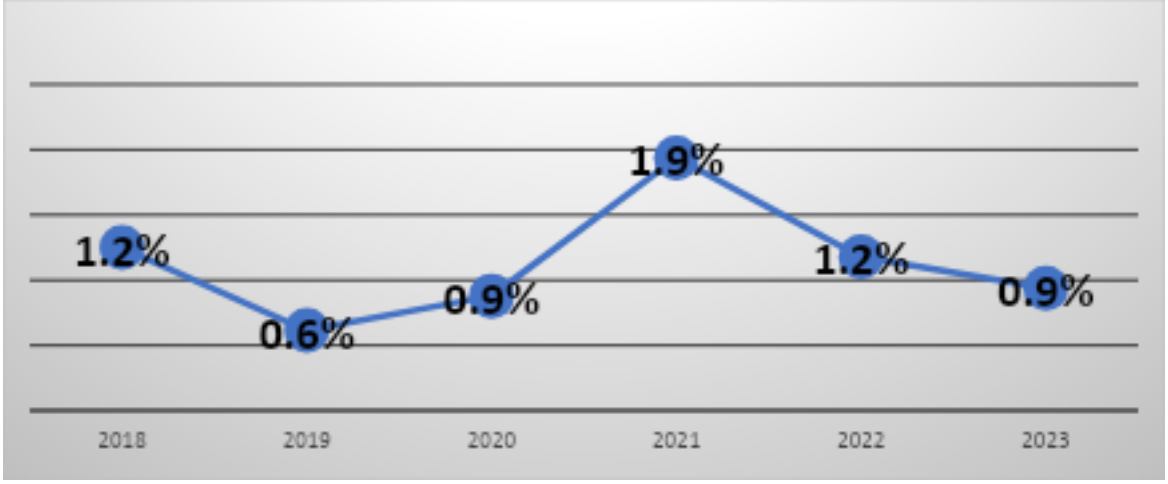
Ο δεύτερος δείκτης μοχλεύσεις είναι ο ξένα προς ιδιά κεφάλαια ο οποίος είναι αναπροσαρμογή του πρώτου και δείχνει πόσες φορές περισσότερα είναι τα ξένα κεφάλαια προς τα ιδιά κεφάλαια. Για το πλαίσιο ο συγκεκριμένος δείκτης αυξάνεται τα τελευταία χρονιά το οποίο θα μπορούσε να πει κάνεις ότι είναι αποτέλεσμα της εξόδου της Ελλάδας από τα capital controls (ευκολότερη πρόσβαση σε δίφορου μορφής δάνειων) , της ανόδου της ζήτησης για προϊόντα που ωθεί την εταιρεία να αποκτήσει ρευστά διαθέσιμα για να ανταποκριθεί σε αυτή την άνοδο της αγοράς. Ο μέσος ορός των τελευταίων έξι ετών ήταν ίσος με 1,2 φορές δηλαδή τα ξένα κεφάλαια ήταν λίγο μεγαλύτερα από τα χρήματα που έχουν επενδύσει οι ιδιοκτήτες της εταιρίας και το διάστημα εμπιστοσύνης διαμορφώθηκε στο [0,9-1,5] φορές.

Στην συνέχεια γίνεται ανάλυση μερικών αριθμοδεικτών απόδοσης δηλαδή αριθμοδεικτών που δείχνουν ποσό αποδοτικά διαχειρίζονται τα στελέχη της επιχείρησης συνολικά της εφοδιαστική αλυσίδας από την επιλογή προμηθευτή την μεταφορά των προϊόντων τα έξοδα διαφήμισης μισθών ενοικίων μέχρι την τελική πώληση των προϊόντων και την δημιουργία κερδών.

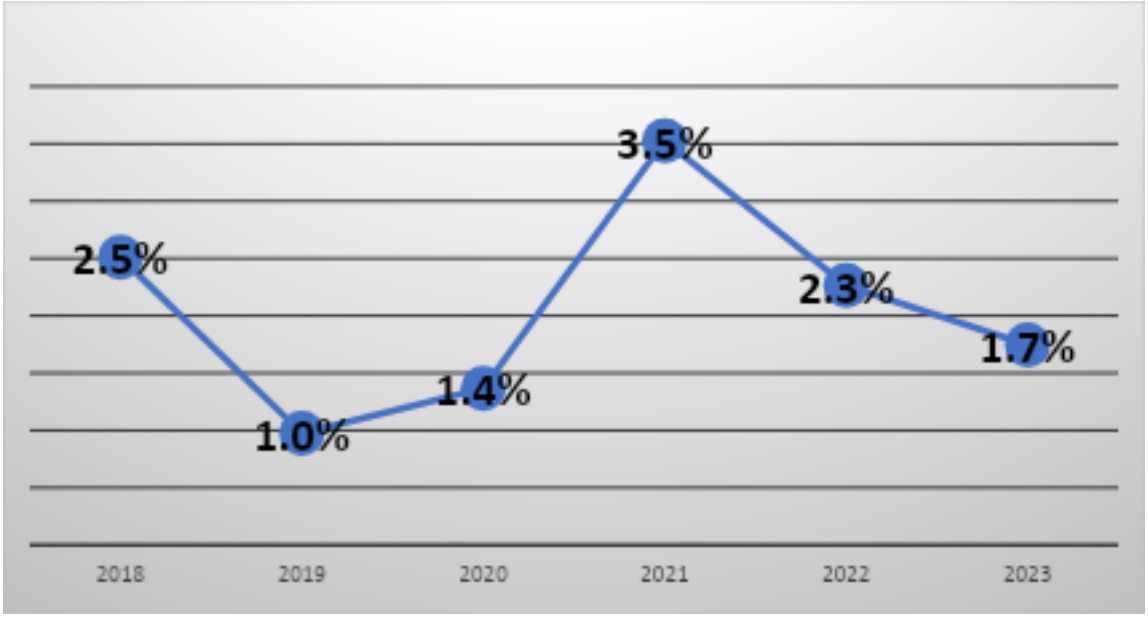


Ο πρώτος που θα εξετάσουμε είναι ο αριθμοδείκτης περιθωρίου μικτού κέρδους ο οποίος υπολογίζεται έχοντας στον αριθμητή το σύνολο των πωλήσεων μείον το κόστος πωληθεντων (δηλαδή όλα εκείνα τα εξάδα για την αγορά και μεταφορά των προϊόντων από τον προμηθευτή μέχρι την αποθήκη της επιχείρησης δηλαδή συμπεριλαβαίνετε το κόστος αγοράς του εμπορεύματος το μεταφορικό κόστος πχ μεταφορά κινητών apple με container πλοίο από την Ασία στην Ελλάδα μέσα σε αυτό μπορεί να συμπεριλαμβάνονται και κόστη όπως οι δασμοί και τα κόστη ασφάλισης των εμπορευμάτων κατά την μεταφορά , επίσης συμπεριλαμβάνονται και ειδικά κόστη τοποθέτησης των προϊόντων στην αποθήκη κατά την παραλαβή ειδικά όταν μιλάμε για προϊόντα όπως τις λευκές συσκευές που χιάζονται ιδική μεταχείριση(λόγο βάρους και όγκου) στα παραπάνω κόστη προφανώς δεν συμπεριλαμβάνονται λειτουργικά έξοδα όπως μισθοί εργαζομένων ή έξοδα όπως η κατανάλωση ρεύματος και τα ενοίκια των καταστημάτων) δια το σύνολο των πωλήσεων. Η παραπάνω πράξη ερμηνεύεται ως η αποτελεσματικότητα της διοίκησης της εταιρίας στον να μπορεί να μειώσει το κόστος πωληθεντων από την μια πλευρά (κάτι που είναι δύσκολο σε εταιρίες λιανικής πώλησης και επιτυγχάνεται μόνο με μακροχρόνιες σχέσεις με τους προμηθευτές και ίσως με παραγγελίες μεγάλων ποσοτήτων) και από την άλλη να πουλάει τα προϊόντα της σε υψηλότερη τιμή από ότι τα αγόρασε αποφεύγοντας δηλαδή τις εκπτώσεις και τις μειώσεις τιμών (και ταυτόχρονα χωρίς τα προϊόντα να μένουν στα ράφια και να χάνουν της αξία τους) προφανώς και αυτό είναι δύσκολο λόγο του εντόνου ανταγωνισμού στον κλάδο και ίσως μπορεί να βελτιωθεί μέσο της μάρκας της εταιρίας και της μακροχρόνιας εμπιστοσύνης του καταναλωτή προς την εταιρία. Όπως βλέπουμε η χρονοσειρα του δείκτη έχει πτωτική πορεία κάτι που αναδεικνύει τα προηγούμενα

(δυσκολία μείωσης κόστους πωληθεντων και υψηλός ανταγωνισμός στο κλάδο που οδηγεί σε επιπλέον έξοδα μάρκετινγκ και εκπτώσεις) ο μέσος ορός των τελευταίων έξι ετών διαμορφώθηκε στο 18,5% και το διάστημα εμπιστοσύνης είναι [17%-20%].

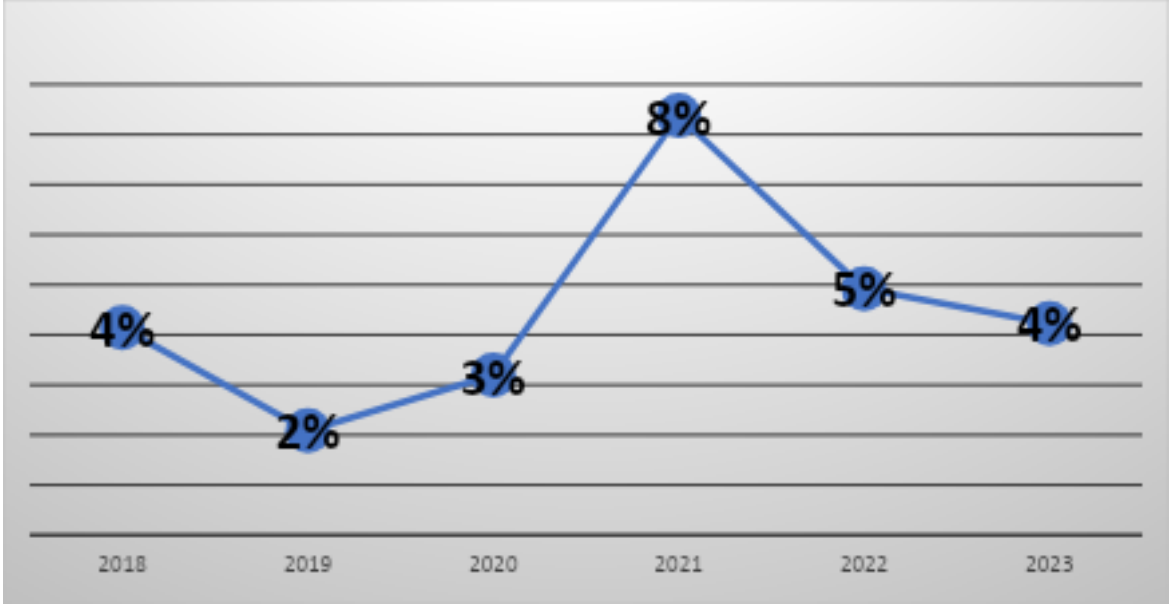


Ο δεύτερος αριθμοδείκτης αυτής της κατηγορίας είναι του περιθώριού καθαρού κέρδους ο οποίος υπολογίζεται διαιρώντας τα καθαρά κέρδη με το σύνολο των πωλήσεων , και δείχνει ποσό καλά διαχειρίζεται η εταιρία συνολικά τα έσοδα της (πωλήσεις) σε σχέση με τα καθαρά κέρδη της δηλαδή το ποσό που προκύπτει εφόσον αφαιρεθούν το κόστος πωληθεντων τα λειτουργικά έξοδα οι φόροι οι τόκοι δάνειων και οι αποσβέσεις, και το αποτέλεσμα της πράξης είναι το ποσοστό καθαρού κέρδους που μένει στην εταιρία δηλαδή για το 2023 είναι σχεδόν 1% αυτό σημαίνει ότι εάν το πλαίσιο πουλήσει προϊόντα αξίας 1000 ευρώ το κέρδος του θα είναι ίσο με 10 ευρώ. Ο μέσος ορός του δείκτη είναι ίσος με 1,2% και το διάστημα εμπιστοσύνης ίσο με [0,7%-1,6%].



Ένας από τους πιο γνωστούς αριθμοδείκτες είναι ο roa ο οποίος δείχνει την απόδοση του ενεργητικού της εταιρίας στην δημιουργία κερδών , ο roa υπολογίζεται διαιρώντας τα καθαρά κέρδη με το σύνολο του ενεργητικού δηλαδή για να αυξηθεί ο roa θα πρέπει η επιχείρηση με όσο το δυνατό μικρότερες επενδύσεις στο ενεργητικό να παράγει όσο το δυνατόν περισσότερα κέρδη, γεγονός το οποίο είναι αρκετά δύσκολο στο κλάδο του λιανικού εμπορίου καθώς απαιτούνται επενδύσεις μεγάλων ποσών σε αποθέματα αποθήκες μηχανήματα και κτίρια για την στέγαση των καταστημάτων. Όπως είδαμε και παραπάνω το πλαίσιο προσπαθεί αντί να αγοράσει τους χώρους που στεγάζουν τα καταστήματα να τα ενοικιάζει με μακροχρόνιες συμβάσεις, προφανώς το να αλλάξει στρατηγική και να πουλάει προϊόντα έχοντας πρώτα πραγματοποιηθεί παραγγελία από τον πελάτη για όλα τα προϊόντα αλλάζει ριζικά το μοντέλο της επιχείρησης το οποίο είναι εστιασμένο στην επαφή με τον πελάτη στον χώρο των καταστημάτων, παρόλα αυτά θα μπορούσε ίσως να γίνονται προβλέψεις της ζήτησης ανά προϊόν και οποίο προϊόν δεν έχει πάνω από ένα χ ποσό ζήτησης ανά μηνά να μην διατίθεται στο κατάστημα πάρα μόνο μέσο παραγγελίας και προπληρωμής του πελάτη. Γενικότερα ισχύουν όλα που αναφέρθηκαν προηγούμενος με τον υψηλό ανταγωνισμό και σταθερές τιμές αγοράς εμπορευμάτων από τους προμηθευτές που έχουν ως αποτέλεσμα ο δείκτης τα τελευταία έξη χρονιά να έχει μέσο ορό ίσο με 2,1% δηλαδή μια επένδυση 100 ευρώ στο ενεργητικό της εταιρίας οδηγεί σε καθαρά κέρδη αξίας 2,1 ευρώ και το διάστημα εμπιστοσύνης διαμορφώθηκε στο [1,1%-3%].

Στην συνέχεια για τον υπολογισμό του αριθμοδείκτη roe θα πρέπει πρώτα να υπολογιστεί ο πολλαπλασιαστής μοχλεύσεις ο οποίος είναι το αποτέλεσμα της πράξης συνολικά περιουσιακά στοιχεία ενεργητικού διαιρώντας τα με τα ιδιά κεφάλαια. Αυτός ο δείκτης μοιάζει με το δείκτη ξένα προς ιδιά κεφάλαια που είδαμε πιο πάνω μόνο που αυτός εστιάζει στο κατά ποσό τα στοιχεία του ενεργητικού έχουν χρηματοδοτηθεί από τα ιδιά κεφάλαια. Γενικά αν ο Πολλαπλασιαστής μιας εταιρίας είναι κοντά στο 1 σημαίνει ότι η επιχείρηση έχει χρηματοδοτήσει το ενεργητικό της αποκλειστικά και μόνο μέσο δικών της κεφαλαίων κάτι που την καταστεί πιο ασφαλή καθώς δεν χρειάζεται να ανησυχεί για αποπληρωμές προμηθευτών τόκων δάνειων κλπ. αλλά από την άλλη δεν της επιτρέπει να αναπτυχθεί με μεγαλύτερο ρυθμό χρησιμοποιώντας ξένα κεφάλαια. Από την άλλη ένας μεγάλος πολλαπλασιαστής σημαίνει ότι μεγάλο μέρος του ενεργητικού έχει αγοραστεί με ξένα κεφάλαια που οδηγούν σε περισσότερα έξοδα όπως έξοδα τόκων δάνειων. Για το πλαίσιο ο μέσος ορός των έξη ετών είναι ίσος με 2,2 δηλαδή οι επενδύσεις ξένων κεφαλαίων για την αγορά του ενεργητικού της είναι σχεδόν διπλάσιες σε σχέση με τις επενδύσεις των ιδιοκτήτων της. Το διάστημα εμπιστοσύνης που προκύπτει είναι ίσο με [1,9%-2,5%].



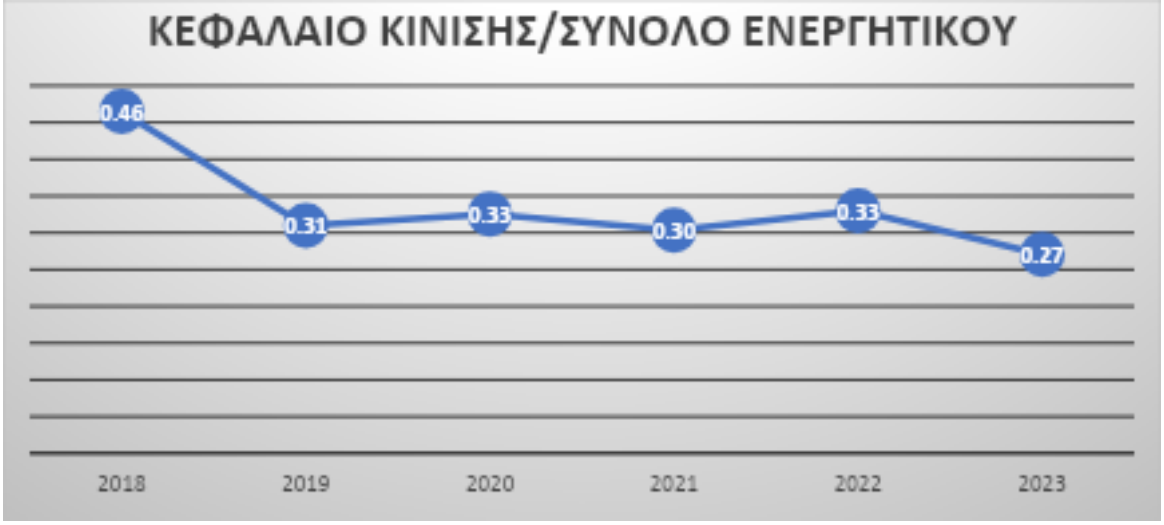
Έχοντας υπολογίσει τον roa και τον πολλαπλασιαστή μοχλεύσεις μπορούμε να υπολογίσουμε πολλαπλασιάζοντας αυτούς τους δυο δείκτες και να προκύψει το roe δηλαδή τον δείκτη απόδοσης ιδίων κεφαλαίων (ή διαφορετικά μπορούμε να διαιρέσουμε τα καθαρά κέρδη με τα ιδιά κεφάλια και το αποτέλεσμα θα είναι το ίδιο). Ο συγκεκριμένος δείκτης για το πλαίσιο είναι 4% το οποίο σημαίνει ότι για κάθε 100 ευρώ επένδυσης των

ιδιοκτήτων της επιχείρησης η εταιρία παράγει 4 ευρώ καθαρά κέρδη, ο συγκεκριμένος δείκτης είναι σχετικά χαμηλός εξαιτίας του χαμηλού roa καθώς η μοχλευση δεν είναι ούτε πολύ μεγάλη ούτε πολύ μικρή και ουσιαστικά αυτή ανεβάζει τον δείκτη. Επίσης βλέπουμε ότι το 2021 το έτος δηλαδή της καραντίνας ο συγκεκριμένος δείκτης πείρε την υψηλότερη τιμή του μέσα στα τελευταία έξη χρονιά. Ο μέσος ορός για τον συγκεκριμένο δείκτη είναι ίσος με 4% και το διάστημα εμπιστοσύνης ίσο με [2%-7%].

Τέλος ολοκληρώνουμε την ανάλυση μας με τον υπολογισμό του z score για το πλαίσιο του Edward Altman. Το z score αναπτύχθηκε το 1968 και είναι ένας δείκτης που υποδεικνύει την πιθανότητα χρεοκοπίας μιας εταιρείας πιο συγκεκριμένα ένα z-score μικρότερο του 1,8 σημαίνει ότι η εταιρία είναι στα πρόθυρα της χρεοκοπίας, δείκτης μεταξύ 1,81 και 2,99 σημαίνει ότι η επιχείρηση είναι σε μια μεσαία κατάσταση ως προς την χρεοκοπία της ενώ ένας δείκτης μεγαλύτερος ή ίσος του 3 σηματοδοτεί ότι η επιχείρηση έχει μικρή πιθανότητα χρεοκοπίας. Ο τρόπος υπολογισμού του είναι μέσο της εξίσωσης

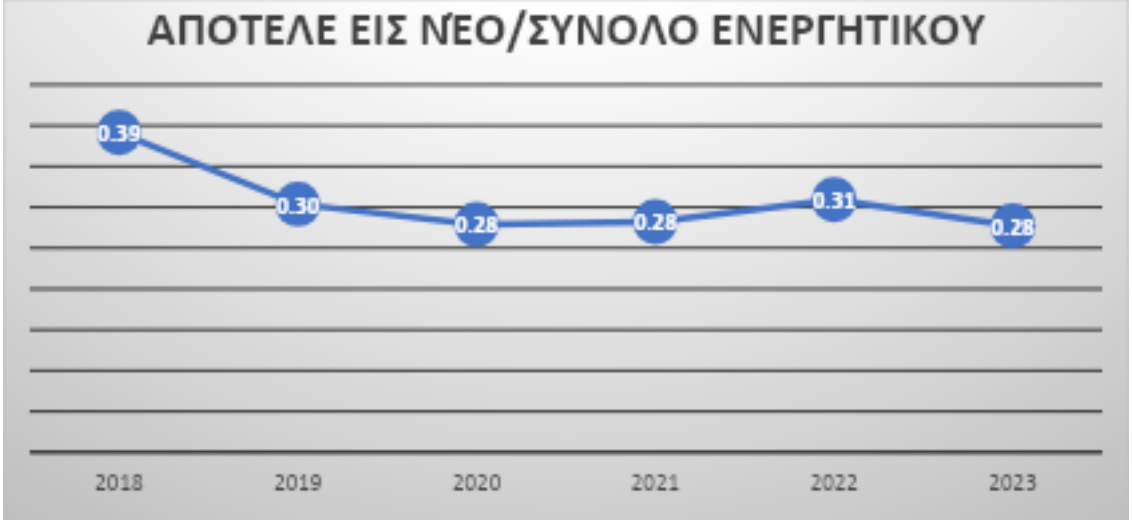
$$zscore = 1,2 * x1 + 1,4 * x2 + 3,3 * x3 + 0,6 * x4 + 1,* x5$$

Ο κάθε αριθμοδείκτης από το χ1 έως χ5 αναλύεται παρακάτω.

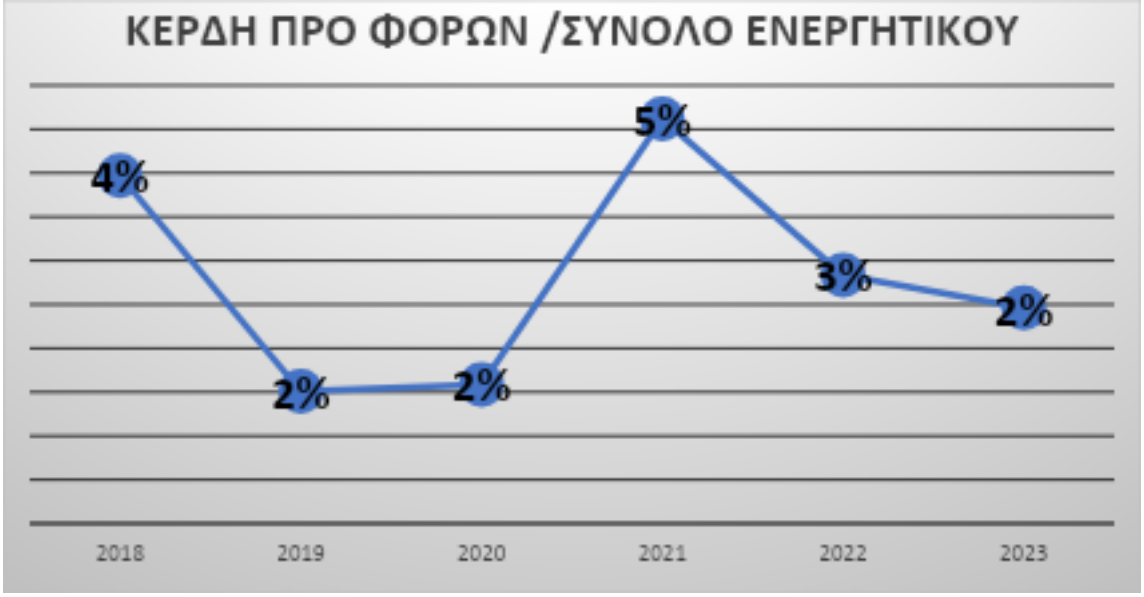


Ο αριθμοδείκτης χ1 του z-score του Altman υπολογίζεται διαιρώντας το κεφάλαιο κίνησης (δηλαδή αφαιρώντας από το κυκλοφορούν ενεργητικό τις βραχυπρόθεσμες υποχρεώσεις) με το σύνολο του ενεργητικού . ο αριθμοδείκτης αυτός δείχνει την συνολική ρευστότητα της εταιρίας σε σχέση με το σύνολο του ενεργητικού της καθώς το κεφάλαιο κίνησης είναι το ποσό των διαθέσιμων περιουσιακών στοιχείων που περισσέψουν στην επιχείρηση αφότου έχει καλύψει τις βραχυπρόθεσμες υποχρεώσεις της. Στην περίπτωση του πλαίσιο

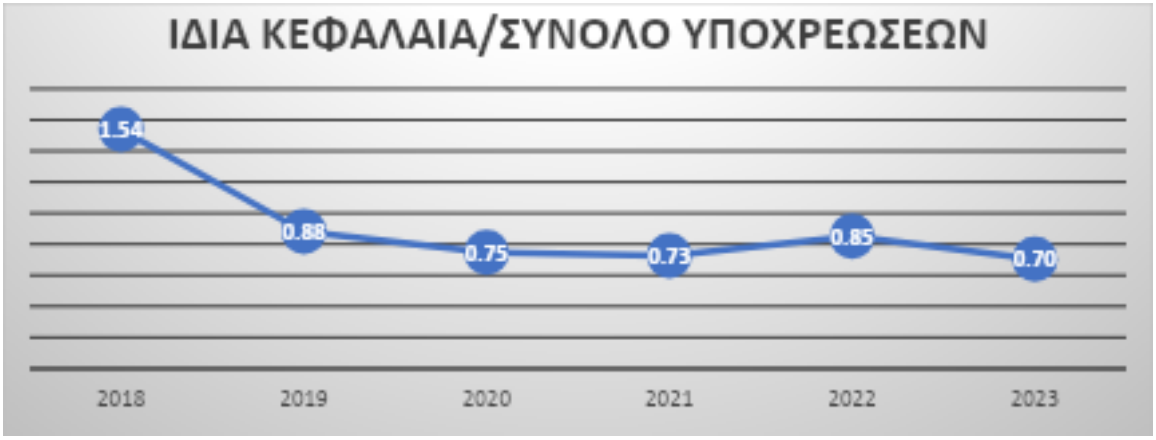
το κεφάλαιο κίνησης μπορεί να καλύψει τις βραχυπρόθεσμες υποχρεώσεις τουλάχιστον μια φορά καθώς ο δείκτης είναι μεγαλύτερος από το μηδέν. Ο μέσος ορός του δείκτη για τα τελευταία έξι χρονιά είναι ίσος με 0,33 και το διάστημα εμπιστοσύνης ίσο με [0,26-0,4].



Ο αριθμοδείκτης χ2 υπολογίζεται διαιρώντας τα αποτελέσματα εις νέο της επιχείρησης δηλαδή τα κέρδη που έχει συγκεντρώσει η επιχείρηση με την πάροδο των χρόνων και δεν έχει διανείμει στους μέτοχους της με το σύνολο των περιουσιακών της στοιχείων. Ο δείκτης αυτός δείχνει τι ποσοστό του ενεργητικού της εταιρίας χρηματοδοτηθεί από τα αδιανέμητα κέρδη της, και κατά επέκταση δείχνει κατά ποσό η επιχείρηση στηρίζεται σε δικούς της πόρους για επενδύσεις στο ενεργητικό της σε σχέση με τα ξένα κεφάλαια(τραπεζικά δάνεια, υποχρεώσεις σε προμηθευτές κλπ.). Για το πλαίσιο ο δείκτης είναι ίσος με 0,28 για το 2023 που σημαίνει ότι το 28% των περιουσιακών του έχει αγοραστεί με αδιανέμητα κέρδη προηγουμένων οικονομικών ετών. Ο μέσος ορός είναι ίσος με 30% και το διάστημα εμπιστοσύνης ίσο με [26%-35%].

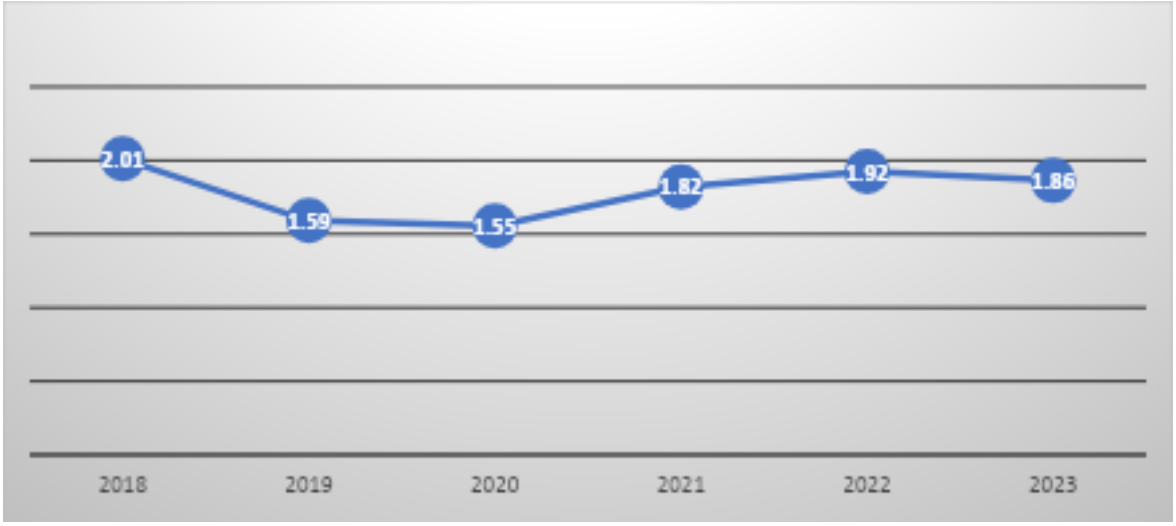


Το χ3 μοιάζει αρκετά με τον roa μόνο που αντί να διαιρέσουμε τα συνολικά κέρδη μετα φορών με το σύνολο του ενεργητικού εδώ βάζουμε στον αριθμητή τα κέρδη προ φορών και τόκων (ebit). Ο συγκεκριμένος αριθμοδείκτης δείχνει ποσό αποδοτικά χρησιμοποιούνται τα περιουσιακά στοιχεία της εταιρίας για την δημιουργία κερδών πριν από φόρους και τόκους για το πλαίσιο ο αριθμοδείκτης είναι ίσος με 2% καθώς όπως έχουμε ήδη αναφέρει υπάρχει υψηλό κόστος αγοράς αποθεμάτων και υψηλός ανταγωνισμός με αποτέλεσμα κάθε 100 ευρώ επενδύσεων στο ενεργητικό της εταιρίας να δημιουργεί κέρδη αξίας 2 ευρώ. Ο μέσος ορός είναι ίσος με 3% και το διάστημα εμπιστοσύνης είναι ίσο με [2%-4%].



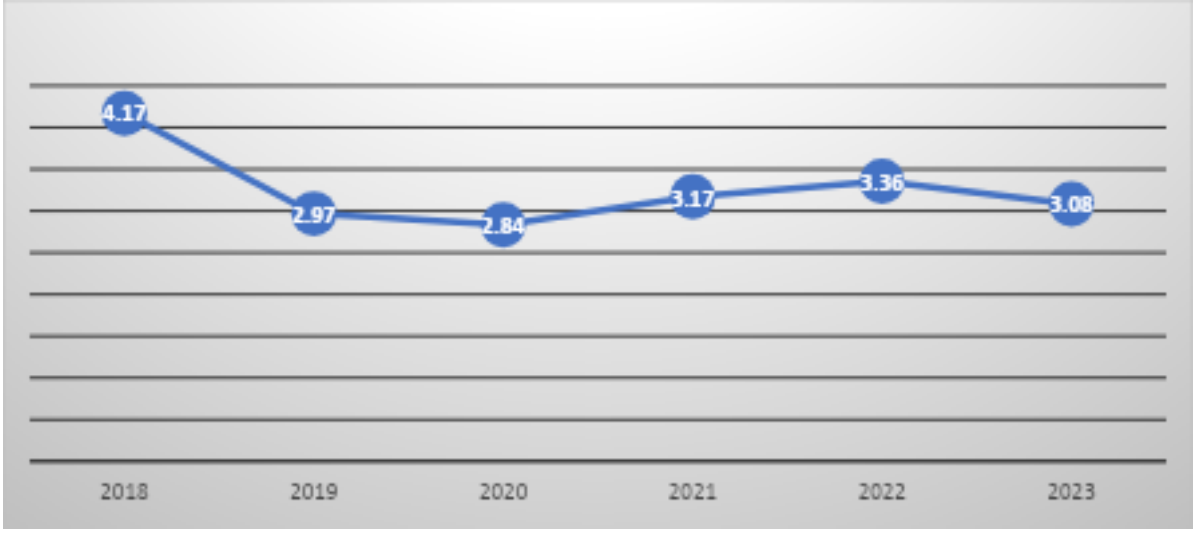
Ο χ4 αριθμοδείκτης υπολογίζεται διαιρώντας το σύνολο των ιδίων κεφαλαίων με το σύνολο των υποχρεώσεων (μακροπρόθεσμες και βραχυπρόθεσμες υποχρεώσεις) και δείχνει την χρηματοοικονομική μοχλευση της εταιρίας δηλαδή κατά ποσό έχουν συνεισφέρει στην εταιρία οι μέτοχοί και κατά ποσό οι δανειολήπτες της. Όσο μεγαλύτερος

είναι αυτός ο δείκτης τόσο περισσότερο η εταιρία χρηματοδοτεί της υποχρεώσεις της με δικά της κεφάλαια, επίσης οι δανειστές της όπως είχαμε αναφέρει και στην ανάλυση τον δεικτών μοχλεύσεις θέλουν αυτός ο δείκτης να είναι όσο το δυνατόν μεγαλύτερος καθώς σε περίπτωση χρεοκοπίας της εταιρίας να υπάρχουν ιδιά κεφάλαια ώστε να αποπληρωθούν. Όπως παρατηρούμε το πλαίσιο δεν εξαρτάται σε μεγάλο βαθμό από τα ξένα κεφάλαια καθώς το 70% των υποχρεώσεων χρηματοδοτούνται από ιδιά κεφάλαια για το έτος 2023 και μάλιστα το 2018 οπού οι ο μακροπρόθεσμος δανεισμός ήταν μικρός τα ιδιά κεφάλαια ήταν περισσότερα από της υποχρεώσεις της στους δανειστές της. Ο μέσος ορός του ομίλου ήταν ίσος με 90% και το διάστημα εμπιστοσύνης ίσο με [57%-124%].



Ο τελευταίος αριθμοδείκτης για τον υπολογισμό του zscore είναι το χ5 το οποίο είναι ακριβός ίδιο με τον αριθμοδείκτη κυκλοφοριακής ταχύτητας ενεργητικού που έχει υπολογιστεί και αναλυθεί ήδη.

Έχοντας υπολογίσει όλα τα παραπάνω είμαστε σε θέση να εφαρμόσουμε τον τύπο του z score για τα έξη τελευταία οικονομικά έτη λειτουργίας του πλαίσιο και να οδηγηθούμε στο συμπέρασμα ότι ο όμιλος έχει μικρή έως ελάχιστη πιθανότητα να χρεοκοπήσει στο βραχυπρόθεσμο μέλλον καθώς το zscore για το 2023 είναι ίσο με 3,08 μεγαλύτερο από το 2,99 που αναφέραμε στην αρχή. Ο μέσος ορός των έξη ετών είναι ίσος με 3,26 και το διάστημα εμπιστοσύνης διαμορφώθηκε στο [2,76-3,76].



7.3 ΠΡΟΒΛΕΨΗ ΠΩΛΗΣΕΩΝ ΚΑΙ ΚΟΣΤΟΥΣ ΠΩΛΗΘΕΝΤΩΝ ΜΕ ΤΙΣ ΜΕΘΟΔΟΥΣ ΧΡΟΝΟΛΟΓΙΚΩΝ ΣΕΙΡΩΝ ΑΠΟ ΤΟ ΚΕΦΑΛΑΙΟ 7

Σε αυτή την ενότητα θα αναλύσουμε και θα προβλέψουμε με βάση τις μεθοδολογίες που αναπτυχθήκαν στο κεφάλαιο 7 τις πωλήσεις και το κόστος πωληθέντων της εταιρίας πλαίσιο. ο κώδικας είναι για την ανάπτυξη των μοντέλων μηχανικής μάθησης είναι παρόμοιος με αυτόν του κεφαλαίου 7 για αυτόν τον λόγο δεν θα δοθεί έμφαση στην ανάπτυξη του και απλά θα παρουσιαστούν τα αποτελέσματα απευθείας.

Τα δεδομένα των πωλήσεων και του κόστους πωληθέντων αποκτήθηκαν μέσα από τους αναρτημένους ισολογισμούς της εταιρίας όπως και με τα δεδομένα για τους υπολογισμούς των αριθμοδεικτών που είδαμε στην προηγουμένη ενότητα. Το πρόβλημα που παρουσιάστηκε κατά την συγκέντρωση των δεδομένων για την είσοδο τους στα μοντέλα χρονολογικών σειρών ήταν ότι η εταιρία το 2023 οπού και η μετοχή της δεν διαπραγματευόταν στο χρηματιστήριο Αθηνών δεν ήταν υποχρεωμένη να εκδίδει εξαμηνιαία τις οικονομικές της καταστάσεις με αποτέλεσμα να δημιουργείτε ένα κενό στην χρονοσειρα δηλαδή υπάρχουν οι συνολικές πωλήσεις της εταιρίας για το 2023 αλλά δεν υπάρχουν οι αντίστοιχες εξαμηνίες που με βάση αυτές είχε δημιουργηθεί η χρονοσειρα καθώς όπως γνωρίζουμε από τα προηγούμενα κεφάλαια όσο περισσότερα είναι τα timesteps δηλαδή τα δεδομένα μας χρονοσειρας τόσο καλυτέρα τα μοντέλα μπορούν να αναγνωρίσουν μοτίβα και να καταλήξουν σε καλύτερες προβλέψεις και εκτιμήσεις των

μελλοντικών τιμών της χρονοσειρας. Για αυτό τον λόγο εισάγουμε αρχικά τις εξαμηνίες πωλήσεις και τα αντίστοιχα κόστη πωληθέντων από το 2012 μέχρι το 2022 και χρησιμοποιούμε κάποια από τα μοντέλα που είδαμε στο κεφάλαιο 7 ώστε να προβλέψουμε τις εξαμηνίες πωλήσεις του 2023.

time	sales	costs
30/6/2012	63600	49577
31/12/2012	223276	172875
30/6/2013	62456	48220
31/12/2013	220283	165730
30/6/2014	70494	53497
31/12/2014	227054	170982
30/6/2015	63005	49612
31/12/2015	208980	161180
30/6/2016	132281	104090
31/12/2016	150709	118429
30/6/2017	127838	100248
31/12/2017	158260	123716
30/6/2018	137035	108704
31/12/2018	171823	137044
30/6/2019	137523	108273
31/12/2019	179626	144629
30/6/2020	148566	120125
31/12/2020	206068	168970
30/6/2021	199219	165269
31/12/2021	237666	192704
30/6/2022	192182	157161
31/12/2022	242499	200781

Στην συνέχεια εισήγαμε τα δεδομένα στην python ως pandas dataframe και εκτελέσαμε τα τρία μοντέλα χωρίς να διαφοροποιήσουμε ή να πάρουμε τις λογαριθμυμενες πωλήσεις και κόστη πωληθεντων , δηλαδή δεν έγινε κάποια μετατροπή(transformation) των δεδομένων. παρόλο που τα δεδομένα δεν είναι στασιμά με βάση το dickey fuller test.

```
from statsmodels.tsa.stattools import adfuller
result = adfuller(df['sales'])
print('ADF Statistic: %f' % result[0])
print('p-value: %f' % result[1])
print('Critical Values:')
for key, value in result[4].items():
    print('\t%s: %.3f' % (key, value))
if result[1] <= 0.05:
    print("Strong evidence against the null hypothesis. Time series is likely stationary.")
else:
    print("Weak evidence against the null hypothesis. Time series is likely non-stationary.")
```

```
ADF Statistic: 0.346492
p-value: 0.979380
Critical Values:
        1%: -3.833
        5%: -3.031
        10%: -2.656
Weak evidence against the null hypothesis. Time series is likely non-stationary.
```

Εικονα 191:Έλεγχος στασιμότητας

Στην συνέχεια τα μοντέλα που αναπτυχθήκαν ήταν το Arima(1,1,0) με mse=2.821.366.176,9 για όλο το dataset δηλαδή δεν έγινε διαχωρισμός σε train και τεστ data καθώς οι παρατηρήσεις είναι μόνο 22 και οι πωλήσεις που πρόβλεψε ήταν για το πρώτο εξάμηνο ίσες με 2023-06-30 =200.528,9 και για το δεύτερο εξάμηνο 2023-12-31 235536.7 να υπενθυμίσουμε ότι οι συνολικές πωλήσεις για το 2023 ήταν 468.778 οπότε η πρόβλεψη έπεσε έξω κατά 32692,4 μονάδες δηλαδή δεν υπάρχει και τόσο μεγάλη απόκλιση από την πραγματική τιμή επίσης το μοντέλο αντιλήφθηκε πολύ σωστά ότι οι πωλήσεις το πρώτο εξάμηνο είναι διαχρονικά μικρότερες από τις πωλήσεις του δευτέρου εξάμηνου λόγο των υψηλότερων πωλήσεων που δημιουργούνται από τις εκπτώσεις του black Friday του cyber Monday αλλά και γενικότερα την υψηλότερη ζήτηση για προϊόντα τεχνολογίας την περίοδο των γιορτών των Χριστουγέννων αλλά και της υψηλές πωλήσεις και του μηνά Σεπτέμβρη οπού το πλαίσιο έχει στο κατάστημα του πλήθος οπό σχολικά προϊόντα όπως τσάντες ,γόμες ,σχολικά βοηθήματα ,ξενόγλωσσα βιβλία αλλά και γραφεία ,καρέκλες γραφείου για τα σπίτια πρωτοετών φοιτητών που πηγαίνουν να σπουδάσουν σε πόλεις διαφορετικές από τον τόπο κατοικίας τους. Η επόμενη μέθοδος που χρησιμοποιήθηκε για την πρόβλεψη των πωλήσεων ήταν το Arma(3,0,3) με μέσο τετραγωνικό σφάλμα ίσο με 916.730.648.8 πολύ μικρότερο από το αντίστοιχο του Arima(1,1,0) μοντέλου και οι αντίστοιχες προβλέψεις για τις εξαμηνιαίες πωλήσεις του 2023 ήταν 2023-06-30 = 213.169.7 και για το δεύτερο εξάμηνο ίσες με 2023-12-31= 240.996 που δημιουργεί μια απόκλιση των 14.612,3 πωλήσεων από τις πραγματικές ετήσιες πωλήσεις του 2023 και όλα αυτά χωρίς διαφοροποίηση των δεδομένων και χωρίς την χρήση εξωγενών μεταβλητών.(όσο αφορά τα διαγράμματα αυτοσυσχετισης ,μερικής αυτοσυσχετισης και την γενικότερη πορεία των πωλήσεων σε διαγράμματα αυτή θα την αναλύσουμε στην συνέχεια όταν θα έχουμε βρει τις εξαμηνίες πωλήσεις και το κόστος πωληθέντων). Τέλος το τελευταίο μοντέλο που χρησιμοποιήθηκε ήταν το Arima(3,1,3) με μέσο τετραγωνικό σφάλμα ίσο με 3.972.739.039,1 πολύ μεγαλύτερο από το αντίστοιχο του Arma(3,3) δηλαδή του μοντέλου χωρίς την διαφοροποίηση (d=1) και οι εξαμηνιαίες πωλήσεις οπού πρόβλεψε ήταν για το πρώτο εξάμηνο ίσες με 2023-06-30 =194.258,2 και για το δεύτερο εξάμηνο ίσες με 2023-12-31 = 227.006,5 και η διαφορά που προκύπτει από τις πραγματικές πωλήσεις είναι ίση με 47.513,3 μεγαλύτερη διαφορά και από αυτή του μοντέλου Arima(1,1,0) οπότε βλέπουμε ότι τα δεδομένα παρότι δεν είναι στασιμά ίσως η μετατροπή τους σε στασιμά μέσο της διαφοροποίησης τους ίσως δώσει χειρότερες

προβλέψεις (η παραπάνω αναφορά δοκιμαστικέ παίρνοντας την πρώτη διαφορά των πωλήσεων δηλαδή τελικό μείον αρχικό (δηλαδή τ2-τ1) και τα αποτελέσματα ήταν χειροτέρα λόγο της μεγάλης διακύμανσης που υπάρχει στην αρχή της χρονοσειρας) οπότε και η ανάλυση θα γίνει με τα μη διαφοροποιημένα δεδομένα καθώς για την εφαρμογή τεχνικών μηχανικής μάθησης όπως είδαμε και στο κεφάλαιο 7 όπως τα τυχαία δάση δεν χρειάζεται υποχρεωτικά τα δεδομένα να είναι στασιμά. Έχοντας αναλύσει τα παραπάνω θα πάρουμε ως πραγματικές τιμές πωλήσεων για το πρώτο εξάμηνο τις 213.170 και για το δεύτερο εξάμηνο θα κάνουμε την αφαίρεση αυτών των εξαμηνιαίων πωλήσεων από τις πραγματικές ετήσιες πωλήσεις που εμφανίζονται στον ετήσιο ισολογισμό της εταιρίας για το 2023 δηλαδή 468.778-213.170 και με αυτό τον τρόπο οι πωλήσεις του δευτέρου εξάμηνου είναι ίσες με 255.608 ένα λογικό νούμερο πωλήσεων με βάση τις προηγούμενες εξαμηνιαίες πωλήσεις της εταιρίας. Όσο αναφορά το κόστος πωληθέντων και αυτή η χρονοσειρα δεν είναι στάσιμη και είναι παροιμία (δηλαδή ακολουθεί την πορεία) με τις πωλήσεις καθώς ανεξάρτητος ζήτησης όπως είδαμε και στην προηγουμένη ενότητα τα προϊόντα εισάγονται από το εξωτερικό και πωλούνται σε υψηλότερη τιμή και δημιουργούν το αντίστοιχο μικτό κέρδος και αρά η διαφορά(απόσταση) που προκύπτει μεταξύ αυτών των δυο είναι το μικτό κέρδος (το περιθώριο κέρδους) προφανώς δεν μπορούμε να ξέρουμε το ακριβές περιθώριο κέρδους για την κάθε κατηγορία προϊόντων πχ μπορεί να υπάρχει μεγαλύτερο περιθώριο κέρδους στις πωλήσεις λάπτοπ πάρα στις τηλεοράσεις και το ίδιο ισχύει και σε προϊόντα και μάρκες εντός της ιδίας κατηγορίας προϊόντων πχ τα λαπτοπ turbox να δημιουργούν μεγαλύτερο περιθώριο κέρδους σε σχέση με τα λαπτοπ της dell και προφανώς η τιμή αγοράς αυτών των προϊόντων κατά πασά πιθανότητα εξαρτάται και από το μέγεθος των παραγγελιών αλλά και αν οι εταιρίες προμηθεύον πχ τα λαπτοπ στο πλαίσιο χωρίς κάποιο κόστος(χωρίς να χρειάζεται να τα αγοράσει) ώστε να τα προβάλουν και αν αυτά αγοραστούν από τον καταναλωτή να πληρώνει το πλαίσιο την κατασκευάστρια εταιρία. Όσον αφορά τον υπολογισμό του κόστους πωληθέντων αρχικά θα πρέπει να αναφέρουμε ότι ο συντελεστής συσχέτισης μεταξύ πωλήσεων και κόστους πωληθέντων είναι ίσος σχεδόν με 1 δηλαδή υπάρχει τελειά θετική συσχέτιση μεταξύ των δυο μεταβλητών και όσο αναφορά τον συντελεστή β1 στην γραμμική παλινδρόμηση οπού εξαρτημένη μεταβλητή είναι οι πωλήσεις και ερμηνευτική μεταβλητή το μικτό κόστος είναι ίσο με 1,22 δηλαδή αν το πλαίσιο πουλήσει ένα λαπτοπ αξίας 1000 ευρώ το κέρδος του είναι 220 ευρώ δηλαδή η τιμή αγοράς για το πλαίσιο από τον κατασκευαστή είναι 820 ευρώ. Τα παραπάνω φαίνονται στο πίνακα summary model της γραμμικής παλινδρόμησης.

```
[7]  print(df['costs'].corr(df['sales']))

     0.9956720375775931
```

```
import statsmodels.api as sm
X = df['costs']
y = df['sales']
X = sm.add_constant(X)
model = sm.OLS(y, X).fit()
print(model.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  sales   R-squared:                       0.991
Model:                            OLS   Adj. R-squared:                  0.991
Method:                 Least Squares   F-statistic:                     2525.
Date:                Mon, 20 Jan 2025   Prob (F-statistic):           3.37e-24
Time:                        16:41:44   Log-Likelihood:                -240.51
No. Observations:                  24   AIC:                             485.0
Df Residuals:                      22   BIC:                             487.4
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const       4799.5720   3446.725      1.393      0.178   -2348.498    1.19e+04
costs          1.2216      0.024     50.251      0.000       1.171       1.272
==============================================================================
Omnibus:                        6.309   Durbin-Watson:                   1.770
Prob(Omnibus):                  0.043   Jarque-Bera (JB):                4.341
Skew:                           0.984   Prob(JB):                        0.114
Kurtosis:                       3.686   Cond. No.                     4.21e+05
==============================================================================
```

Εικονα 192: γραμμική παλινδρόμηση μεταξύ πωλήσεων και κόστους πωληθέντων

Παρόλα αυτά όσον αφορά την πρόβλεψη για την εύρεση του εξαμηνιαίου κόστους πωληθέντων του έτους 2023 χρησιμοποιήθηκε απευθείας το μοντέλο Arma(3,3) καθώς οι χρονοσειρες είναι σχεδόν ταυτόσημες και εφόσον αυτό το μοντέλο είχε το μικρότερο μέσο τετραγωνικό σφάλμα και ταυτόχρονα οι προβλέψεις των πωλήσεων για το 2023 ήταν σχεδόν ίσες με τις πραγματικές χρησιμοποιήθηκε απευθείας αυτό με μέσο τετραγωνικό σφάλμα ίσο με 604.294.777,2 και οι προβλέψεις για το κόστος πωληθέντων των επομένων εξάμηνων είναι για το πρώτο εξάμηνο ίσο με 2023-06-30 =170.727,9 και για το δεύτερο εξάμηνο ίσο με 2023-12-31 =193.918.2. Το πραγματικό κόστος πωληθέντων για το έτος 2023 όπως είδαμε και στην προηγουμένη ενότητα είναι ίσο με 382.270 οπότε η διαφορά που προκύπτει είναι ότι οι προβλέψεις έπεσαν έξω κατά 17.624 μονάδες μικρότερο κόστος

πωληθέντων. Οπότε οι τιμές που θα χρησιμοποιήσουμε για τα εξαμηνιαία κόστη πωληθέντων για το 2023 είναι για το πρώτο εξάμηνο 170.728 από το Arma(3,3) και για το δεύτερο γίνεται όπως και για τις πωλήσεις η αφαίρεση αυτών από τα πραγματικά ετήσια κόστη πωληθέντων του 2023 δηλαδή 382270-170228=211.542.

Οπότε τώρα που δεν μας λείπουν τιμές στην χρονοσειρα(και οι τιμές με τις οποίες τις συμπληρώσαμε φαίνονται να ταιριάζουν με αυτές που όντως πραγματοποιήθηκαν) μπορούμε να συνεχίσουμε στην πρόβλεψη των πωλήσεων και του κόστους πωληθεντων του 2024.

```
model_arma33c = ARIMA(df['costs'], order=(3, 0, 3))
model_fit_arma33c = model_arma33c.fit()
forecast_arma33c = model_fit_arma33c.forecast(steps=2)
print("Forecasted values using ARIMA(3,0,3):")
print(forecast_arma33c)
forecast_all_arma33c = model_fit_arma33c.predict(start=0, end=len(df['costs']) + 1)
mse_arma33c = mean_squared_error(df['costs'], forecast_all_arma33c.dropna().iloc[:22])
print("MSE for ARMA(3,0,3):", mse_arma33c)
```

```
Forecasted values using ARIMA(3,0,3):
2023-06-30    170727.900357
2023-12-31    193918.203157
Freq: 2QE-DEC, Name: predicted_mean, dtype: float64
MSE for ARMA(3,0,3): 604294777.2620486
```

Ξεκιναμε και εισαγουμε τα ολοκληρωμενα δεδομενα της χρονοσειρας με την χρηση της βιβλιοθηκης pandas στην python. Το dataset αποτελειται από τρεις στειλες που στην συνεχεια γινονται δυο καθως μετατρεπουμε την στειλη με τις ημερομινιες ως δεικτη της χρονοσειρας και αυτές είναι οι πωλησεις(sales) και το κοστος πωληθεντων(costs) σε εξαμινιαια περιοδο για αυτό και η συχνοτητα που επιλεχθηκε (frequency=6ms) είναι ιση με 6 μηνες αυτό θα βοηθησει στην συνεχεια την δημιουργεια των προβλεψεων εκτος χρονοσειρας. Στην συνεχεια με την χρηση της εντολης plot στην pandas αποικονιζονται και οι δυο σειρες ταυτοχρονα (αυτή η μεθοδολογια απεικονισης εχει χρισιμοποιηθει σχεδον σε ολλα τα διαγραμτα για απεικονιση των πραγματικων δεδομενων σε σχεση με τις προβλεψεις οσον αγορα τα δεδομενα εκπαιδευσης και ελενχου αλλα και για τις προβλεψεις εκτος χρονοσειρας). Όπως παρατηρουμε από το ετος 2012 εως και το 2016 υπαρχει μεγαλη διακυμαση των πωλησεων μεταξυ πρωτου και δευτερου εξαμηνου, καθως όπως εχουμε ειδη αναφερει η ζητηση για προιοντα τεχνολογιας και ειδων γραφειου αυξανεται τον σεπτεβρη (αρχη σχολικης / φοιτιτικης χρονιας) και τον νοεμβριο,δεκεμβριο(λογο εκπτωσεων και χριστουγεννων). Δηλαδη υπαρχει ξεκαθαρη ενδειξη για εποχικοτητα.

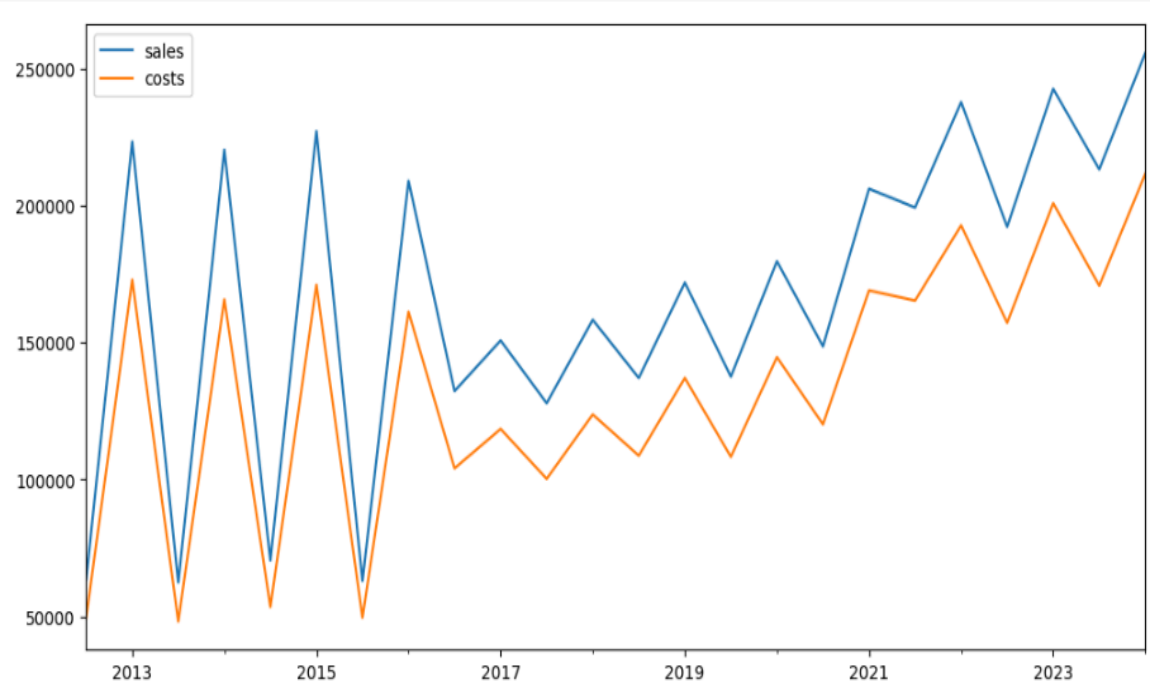
Στα επομενα ετη βλεπουμε ότι η μεγαλη μεταβλητοτητα μεταξυ των δυο εξαμηνων περιοριζεται παροτι και σε αυτά οι πωλησεις του δευτερου εξαμηνου υπερισχυουν του πρωτου ,επισης βλεπουμε μια ανοδικη πορεια συνολικα των πωλησεων (trend) κατι το οποιο δεν ισχυει για τα πρωτα ετη. Δηλαδη η εταιρεια βρηκε τροπους ώστε να αυξησει και τις πωλησεις του πρωτου εξαμηνου βεβαια σε αυτό μπορει να βοηθησε η συνολικη ανοδος του αεπ της ελλινηκης οικονομιας και ετσι περισσοτεροι καταναλωτες να αυξησαν τις δαπανες τους εκοινο το εξαμηνο το οποιο περιλαμβανει εκθεσεις, ανακοινωσεις και κατά επεκταση κυκλοφοριες νεων προιοντων τεχνολογιας.

```
[ ]  df['time'] = pd.to_datetime(df['time'])

     df.set_index('time', inplace=True)
```

```
[ ]  df.index = pd.date_range(start=df.index[0], periods=len(df), freq='6MS')
```

```
     df
```

	sales	costs
2012-07-01	63600	49577
2013-01-01	223276	172875
2013-07-01	62456	48220
2014-01-01	220283	165730
2014-07-01	70494	53497
2015-01-01	227054	170982
2015-07-01	63005	49612
2016-01-01	208980	161180
2016-07-01	132281	104090



Εικονα 193: plot των πωλήσεων και του κόστους πωληθέντων

Ξεκινάμε την ανάλυση με τον απλό κινητό μέσο ορό υπολογίζοντας τον με δυο έως πέντε παρατηρήσεις. Όπως βλέπουμε κανένα από τα παραπάνω μοντέλα δεν μπορεί να ακολουθήσει την έντονη μεταβλητότητα των τιμών στα πρώτα έτη της σειράς ενώ από το 2017 που η διακύμανση εξομαλύνεται οι προβλέψεις είναι πιο κοντά στις πραγματικές τιμές. Όπως έχουμε ήδη αναφέρει όσο αυξάνουμε τις παρατηρήσεις υπολογισμού του μέσου ορού τόσο πιο <<smooth και lean>> γίνονται οι προβλέψεις εξομαλύνοντας την διακύμανση.

```
df['sma2_sales']=df['sales'].rolling(2).mean()
df['sma3_sales']=df['sales'].rolling(3).mean()
df['sma4_sales']=df['sales'].rolling(4).mean()
df['sma5_sales']=df['sales'].rolling(5).mean()
df[['sales','sma2_sales','sma3_sales','sma4_sales','sma5_sales']].plot(figsize=(12,6))
plt.show()
```


```
from sklearn.metrics import mean_squared_error
mse2 = mean_squared_error(df['sales'][1:], df['sma2_sales'][1:])
mse3 = mean_squared_error(df['sales'][2:], df['sma3_sales'][2:])
mse4 = mean_squared_error(df['sales'][3:], df['sma4_sales'][3:])
mse5 = mean_squared_error(df['sales'][4:], df['sma5_sales'][4:])
print(f'MSE for SMA2: {mse2}')
print(f'MSE for SMA3: {mse3}')
print(f'MSE for SMA4: {mse4}')
print(f'MSE for SMA5: {mse5}')
min_mse = min(mse2, mse3, mse4, mse5)
print(f'Minimum MSE: {min_mse}')
```

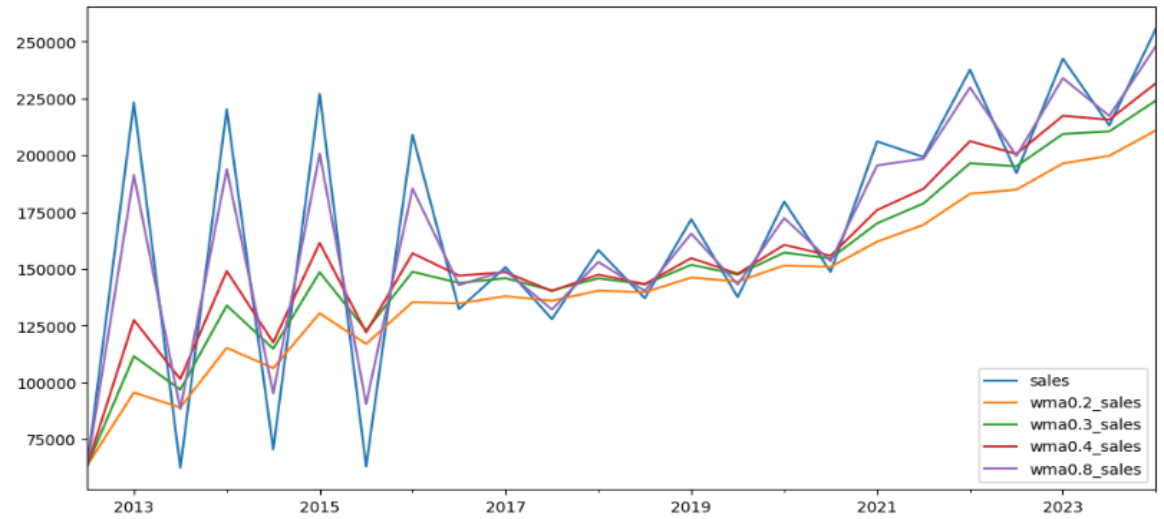
```
MSE for SMA2: 2138872088.9565217
MSE for SMA3: 888558212.2424244
MSE for SMA4: 1796190698.8363094
MSE for SMA5: 1069812926.7519999
Minimum MSE: 888558212.2424244
```

Εικονα 194:Εφαρφογη sma

Στην συνέχεια υπολογίζουμε το μέσο τετραγωνικό σφάλμα κάθε μοντέλου και βλέπουμε ότι το μικρότερο σφάλμα προκύπτει από το μοντέλο που υπολόγιζε τον μέσο ορό τριών παρατηρήσεων. Προφανώς ο αναλυτής θα πρέπει να προσέξει ότι το μοντέλο δεν μπορεί να δημιουργήσει προβλέψεις για τον αριθμό των παρατηρήσεων που έχουν χρησιμοποιηθεί για τον υπολογισμό του μέσου ορού για παράδειγμα εάν έχουν χρησιμοποιηθεί 5 τιμές τότε δεν μπορεί να υπολογιστεί πρόβλεψη για τις πρώτες 4 παρατηρήσεις.

Συνεχίζουμε με την μέθοδο του σταθμισμένου κινητού μέσου ορού ο οποίος έχει υπολογιστεί για α=0,3 ,α=0,4 και α=0,8 αντίστοιχα όπως έχουμε αναφέρει όσο μεγαλύτερο είναι το α τόσο περισσότερο το μοντέλο εστιάζει το υπολογισμό του μέσου ορού των δυο τελευταίων τιμών στην πιο πρόσφατη τιμή. Για αυτό το λόγο και όσο αυξάνουμε το α το μέσο τετραγωνικό σφάλμα μειώνεται καθώς το μοντέλα απλά αντιγράφει την πραγματική τιμή της χρονοσειρας ενώ χρησιμοποιώντας ένα μικρό α το μοντέλο ακολουθεί την γενικότερη τάση της χρονοσειρας.

```
df['wma0.3_sales'] = df['sales'].ewm(alpha=0.3, adjust=False).mean()
df['wma0.4_sales'] = df['sales'].ewm(alpha=0.4, adjust=False).mean()
df['wma0.8_sales'] = df['sales'].ewm(alpha=0.8, adjust=False).mean()
```




```
msewma02=mean_squared_error(df['sales'], df['wma0.2_sales'])
msewma03=mean_squared_error(df['sales'], df['wma0.3_sales'])
msewma04=mean_squared_error(df['sales'], df['wma0.4_sales'])
msewma08=mean_squared_error(df['sales'], df['wma0.8_sales'])
print(f'MSE for WMA0.2: {msewma02}')
print(f'MSE for WMA0.3: {msewma03}')
print(f'MSE for WMA0.4: {msewma04}')
print(f'MSE for WMA0.8: {msewma08}')
min_wmamse = min(msewma02, msewma03, msewma04, msewma08)
print(f'Minimum MSE: {min_wmamse}')
```

```
MSE for WMA0.2: 2469200747.719347
MSE for WMA0.3: 1815724597.8976681
MSE for WMA0.4: 1385306919.6020443
MSE for WMA0.8: 237605338.49100062
Minimum MSE: 237605338.49100062
```

Εικονα 195: εφαρμογη wma

Εν συνέχεια εφαρμόζεται το μοντέλο holt winters με τάση και εποχικότητα προσθετική για την πρόβλεψη των τιμών όλης της χρονοσειρας. Όπως βλέπουμε το μοντέλο κάνει καλές

προβλέψεις για τις πρώτες τιμές τις χρονοσειρας που έχουν μεγάλη μεταβλητότητα αλλά στην συνέχεια που αυτή η μεγάλη διακύμανση εξομαλύνεται δεν μπορεί να κατανοήσει την ανοδική τάση παρόλο που έχει αναγνωρίσει την εποχικότητα δηλαδή ότι το πρώτο εξάμηνο έχει μικρότερες πωλήσεις σε σχέση με το δεύτερο. Λόγο των παραπάνω εφαρμόζεται το ίδιο μοντέλο αλλά με πολλαπλασιαστική τάση και εποχικότητα και οι περίοδοι εποχικότητας ίσοι με 2 (όσα και τα εξάμηνα). Και όντως τα αποτελέσματα όσον αφορά το μέσο τετραγωνικό σφάλμα του δευτέρου μοντέλου είναι καλυτέρα από αυτά του πρώτου.

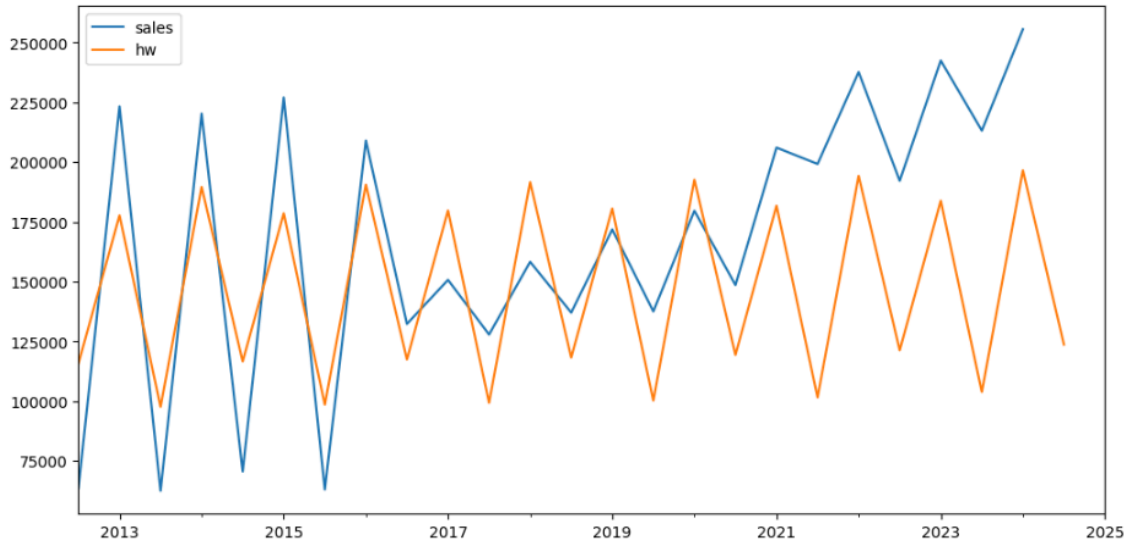
```
from statsmodels.tsa.holtwinters import ExponentialSmoothing
```

```
hw=ExponentialSmoothing(df['sales'],trend='add',seasonal='add',seasonal_periods=4)
reshw=hw.fit()
```

```
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/holtwinters/model.py:918: C
  warnings.warn(
```

```
df['hw']=reshw.predict(start=df.index[0],end=df.index[-1])
```

```
df[['sales','hw']].plot(figsize=(12,6))
plt.show()
```




```
mseholt=mean_squared_error(df['sales'].dropna(), df['hw'][:24].dropna())
print(mseholt)
```

```
2284250292.281329
```

Εικονα 196:Εφαρμογη holt winters

```
[6]  hw=ExponentialSmoothing(df['sales'],trend='mul',seasonal='mul',seasonal_periods=2)
     reshw=hw.fit()
```

```
/usr/local/lib/python3.11/dist-packages/statsmodels/tsa/holtwinters/model.py:918: C
  warnings.warn(
```


```
from sklearn.metrics import mean_squared_error
mseholt=mean_squared_error(df['sales'].dropna(), df['hw'][:24].dropna())
print(mseholt)
```

```
635491227.2475538
```

Εικονα 196 μέρος δεύτερο: holt winters mul,mul

Στην συνέχεια ακολουθεί η πρόβλεψη των πωλήσεων μέσο μοντέλων μηχανικής μάθησης όπως η γραμμική παλινδρόμηση το support vector machine και το random forest. Η μεθοδολογία είναι ακριβός ιδιά με αυτή που εφαρμόστηκε στο κεφάλαιο 7 για την πρόβλεψη της τιμής του Samsung s24 ,μόνο που σε αυτή την περίπτωση επειδή τα δεδομένα είναι σε εξαμηνιαία βάση από το 2012 έως το 2024 δηλαδή έχουμε μόνο 24 παρατηρήσεις αντί για έλεγχο της απόδοσης του μοντέλου στις 30 τελευταίες ημέρες που είχε εφαρμοστεί για τις τιμές του κινητού τηλεφώνου εδώ ο έλεγχος γίνεται στα τελευταία δυο εξάμηνα της χρονοσειρας(τις εξαμηνιαίες πωλήσεις του 2023) και όλα τα υπόλοιπα δεδομένα χρησιμοποιούνται για την εκπαίδευση του μοντέλου (ntest=2).

```
from sklearn.linear_model import LinearRegression
from sklearn.svm import SVR
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error
```

```
[ ] ntest=2
    train=dflr.iloc[:-ntest]
    test=dflr.iloc[-ntest:]
```

```
[ ] series=dflr.to_numpy()
    T=2
    X=[]
    Y=[]
    for t in range(len(series)-T):
        x=series[t:t+T]
        X.append(x)
        y=series[t+T]
        Y.append(y)
    X=np.array(X).reshape(-1,T)
    Y=np.array(Y)
    N=len(X)
    print(X.shape)
    print(Y.shape)
```

```
(22, 2)
(22, 1)
```

```
[ ] X_train=X[:-ntest]
    Y_train=Y[:-ntest]
    X_test=X[-ntest:]
    Y_test=Y[-ntest:]
```

```
[ ] clr=LinearRegression()
    clr.fit(X_train,Y_train)
    train_idx=dflr.index<=train.index[-1]
    test_idx=~train_idx
    train_idx[:T]=False
    dflr=pd.DataFrame(dflr)
    dflr.loc[train_idx,'lr_train']=clr.predict(X_train)
    dflr.loc[test_idx,'lr_test']=clr.predict(X_test)
    dflr[['sales','lr_train','lr_test']].plot(figsize=(12,6))
    plt.show()
```

Στην συνέχεια υπολογίζεται το μέσο τετραγωνικό σφάλμα της γραμμικής παλινδρόμησης για τα δεδομένα της εκπαίδευσης και για τα δεδομένα του test set το οποίο είναι ίσο με 80.086.740 και γίνονται οι προβλέψεις των πωλήσεων για το 2024. Επίσης θα πρέπει να ενημερώσουμε ότι δεν εφαρμόστηκε κάποια μέθοδος τροποποίησης των δεδομένων πριν την εφαρμογή του μοντέλου.

```
mselr=mean_squared_error(dflr['sales'][2:-2],dflr['lr_train'][2:-2])
print('mse for lr=',mselr)
```

```
mse for lr= 544933198.9419698
```

```
from sklearn.linear_model import LinearRegression


last_T_values = X[-1].reshape(1, -1)

next_3_values = []
for _ in range(3):
    next_value = clr.predict(last_T_values)
    next_3_values.append(next_value[0])
    last_T_values = np.roll(last_T_values, -1)
    last_T_values[0,-1] = next_value

next_3_values
```

```
new_index_values = pd.to_datetime(['2024-07-01', '2025-01-01'])
new_rows = pd.DataFrame(index=new_index_values, columns=dflr.columns)
dflr = pd.concat([dflr, new_rows])
dflr
```

```
dflr['forecastlr'].iloc[-3:] = [x[0] for x in next_3_values]
```

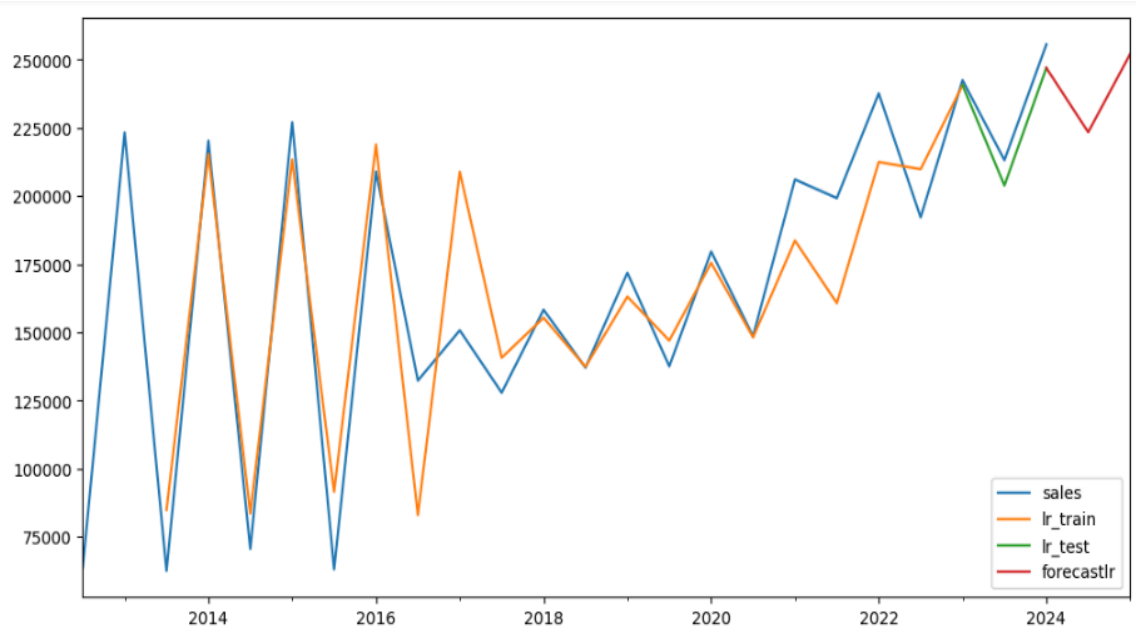
2023-01-01	242499	240642.497106	NaN	NaN
2023-07-01	213170	NaN	203878.032789	NaN
2024-01-01	255608	NaN	247014.472071	247014.472071
2024-07-01	NaN	NaN	NaN	223438.942448
2025-01-01	NaN	NaN	NaN	252092.848170

Εικονα 197: κώδικας γραμμικής παλινδρόμησης

Ο πίνακας που εμφανίζεται παραπάνω έχει στην πρώτη στήλη τον δείκτη δηλαδή την ημερομηνία στην δεύτερη στήλη τις πραγματικές πωλήσεις στην τρίτη στήλη τις προβλέψεις όσο αφορά τα δεδομένα εκπαίδευσης στην τέταρτη στήλη τις προβλέψεις του μοντέλου όσον αφορά το test set και η τελευταία στήλη τις προβλέψεις εκτός dataset δηλαδή τις προβλέψεις των πωλήσεων για το 2024.

Όσο αναφορά τις προβλέψεις του μοντέλου για το 2024 αυτές δεν απέχουν πολύ από την πραγματικότητα, συγκεκριμένα το μοντέλο προβλέπει ότι για το πρώτο εξάμηνο του 2024

οι πωλήσεις θα είναι μεγαλύτερες σε σχέση με αυτές του πρώτου εξάμηνου του 2023 αλλά οι πωλήσεις του δευτέρου εξάμηνου μικρότερες σε σχέση με αυτές του 2023.



Εικονα 198: Εφαρμογή γραμμικής παλινδρόμησης

Το επόμενο μοντέλο είναι το support vector regression οπού πριν το εφαρμόσουμε κάνουμε scale τα δεδομένα έτσι ώστε όλες οι παρατηρήσεις να έχουν τιμές μεταξύ μηδέν και 1 , και στην συνέχεια αφότου έχουμε κάνει τις προβλέψεις για το train και test set και το forecast για το 2024 επαναφέρουμε τα δεδομένα στην αρχική τους κατάσταση. Στην συνέχεια υπολογίζουμε το μέσο σταθμικό σφάλμα και δημιουργούμε το διάγραμμα όλων των παραπάνω μεταβλητών. Όπως μπορούμε να παρατηρήσουμε το μοντέλο δημιουργεί καλές προβλέψεις για τα δεδομένα εκπαίδευσης, αναγνωρίζει την εποχικότητα και την τάση αλλά οι προβλέψεις για τις πωλήσεις όσο αναφορά τα δεδομένα ελέγχου(test set) είναι αρκετά πιο χαμηλές από τις πραγματικές, με αποτέλεσμα η γραμμική παλινδρόμηση να έχει μικρότερο τετραγωνικό σφάλμα. Όσον αφορά τις προβλέψεις για το 2024 το μοντέλο εκτιμά ότι θα είναι μικρότερες σε σχέση με το 2023 και για τα δυο εξάμηνα.

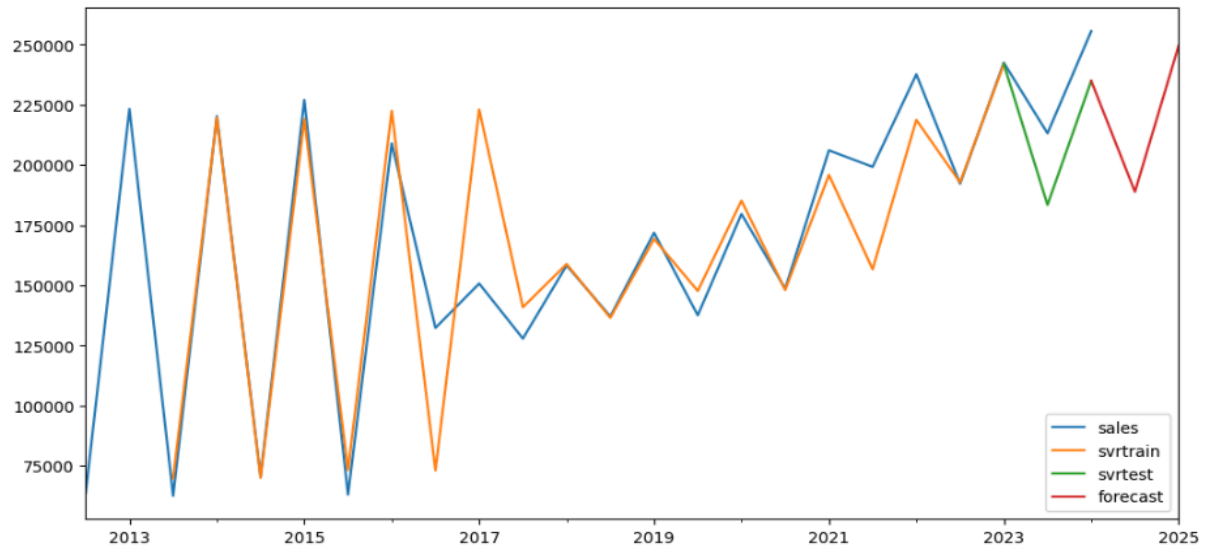
```
scaler_X = StandardScaler()
scaler_Y = StandardScaler()

X_train_scaled = scaler_X.fit_transform(X_train)
X_test_scaled = scaler_X.transform(X_test)
Y_train_scaled = scaler_Y.fit_transform(Y_train.reshape(-1, 1)).ravel()

sv = SVR(kernel='rbf', C=10, gamma=0.1, epsilon=0.01)
sv.fit(X_train_scaled, Y_train_scaled)

Y_train_pred_scaled = sv.predict(X_train_scaled)
Y_test_pred_scaled = sv.predict(X_test_scaled)
Y_train_pred = scaler_Y.inverse_transform(Y_train_pred_scaled.reshape(-1, 1)).ravel()
Y_test_pred = scaler_Y.inverse_transform(Y_test_pred_scaled.reshape(-1, 1)).ravel()
train_idx = dfsvr.index <= train.index[-1]
test_idx = ~train_idx
train_idx[:T] = False
dfsvr = pd.DataFrame(dfsvr)
dfsvr['svrtrain'] = None
dfsvr['svrtest'] = None
dfsvr.loc[train_idx, 'svrtrain'] = Y_train_pred
dfsvr.loc[test_idx, 'svrtest'] = Y_test_pred
mse_trainsvr = mean_squared_error(Y_train, Y_train_pred)
mse_testsvr = mean_squared_error(Y_test, Y_test_pred)
print(f"Train MSE: {mse_trainsvr}")
print(f"Test MSE: {mse_testsvr}")
```

```
Train MSE: 586603744.7349151
Test MSE: 655983085.377289
```

```
num_forecasts = 2
forecast_values = []

last_T_values = series[-T:].reshape(1, -1)

for _ in range(num_forecasts):
    last_T_values_scaled = scaler_X.transform(last_T_values)
    next_value_scaled = sv.predict(last_T_values_scaled)
    next_value = scaler_Y.inverse_transform(next_value_scaled.reshape(-1, 1)).ravel()[0]
    forecast_values.append(next_value)
    last_T_values = np.roll(last_T_values, -1)
    last_T_values[0, -1] = next_value
forecast_indices = pd.date_range(start=dfsvr.index[-1], periods=num_forecasts + 1, freq="6MS")[1:]
forecast_df = pd.DataFrame({'forecast': forecast_values}, index=forecast_indices)
dfsvr = pd.concat([dfsvr, forecast_df], axis=0)
print(forecast_df)
```

```
                forecast
2024-07-01  188847.059795
2025-01-01  249586.983971
```

2023-01-01	242499.0	241950.334876	None	NaN
2023-07-01	213170.0	None	183424.099827	NaN
2024-01-01	255608.0	None	234940.45071	NaN
2024-07-01	NaN	NaN	NaN	188847.059795
2025-01-01	NaN	NaN	NaN	249586.983971



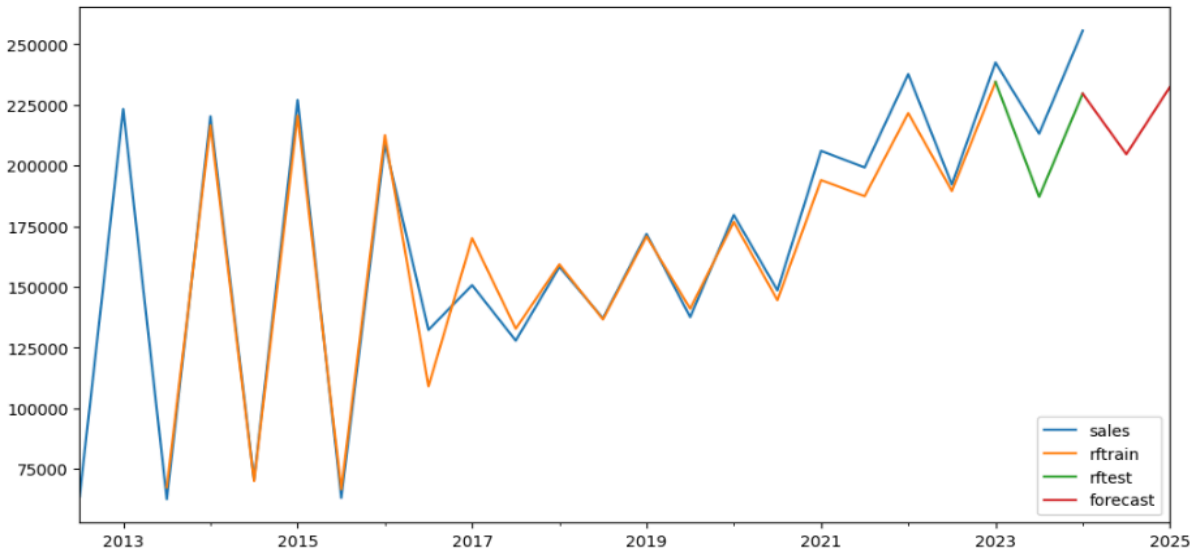
Εικονα 199: κώδικας και εφαρμογή svr

Το τελευταίο μοντέλο μηχανικής μάθησης που εφαρμόστηκε στις πωλήσεις είναι ο αλγόριθμος των τυχαίων δασών. Όπως και στον αλγόριθμο support vector machine έτσι και εδώ κάνουμε scale τα δεδομένα πριν την εφαρμογή του μοντέλου και τα επαναφέρουμε στην αρχική τους κατάσταση αφότου έχουμε κάνει τις προβλέψεις. ο αριθμός των δέντρων απόφασης που επιλέχθηκε είναι τα 100, παρόλα αυτά το μοντέλο δίνει τις χειρότερες προβλέψεις και στο train και στο test data και από τα δυο προηγούμενα μοντέλα. Επίσης προβλέπει ότι οι πωλήσεις του 2024 θα είναι μικρότερες από αυτές του 2023.

```
scaler_X = StandardScaler()
scaler_Y = StandardScaler()
X_train_scaled = scaler_X.fit_transform(X_train)
X_test_scaled = scaler_X.transform(X_test)
Y_train_scaled = scaler_Y.fit_transform(Y_train.reshape(-1, 1)).ravel()
rf = RandomForestRegressor(n_estimators=100, random_state=42)
rf.fit(X_train_scaled, Y_train_scaled)
Y_train_pred_scaled = rf.predict(X_train_scaled)
Y_test_pred_scaled = rf.predict(X_test_scaled)
Y_train_pred = scaler_Y.inverse_transform(Y_train_pred_scaled.reshape(-1, 1)).ravel()
Y_test_pred = scaler_Y.inverse_transform(Y_test_pred_scaled.reshape(-1, 1)).ravel()
dfrf = pd.DataFrame(dfrf)
dfrf['rftrain'] = None
dfrf['rftest'] = None
train_idx = dfrf.index <= train.index[-1]
test_idx = ~train_idx
train_idx[:T] = False
dfrf.loc[train_idx, 'rftrain'] = Y_train_pred
dfrf.loc[test_idx, 'rftest'] = Y_test_pred
mse_train_rf = mean_squared_error(Y_train, Y_train_pred)
mse_test_rf = mean_squared_error(Y_test, Y_test_pred)
print(f"Train MSE (Random Forest): {mse_train_rf}")
print(f"Test MSE (Random Forest): {mse_test_rf}")
```

```
Train MSE (Random Forest): 84914048.13129964
Test MSE (Random Forest): 677107274.6656995
```

2023-01-01	242499.0	234464.47	None	NaN
2023-07-01	213170.0	None	187100.35	NaN
2024-01-01	255608.0	None	229635.17	NaN
2024-07-01	NaN	NaN	NaN	204672.07
2025-01-01	NaN	NaN	NaN	232441.12



Εικονα 200: κώδικας και εφαρμογή random forest regressor

Έχοντας κάνει προβλέψεις με τα μοντέλα μηχανικής μάθησης συνεχίζουμε την ανάλυση με πιο παραδοσιακά μοντέλα προβλέψεων χρονολογικών σειρών όπως τα αυτοπλαινδρομα, τον κινητό μέσο ορό το Arma και το Arima.Αρχικά εισάγουμε τις βιβλιοθήκες που εκτελούν τα παραπάνω μοντέλα στην συνέχεια ελέγχουμε την στασιμότητα της χρονοσειρα με τις πωλήσεις και βλέπουμε ότι σε καμία περίπτωση δεν είναι στάσιμη με βάση το dickey fuller test παρόλα αυτά δεν διαφοροποιούμε τα δεδομένα και συνεχίζουμε την ανάλυση μας δημιουργώντας το διάγραμμα αυτοσυσχετισης στο οποίο παρατηρούμε ότι υπάρχει σημαντική αυτοσυσχετιση στα πρώτα δυο lags της χρονοσειρας αρά θα χρησιμοποιήσουμε ma(2). Όσον αφορά το την μερική αυτοσυσχετιση βλέπουμε στο διάγραμμα pacf ότι στο lag=2 υπάρχει έντονη συσχέτιση οπότε θα χρησιμοποιηθεί το μοντέλο ar(2) και κατά επέκταση το Arma(2,2) και Arima(2,1,2). Όσον αφορά τον διαχωρισμό των δεδομένων σε train και test set είναι ίδιο με τα μοντέλα μηχανικής μάθησης δηλαδή κρατάμε τα δυο τελευταία εξάμηνα για έλεγχο της απόδοσης των μοντέλων (ntest=2).

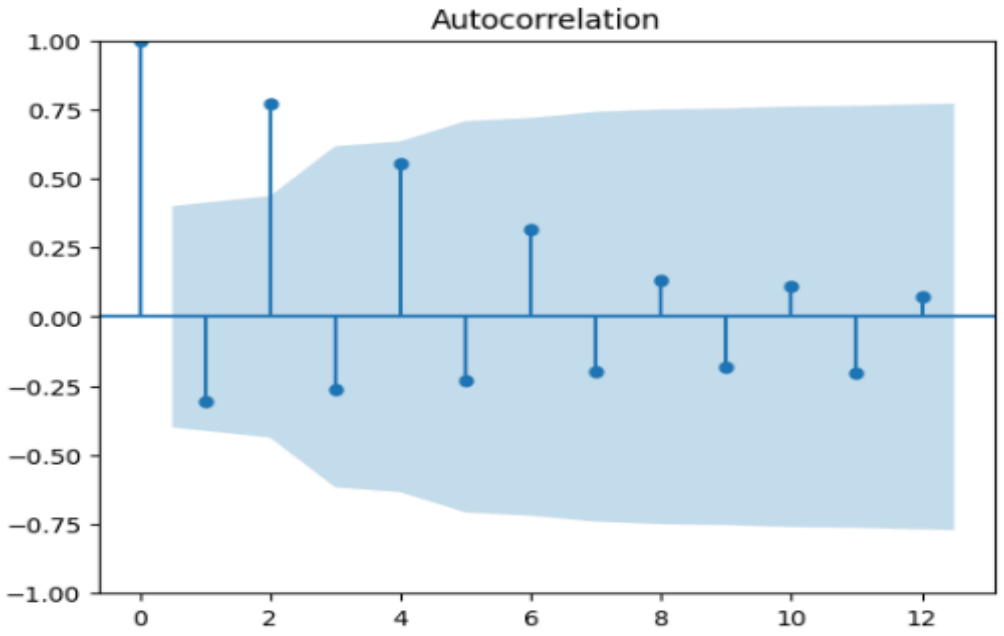
```
result=adfuller(dfar)
print('ADF Statistic:',result[0])
print('p-value:',result[1])
print('Critical Values:')
for key,value in result[4].items():
    print('\t',key,':',value)
```

```
ADF Statistic: 0.6043878863924554
p-value: 0.987723285140875
Critical Values:
        1% : -3.7883858816542486
        5% : -3.013097747543462
        10% : -2.6463967573696143
```

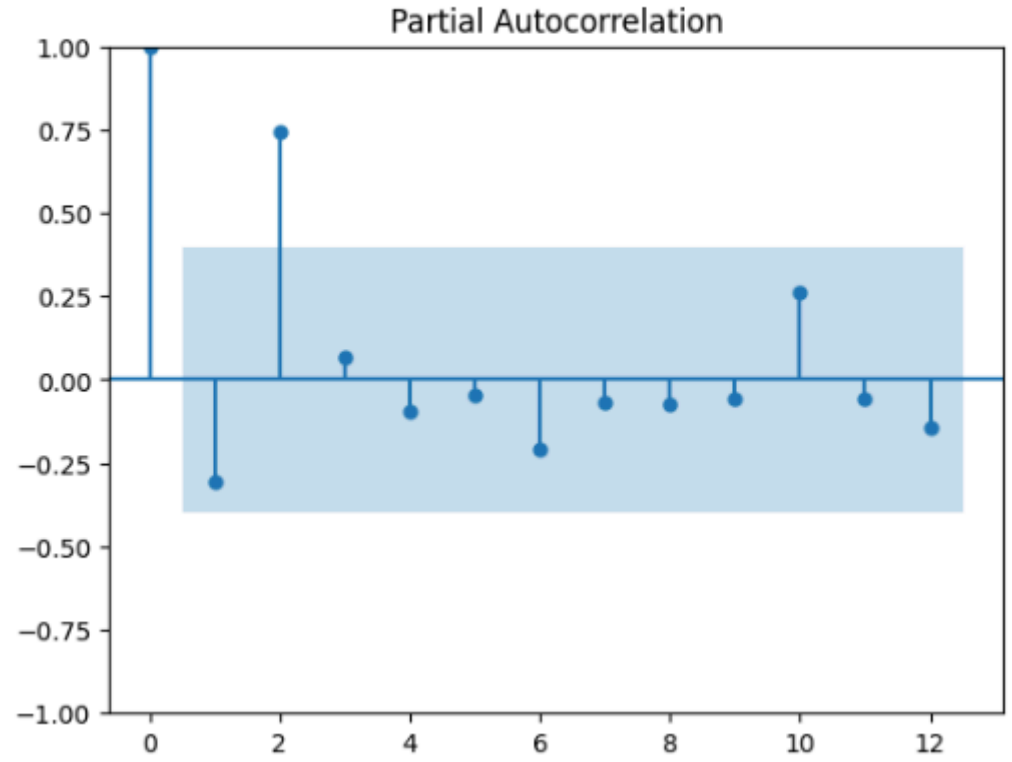
```
from statsmodels.tsa.ar_model import AutoReg
from statsmodels.tsa.arima.model import ARIMA
from statsmodels.graphics.tsaplots import plot_acf, plot_pacf
from statsmodels.tsa.stattools import adfuller
```

```
ntest=2
train=dfar.iloc[:-ntest]
test=dfar.iloc[-ntest:]
train_idx=dfar.index<=train.index[-1]
test_idx=dfar.index>train.index[-1]
```

```
plot_acf(dfar['sales'])
plt.show()
```




```
plot_pacf(dfar['sales'])
plt.show()
```



Εικονα 201: pacf και acf plot

Ξεκινάμε με το αυτοπαλινδρομο(ar) με p=2 βλέπουμε ότι από το model summary ότι η πρώτη lagged version της χρονοσειρας δεν είναι στατιστικά σημαντική με βάση το p-value παρόλα αυτά επειδή η δεύτερη lagged version της χρονοσειρας είναι στατιστικά σημαντική συνεχίζουμε με αυτό το μοντέλο. Στην συνέχεια κάνουμε predict της τιμές της χρονοσειρας όσον αφορά το train και test set, και ταυτόχρονα υπολογίζουμε τα αντίστοιχα μέσα τετραγωνικά σφάλματα αυτών των δυο, και ολοκληρώνουμε την διαδικασία υπολογίζοντας τις forecasted πωλήσεις για το 2024. Όπως μπορούμε να παρατηρήσουμε το μέσο τετραγωνικό σφάλμα είναι πολύ μεγαλύτερο και στις δυο περιπτώσεις(train και test set) σε σχέση με αυτό που προέκυψε από τα μοντέλα μηχανικής μάθησης. Σχετικά με τις προβλέψεις για το 2024 το μοντέλο εκτιμά ότι θα είναι λίγο μικρότερες και για τα δυο εξάμηνα σε σχέση με το 2023.

```
ar2=ARIMA(dfar['sales'],order=(2,0,0))
ar2_fit=ar2.fit()
print(ar2_fit.summary())
```

```
                               SARIMAX Results
==============================================================================
Dep. Variable:                   sales   No. Observations:                   24
Model:                 ARIMA(2, 0, 0)   Log Likelihood                -279.228
Date:                Sun, 19 Jan 2025   AIC                            566.456
Time:                        19:11:02   BIC                            571.168
Sample:                    07-01-2012   HQIC                           567.706
                         - 01-01-2024
Covariance Type:                  opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const       1.679e+05   4.28e+04      3.921      0.000     8.4e+04    2.52e+05
ar.L1         -0.0181      0.075     -0.241      0.810      -0.166       0.129
ar.L2          0.9308      0.078     11.961      0.000       0.778       1.083
sigma2      5.596e+08      5.095    1.1e+08      0.000     5.6e+08     5.6e+08
===================================================================================
Ljung-Box (L1) (Q):                   0.31   Jarque-Bera (JB):                 1.19
Prob(Q):                              0.58   Prob(JB):                         0.55
Heteroskedasticity (H):               2.77   Skew:                            -0.11
Prob(H) (two-sided):                  0.17   Kurtosis:                         4.07
===================================================================================
```

```
[ ] dfar.loc[train_idx,'ar2train']=ar2_fit.predict(start=train.index[0],end=train.index[-1])
    dfar.loc[test_idx,'ar2test']=ar2_fit.predict(start=test.index[0],end=test.index[-1])
```

```
mse_train_ar2=mean_squared_error(train['sales'],dfar.loc[train_idx,'ar2train'])
mse_test_ar2=mean_squared_error(test['sales'],dfar.loc[test_idx,'ar2test'])
print('mse train ar2=',mse_train_ar2)
print('mse test ar2=',mse_test_ar2)
```

```
mse train ar2= 1103337376.3868546
mse test ar2= 470854811.40116465
```

```
forecastar2=ar2_fit.forecast(steps=2)
forecastar2
```

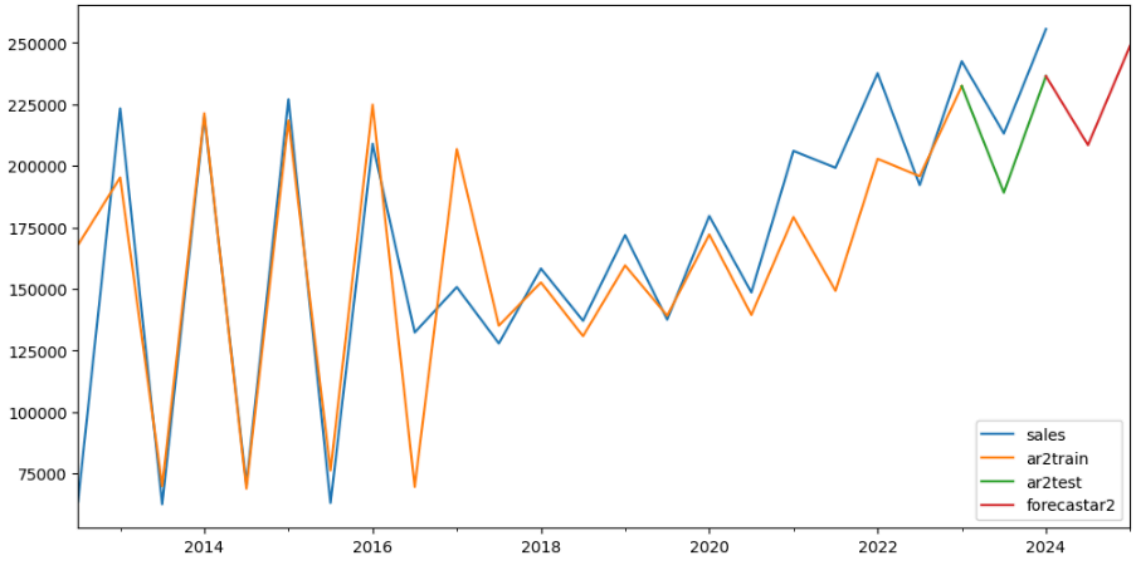
	predicted_mean
2024-07-01	208444.593694
2025-01-01	248800.796951

```
new_index_values = pd.to_datetime(['2024-07-01', '2025-01-01'])
new_rows = pd.DataFrame(index=new_index_values, columns=dfar.columns)
dfar = pd.concat([dfar, new_rows])
```

```
<ipython-input-67-85a560fad46f>:5: FutureWarning: The behavior of DataF
  dfar = pd.concat([dfar, new_rows])
```

```
dfar['forecastar2']=np.nan
dfar.loc['2024-07-01', 'forecastar2'] = forecastar2[0]
dfar.loc['2025-01-01', 'forecastar2'] = forecastar2[1]
dfar
```

2023-01-01	242499	232395.560951	NaN	NaN
2023-07-01	213170	NaN	189146.979425	NaN
2024-01-01	255608	NaN	236513.390675	NaN
2024-07-01	NaN	NaN	NaN	208444.593694
2025-01-01	NaN	NaN	NaN	248800.796951



Εικονα 202: κωδικας και εφαρμογη ar(2)

Συνεχίζουμε με το μοντέλο του κινητού μέσου ορού ma(2), ο κώδικάς είναι ακριβός ίδιος με αυτόν που εφαρμόσαμε για το ar(2) με μονή διαφορά στο order που επιλέγεται το οποίο είναι (0,0,2). Η πρώτη lagged version δεν είναι στατιστικά σημαντική με βάση το p-value δεν ισχύει το ίδιο για την δεύτερη. Το μέσο τετραγωνικό σφάλμα για το train και test set είναι μεγαλύτερο από τα αντίστοιχα του ar(2), και όσον αφορά τις προβλέψεις για το 2024 το μοντέλο εκτιμά ότι θα είναι αρκετά μειωμένες σε σχέση με το 2023.

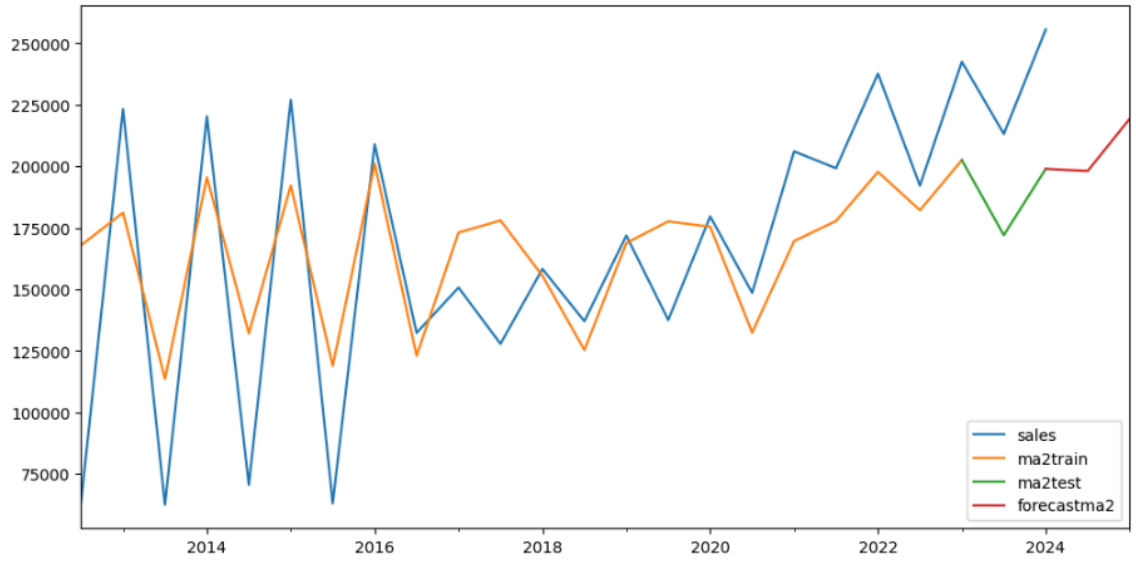
```
dfma=df.copy()
```

```
ma2=ARIMA(dfma['sales'],order=(0,0,2))
ma2_fit=ma2.fit()
print(ma2_fit.summary())
dfma.loc[train_idx,'ma2train']=ma2_fit.predict(start=train.index[0],end=train.index[-1])
dfma.loc[test_idx,'ma2test']=ma2_fit.predict(start=test.index[0],end=test.index[-1])
mse_train_ma2=mean_squared_error(train['sales'],dfma.loc[train_idx,'ma2train'])
mse_test_ma2=mean_squared_error(test['sales'],dfma.loc[test_idx,'ma2test'])
print('mse train ma2=',mse_train_ma2)
print('mse test ma2=',mse_test_ma2)
forecastma2=ma2_fit.forecast(steps=2)
forecastma2
new_index_values = pd.to_datetime(['2024-07-01', '2025-01-01'])
new_rows = pd.DataFrame(index=new_index_values, columns=dfma.columns)
dfma = pd.concat([dfma, new_rows])
dfma['forecastma2']=np.nan
dfma.loc['2024-07-01', 'forecastma2'] = forecastma2[0]
dfma.loc['2025-01-01', 'forecastma2'] = forecastma2[1]
dfma
```

```
                               SARIMAX Results
==============================================================================
Dep. Variable:                   sales   No. Observations:                   24
Model:                 ARIMA(0, 0, 2)   Log Likelihood                -287.555
Date:                Sun, 19 Jan 2025   AIC                            583.109
Time:                        19:26:11   BIC                            587.821
Sample:                    07-01-2012   HQIC                           584.359
                         - 01-01-2024
Covariance Type:                  opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const       1.679e+05   1.55e+04     10.819      0.000    1.37e+05    1.98e+05
ma.L1         -0.1241      0.253     -0.492      0.623      -0.619       0.371
ma.L2          0.9325      0.420      2.218      0.027       0.108       1.757
sigma2      1.556e+09      0.110   1.42e+10      0.000    1.56e+09    1.56e+09
===================================================================================
Ljung-Box (L1) (Q):                   0.14   Jarque-Bera (JB):                 2.09
Prob(Q):                              0.70   Prob(JB):                         0.35
Heteroskedasticity (H):               0.63   Skew:                            -0.63
Prob(H) (two-sided):                  0.53   Kurtosis:                         2.29
===================================================================================
```

```
mse train ma2= 1558542709.6599302
mse test ma2= 2450861776.7869034
```

2023-07-01	213170	NaN	172008.743354	NaN
2024-01-01	255608	NaN	198973.430125	NaN
2024-07-01	NaN	NaN	NaN	198107.465114
2025-01-01	NaN	NaN	NaN	219281.366161



Εικονα 202: κώδικας και εφαρμογή ma(2)

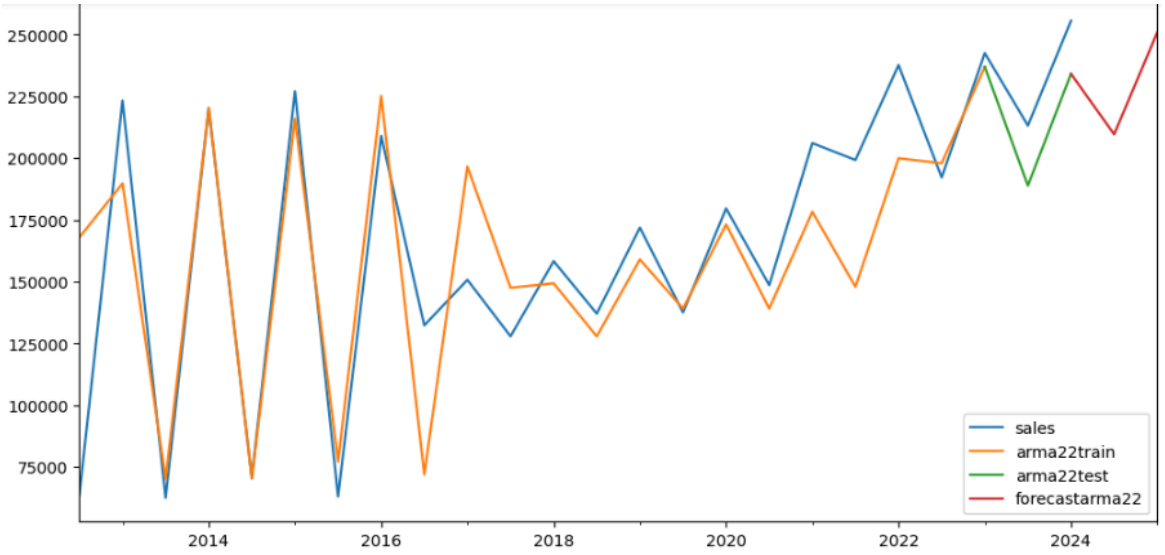
Έχοντας ολοκληρώσει τα παραπάνω συνδυάζουμε τα δυο αυτά μοντέλα της αυτοπαλινδρομισης και του κινητού μέσου ορού στο μοντέλο Arma(2,2). Το οποίο εφαρμόζεται με τον ίδιο τρόπο όπως τα δυο προηγούμενα αλλάζοντας απλά το order. Με βάση το p-value η μονή στατιστικά σημαντική μεταβλητή του μοντέλου είναι η δεύτερη lagged version του αυτοπαλινδρομου μοντέλου. Σχετικά με το μέσο τετραγωνικό σφάλμα του train και test set είναι μικρότερο από τα αντίστοιχα του ma και ar μοντέλου, οι προβλέψεις για τις πωλήσεις του 2024 είναι σχεδόν παρόμοιες(λίγο μικρότερες) με αυτές του 2023 και για τα δυο εξάμηνα

```
arma22=ARIMA(dfarma['sales'],order=(2,0,2))
arma22_fit=arma22.fit()
print(arma22_fit.summary())
dfarma.loc[train_idx,'arma22train']=arma22_fit.predict(start=train.index[0],end=train.index[-1])
dfarma.loc[test_idx,'arma22test']=arma22_fit.predict(start=test.index[0],end=test.index[-1])
mse_train_arma22=mean_squared_error(train['sales'],dfarma.loc[train_idx,'arma22train'])
mse_test_arma22=mean_squared_error(test['sales'],dfarma.loc[test_idx,'arma22test'])
print('mse train arma22=',mse_train_arma22)
print('mse test arma22=',mse_test_arma22)
forecastarma22=arma22_fit.forecast(steps=2)
forecastarma22
new_index_values = pd.to_datetime(['2024-07-01', '2025-01-01'])
new_rows = pd.DataFrame(index=new_index_values, columns=dfarma.columns)
dfarma = pd.concat([dfarma, new_rows])
dfarma['forecastarma22']=np.nan
dfarma.loc['2024-07-01', 'forecastarma22'] = forecastarma22[0]
dfarma.loc['2025-01-01', 'forecastarma22'] = forecastarma22[1]
dfarma
```

```
                               SARIMAX Results
==============================================================================
Dep. Variable:                   sales   No. Observations:                   24
Model:                 ARIMA(2, 0, 2)   Log Likelihood                -278.986
Date:                Sun, 19 Jan 2025   AIC                            569.971
Time:                        19:33:30   BIC                            577.040
Sample:                    07-01-2012   HQIC                           571.846
                         - 01-01-2024
Covariance Type:                  opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const       1.679e+05   6.49e+04      2.586      0.010    4.06e+04    2.95e+05
ar.L1         -0.0002      0.085     -0.002      0.998      -0.166       0.166
ar.L2          0.9184      0.154      5.956      0.000       0.616       1.221
ma.L1         -0.1199      0.226     -0.529      0.597      -0.564       0.324
ma.L2          0.1127      0.285      0.396      0.692      -0.445       0.671
sigma2      5.566e+08      9.026   6.17e+07      0.000    5.57e+08    5.57e+08
===================================================================================
Ljung-Box (L1) (Q):                   0.01   Jarque-Bera (JB):                 0.12
Prob(Q):                              0.91   Prob(JB):                         0.94
Heteroskedasticity (H):               2.78   Skew:                             0.10
Prob(H) (two-sided):                  0.17   Kurtosis:                         3.28
===================================================================================
```

2023-07-01	213170	NaN	188870.829398	NaN
2024-01-01	255608	NaN	234122.619757	NaN
2024-07-01	NaN	NaN	NaN	209624.311263
2025-01-01	NaN	NaN	NaN	250868.385335

```
mse train arma22= 1097377424.3301885
mse test arma22= 526035628.07862115
```



Εικονα 204: κώδικας και εφαρμογή arma(2,2)

Και ολοκληρώνουμε τις προβλέψεις με στατιστικά μοντέλα εφαρμόζοντας το Arima(2,1,2), δηλαδή το μοντέλο διαφοροποιεί τα δεόμενα μια φορά αυτό έχεις ως αποτέλεσμα με βάση το p-value να είναι στατιστικά σημαντικοί και οι δυο lagged versions του ma αλλά καμία μεταβλητή του ar μοντέλου. Είναι το πρώτο μοντέλο το οποίο δεν μπορεί να αναγνωρίσει την έντονη μεταβλητότητα των πρώτων εξάμηνων επίσης είναι το πρώτο μοντέλο οπού προβλέπει πωλήσεις για το 2023 (test set) μεγαλύτερες από της πραγματικές παρόλο που στην συνέχεια οι προβλέψεις για τις πωλήσεις του 2024 είναι μικρότερες από τις πραγματικές πωλήσεις του 2023. Όλα τα παραπάνω έχουν ως αποτέλεσμα το μοντέλο να αποδίδει χειροτέρα σε σχέση με το Arma(2,2) σε ορούς μέσου τετραγωνικού σφάλματος. Επίσης ολοκληρώνοντας την ενότητα θα πρέπει να αναφέρουμε ότι έγινε δοκιμή εφαρμογής όλων των παραπάνω μοντέλων παίρνοντας τη πρώτη διαφορά των δεδομένων (τελικές – αρχικές πωλήσεις) και επαναφοράς των προβλέψεων στην κανονική μορφή και τα αποτελέσματα ήταν λιγότερο αποτελεσματικά σε σχέση με τα μη διαφοροποιημένα δεδομένα.

```
arima212=ARIMA(dfarima['sales'],order=(2,1,2))
arima212_fit=arima212.fit()
print(arima212_fit.summary())
dfarima.loc[train_idx,'arima212train']=arima212_fit.predict(start=train.index[0],end=train.index[-1])
dfarima.loc[test_idx,'arima212test']=arima212_fit.predict(start=test.index[0],end=test.index[-1])
mse_train_arima212=mean_squared_error(train['sales'],dfarima.loc[train_idx,'arima212train'])
mse_test_arima212=mean_squared_error(test['sales'],dfarima.loc[test_idx,'arima212test'])
print('mse train arima22=',mse_train_arima212)
print('mse test arima22=',mse_test_arima212)
forecastarima212=arima212_fit.forecast(steps=2)
forecastarima212
new_index_values = pd.to_datetime(['2024-07-01', '2025-01-01'])
new_rows = pd.DataFrame(index=new_index_values, columns=dfarima.columns)
dfarima = pd.concat([dfarima, new_rows])
dfarima['forecastarima212']=np.nan
dfarima.loc['2024-07-01', 'forecastarima212'] = forecastarima212[0]
dfarima.loc['2025-01-01', 'forecastarima212'] = forecastarima212[1]
dfarima
```

SARIMAX Results

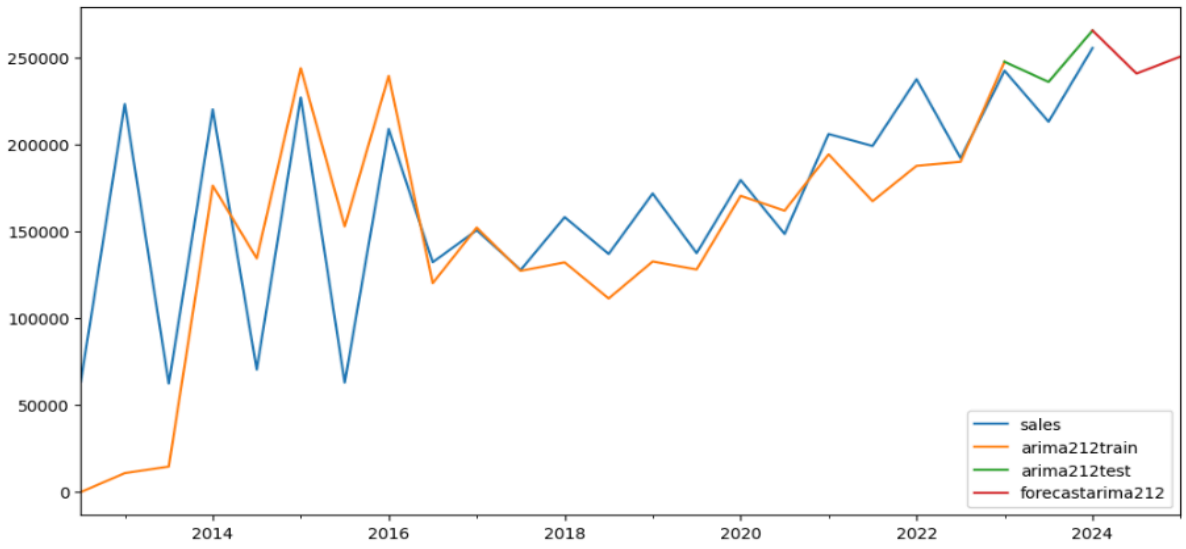
Dep. Variable:	sales	No. Observations:	24
Model:	ARIMA(2, 1, 2)	Log Likelihood	-280.972
Date:	Sun, 19 Jan 2025	AIC	571.945
Time:	19:39:07	BIC	577.622
Sample:	07-01-2012	HQIC	573.372
	- 01-01-2024		
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
ar.L1	0.1177	0.282	0.417	0.677	-0.436	0.671
ar.L2	0.4911	0.349	1.407	0.159	-0.193	1.175
ma.L1	-1.6349	0.230	-7.098	0.000	-2.086	-1.183
ma.L2	1.0000	0.192	5.199	0.000	0.623	1.377
sigma2	2.301e+09	1.53e-10	1.5e+19	0.000	2.3e+09	2.3e+09

Ljung-Box (L1) (Q):	2.26	Jarque-Bera (JB):	23.27
Prob(Q):	0.13	Prob(JB):	0.00
Heteroskedasticity (H):	0.11	Skew:	1.28
Prob(H) (two-sided):	0.00	Kurtosis:	7.21

2023-07-01	213170	NaN	236070.946991	NaN
2024-01-01	255608	NaN	265669.986912	NaN
2024-07-01	NaN	NaN	NaN	240881.766067
2025-01-01	NaN	NaN	NaN	250685.595177

```
mse train arima22= 3354022380.677275
mse test arima22= 312848476.84159374
```



Εικονα 205: κωδικας και εφαρμογη arima(2,1,2)

Το επόμενο μοντέλο που εφαρμόζουμε είναι τα νευρωνικα δίκτια rnn με κύτταρα lstm αρχικά εισάγουμε τις βιβλιοθήκες TensorFlow ,sklearn στην συνέχεια κάνουμε scale τα δεδομένα με τον minmaxscaler ώστε να περνούν τιμές μεταξύ του μηδέν και του ένα στην συνέχεια εφαρμόζουμε την pre processing διαδικασία ώστε τα δεδομένα να γίνουν insert στο μοντέλο το οποίο έχει ως optimizer το Adam loss function το mse activation την συνάρτηση relu και αποτελείται από δυο συστοιχίες νευρώνων lstm η πρώτη με 128 κύτταρα και η δεύτερη με 64, όσον αφορά την εκπαίδευση των κυττάρων χρησιμοποιούνται 200 epochs και το batch size είναι ίσο με 16. Στην συνέχεια κάνουμε τις προβλέψεις για το train, test set και για τις πωλήσεις του 2024 και τις επαναφέρουμε στην αρχική τους κατάσταση. Όσον αφορά το τετραγωνικό σφάλμα δεν είναι μικρότερο από εκείνο των μοντέλων μηχανικής μάθησης όπως της γραμμικής παλινδρόμησης. Σχετικά με της προβλέψεις των πωλήσεων του 2023 (train set) το μοντέλο υπερεκτίμα τις πωλήσεις του δευτέρου εξάμηνου ενώ υποεκτίμα της πωλήσεις του πρώτου εξάμηνου επηρεασμένο από την μεγάλη μεταβλητότητα των πρώτων ετών, ενώ αναφορικά με της πωλήσεις για το 2024 το μοντέλο εκτιμά ότι οι πωλήσεις του πρώτου εξάμηνου θα είναι λίγο λιγότερες από τις αντίστοιχες του 2023 ενώ εκτιμά ότι οι πωλήσεις του δευτέρου εξάμηνου θα είναι πολύ μεγαλύτερες από αυτές του 2023.

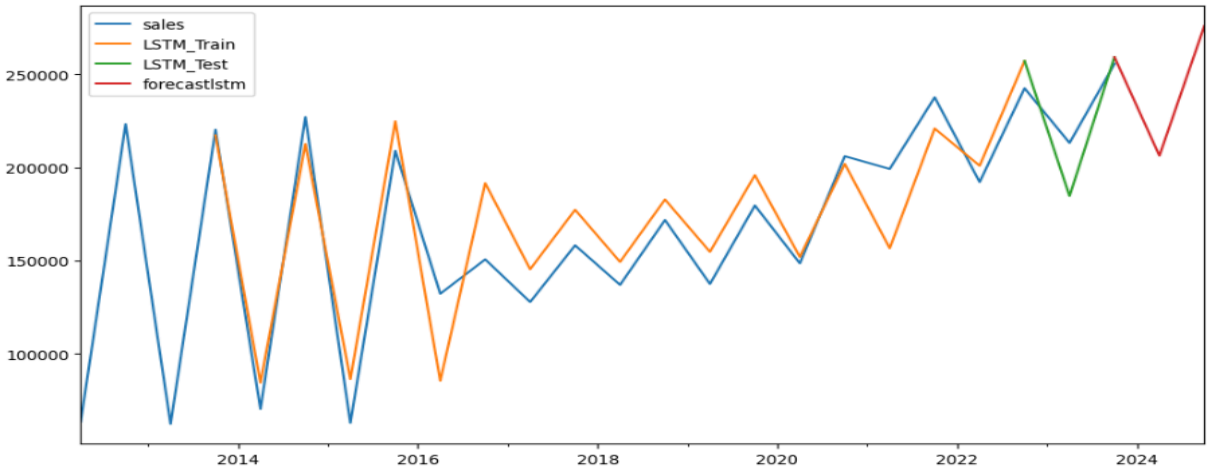
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import LSTM, Dense
from sklearn.preprocessing import MinMaxScaler
df=pd.read_excel('plaisio2.xlsx')
df['time'] = pd.to_datetime(df['time'])
df.set_index('time', inplace=True)
data = df['sales'].values.reshape(-1, 1)
scaler = MinMaxScaler(feature_range=(0, 1))
data_scaled = scaler.fit_transform(data)
T = 3
train_size = len(data) - 2
train_data = data_scaled[:train_size]
test_data = data_scaled[train_size - T:]

def create_sequences(data, T):
    X, Y = [], []
    for i in range(len(data) - T):
        X.append(data[i:i + T])
        Y.append(data[i + T])
    return np.array(X), np.array(Y)
X_train, Y_train = create_sequences(train_data, T)
X_test, Y_test = create_sequences(test_data, T)
model = Sequential([
    LSTM(128, activation='relu', input_shape=(T, 1)),
    Dense(64, activation='relu'),
    Dense(1)
])
model.compile(optimizer='adam', loss='mse')
model.fit(X_train, Y_train, epochs=200, batch_size=16, verbose=1)
train_predictions = model.predict(X_train)
test_predictions = model.predict(X_test)
```

```
model.compile(optimizer='adam', loss='mse')
model.fit(X_train, Y_train, epochs=200, batch_size=16, verbose=1)
train_predictions = model.predict(X_train)
test_predictions = model.predict(X_test)
train_predictions = scaler.inverse_transform(train_predictions)
test_predictions = scaler.inverse_transform(test_predictions)
Y_train_actual = scaler.inverse_transform(Y_train)
Y_test_actual = scaler.inverse_transform(Y_test)
future_values = []
last_sequence = data_scaled[-T:].reshape(1, T, 1)
for _ in range(2):
    next_value = model.predict(last_sequence)
    future_values.append(next_value[0, 0])
    next_value_scaled = np.append(last_sequence[0, 1:], next_value).reshape(1, T, 1)
    last_sequence = next_value_scaled
future_values = scaler.inverse_transform(np.array(future_values).reshape(-1, 1))
train_idx = df.index[:train_size]
test_idx = df.index[train_size:]
future_idx = pd.date_range(start=df.index[-1], periods=3, freq='6MS')[1:]
df.loc[train_idx[T:], 'LSTM_Train'] = train_predictions.flatten()
df.loc[test_idx, 'LSTM_Test'] = test_predictions.flatten()
forecast_df = pd.DataFrame({'Forecast': future_values.flatten()}, index=future_idx)
```

2022-12-31	242499	2[redacted]	257117.250000	NaN	NaN
2023-06-30	213170	1[redacted]	NaN	184702.781250	NaN
2023-12-31	255608	2[redacted]	NaN	259228.390625	NaN
2024-06-30	NaN	[redacted]	NaN	NaN	206323.421875
2024-12-31	NaN	[redacted]	NaN	NaN	275948.281250

```python
from sklearn.metrics import mean_squared_error
mse_train_lstm = mean_squared_error(Y_train_actual, train_predictions)
mse_test_lstm = mean_squared_error(Y_test_actual, test_predictions)
print(f"Train MSE (LSTM): {mse_train_lstm}")
print(f"Test MSE (LSTM): {mse_test_lstm}")
```

```
Train MSE (LSTM): 558448564.7892516
Test MSE (LSTM): 176692316.97875977
```



εικόνα 206: Κώδικας και εφαρμογή lstm

Τέλος ολοκληρώνουμε την ανάλυση των πωλήσεων με την εφαρμογή του μοντέλου Facebook prophet.Όπως μπορούμε να δούμε τα αποτελέσματα όσων αφορά τις προβλέψεις για το test set και train set δεν είναι καθόλου κοντά στις πραγματικές τιμές της χρονοσειρας. Το μοντέλο φαίνεται να επηρεάζεται από την αρχική έντονη διακύμανση και στην συνέχεια δεν μπορεί να προσαρμοστεί στην εξομάλυνση της. Τέλος οι προβλέψεις για το 2024 του μοντέλου έχουν πτωτική πορεία.

```
dfprophet=pd.read_excel('plaisio2.xlsx')
dfprophet.drop(columns=['costs'],inplace=True)
dfprophet.rename(columns={'time':'ds','sales':'y'},inplace=True)
!pip install prophet
from prophet import Prophet
ntest=2
train=dfprophet.iloc[:-ntest]
test=dfprophet.iloc[-ntest:]
from sklearn.metrics import mean_squared_error
m = Prophet()
m.fit(train)
future = m.make_future_dataframe(periods=ntest)
forecast = m.predict(future)
train_predictions = forecast['yhat'][:-ntest]
test_predictions = forecast['yhat'][-ntest:]
mse_train = mean_squared_error(train['y'], train_predictions)
mse_test = mean_squared_error(test['y'], test_predictions)
print(f"Train MSE: {mse_train}")
print(f"Test MSE: {mse_test}")
fig1 = m.plot(forecast)
fig2 = m.plot_components(forecast)
```

```
Train MSE: 1036366614.6311299
Test MSE: 650741150.7423558
```

```
dfprophet['prophet_train'] = None
dfprophet['prophet_test'] = None
train_idx = dfprophet['ds'] <= train['ds'].iloc[-1]
test_idx = ~train_idx
dfprophet.loc[train_idx, 'prophet_train'] = train_predictions.values
dfprophet.loc[test_idx, 'prophet_test'] = test_predictions.values
dfprophet
```

```
future_two = m.make_future_dataframe(periods=4)
forecast_two = m.predict(future_two)
next_two_predictions = forecast_two['yhat'][-4:]
next_two_predictions
```

	yhat
22	241965.243429
23	233875.161774
24	216273.799940
25	189775.396016

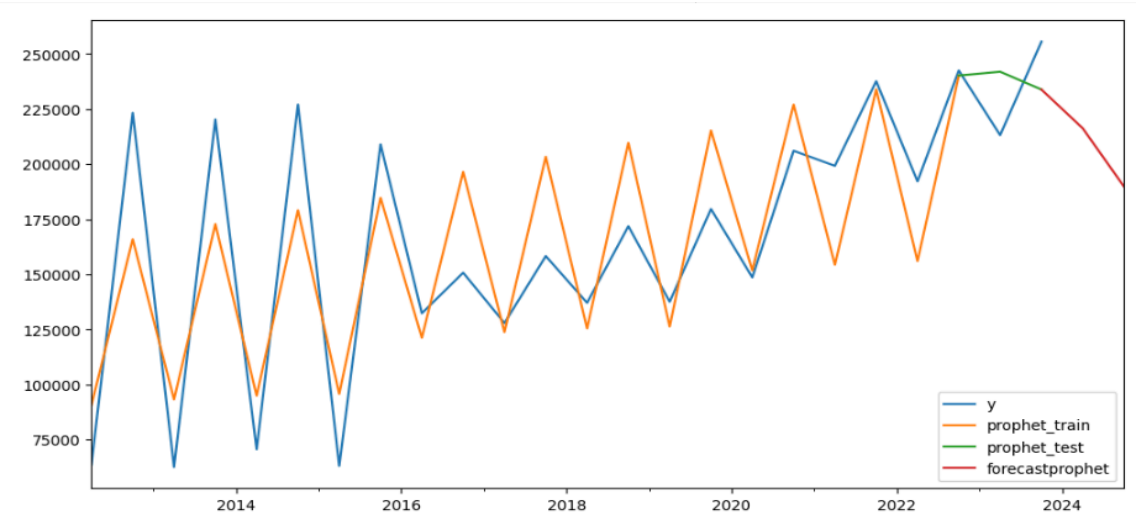
dtype: float64

```
dfprophet['ds'] = pd.to_datetime(dfprophet['ds'])
dfprophet.set_index('ds', inplace=True)
```

```
new_index_values = pd.to_datetime(['2024-06-30', '2024-12-31'])
new_rows = pd.DataFrame(index=new_index_values, columns=dfprophet.columns)
dfprophet = pd.concat([dfprophet, new_rows])
dfprophet['forecastprophet']=np.nan
```

```
dfprophet.loc[dfprophet.index[-3:], 'forecastprophet'] = next_two_predictions.values[1:]
```

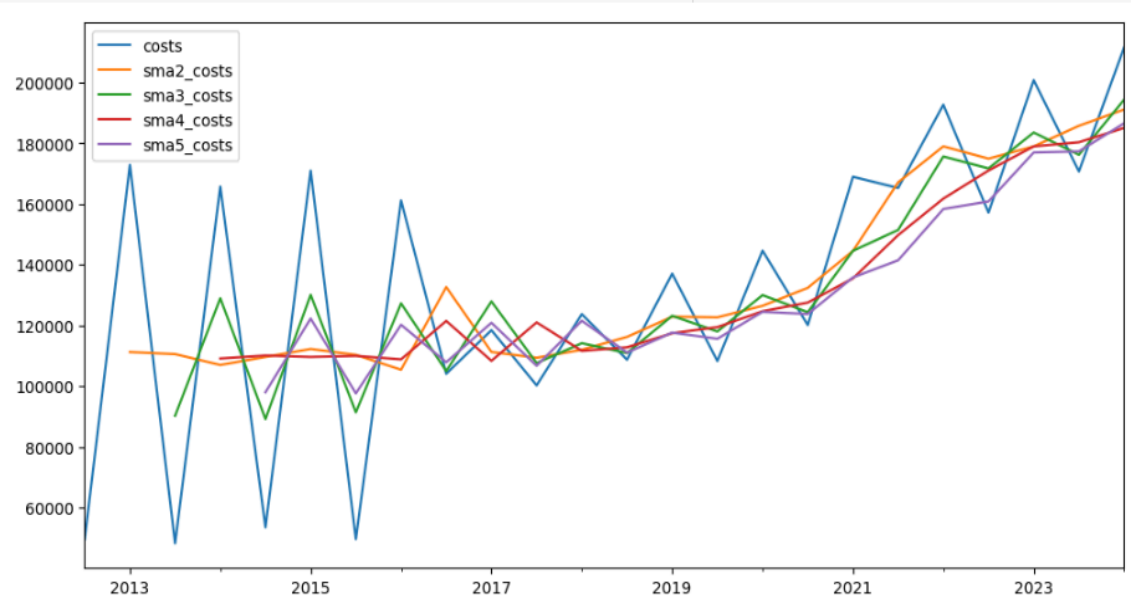
2023-06-30	213170	None	241965.243429	NaN
2023-12-31	255608	None	233875.161774	233875.161774
2024-06-30	NaN	NaN	NaN	216273.799940
2024-12-31	NaN	NaN	NaN	189775.396016



Εικόνα 207: κώδικας και εφαρμογή Facebook prophet

Το συμπέρασμα αναφορικά με τις πωλήσεις του πλαίσιο και την μοντελοποιηση τους είναι ότι είναι αρκετά δύσκολο να προβλεφθούν ακριβός οι πωλήσεις ακόμα και με μοντέλα νευρωνικων δικτυών όπως τα lstm ο βασικός παράγοντας είναι ότι δεν υπάρχουν αρκετά timesteps(λίγες παρατηρήσεις μέσα στο έτος) δηλαδή δεν γνωρίζουμε την πορεία των πωλήσεων ανά ημέρα, μηνά, τρίμηνο παραμονο τις εξαμηνίες πωλήσεις με αποτέλεσμα τα παραπάνω μοντέλα να δυσκολεύονται να κατανοήσουν την συνολικη τάση, επίσης δεν υπάρχουν πληροφορίες αναφορικά με τις πωλήσεις συγκεκριμένων τύπων προϊόντων όπως ηλεκτρικές συσκευές, λευκές συσκευές ,είδη γραφείου κλπ. ξεχωριστά για την κάθε μια, επίσης ένας άλλος σημαντικός παράγοντας είναι η έλλειψη εξωγενών μεταβλητών όπως για παράδειγμα το ΑΕΠ της Ελλάδας ή μεταβλητές όπως το επίπεδο του ανταγωνισμού, επίσης αν παρατηρήσουμε κανένα μοντέλο δεν μπόρεσε να προβλέψει εξωγενείς παράγοντες όπως ο covid που αύξησε της πωλήσεις το 2020 με 2021. Παρόλα αυτά οι προβλέψεις των μοντέλων για κάποιον αναλυτή που δεν έχει εσωτερική πληροφόρηση είναι ρεαλιστικές ένα έπρεπε να επιλέξω κάποια οπού εκτιμά ότι οι πωλήσεις θα αυξηθούν το 2024 θα επέλεγα τις προβλέψεις του μοντέλου lstm, εάν από την άλλη πλευρά έπρεπε να επιλέξω μια πρόβλεψη οπού δείχνει ότι οι πωλήσεις μειώνεται το 2024 πχ χαμηλότερη ζήτηση λόγο της μη ύπαρξης επιδοτούμενων προγραμμάτων όπως το αλλάζω συσκευή τότε θα επέλεγα τις προβλέψεις του Arma(2,2).

Στην συνέχεια ακολουθούν οι υπολογισμοί των μέσων τετραγωνικών σφαλμάτων και τα διαγράμματα των μοντέλων για την πρόβλεψη του κόστους πωληθέντων, χωρίς τον κώδικα καθώς είναι ίδιος με αυτόν που χρησιμοποιήθηκε για τις πωλήσεις. Επίσης επειδή οι πωλήσεις και το κόστος πωληθέντων έχουν σχεδόν τελειά θετική συσχέτιση τα συμπεράσματα είναι παρόμοια.

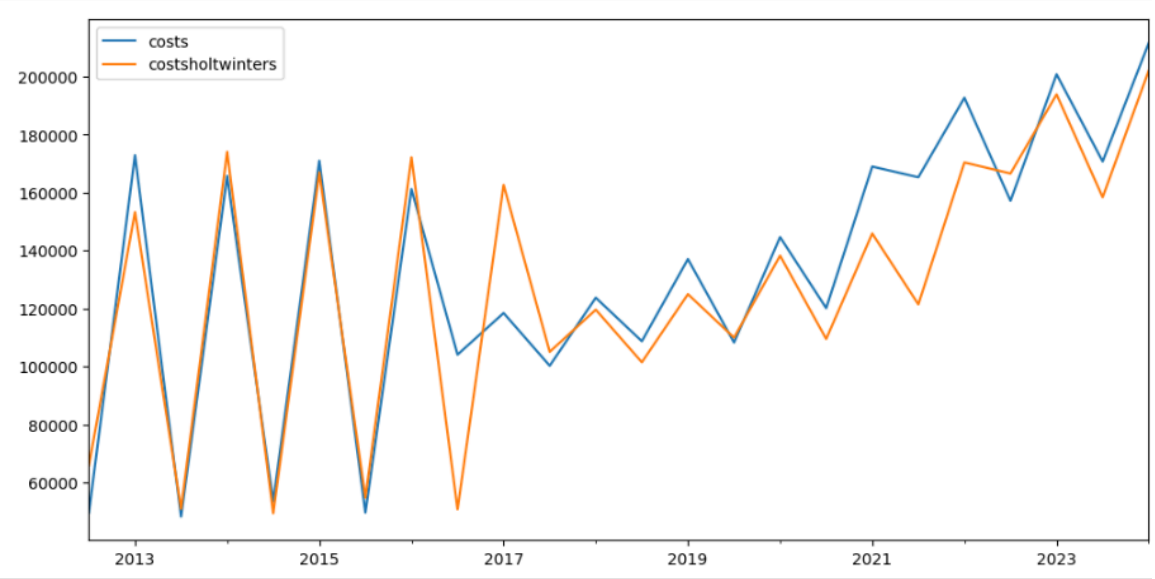



```
MSE for SMA2: 1251508478.423913
MSE for SMA3: 528581099.4090908
MSE for SMA4: 1061816332.9166666
MSE for SMA5: 667612150.4239998
Minimum MSE: 528581099.4090908
```

Εικονα 208: sma


```
MSE for WMA0.2: 1508767734.6864684
MSE for WMA0.3: 823661130.9017844
MSE for WMA0.4: 433686211.1213725
MSE for WMA0.8: 139487753.32785067
Minimum MSE: 139487753.32785067
```

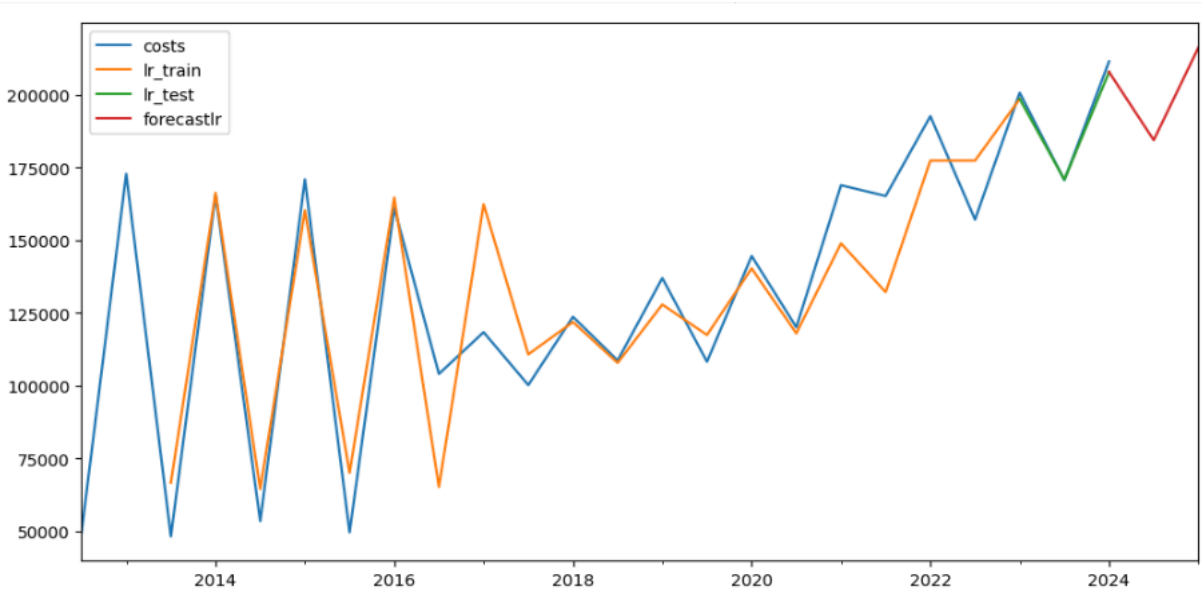
Εικονα 209: wma




```
mseholtwinters=mean_squared_error(dfcosts['costs'], dfcosts['costsholtwinters'])
print('mse for holtwinters=',mseholtwinters)
```

```
mse for holtwinters= 393286029.97742814
```

Εικονα 210:Holt winters

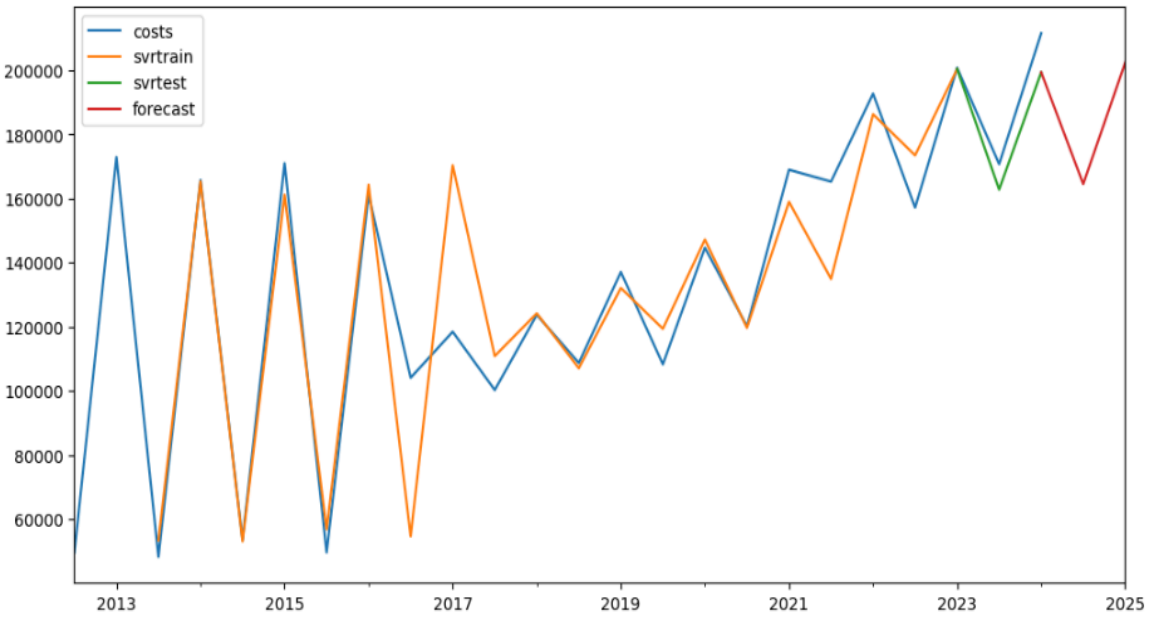


```
mselr=mean_squared_error(dfcostslr['costs'][2:-2],dfcostslr['lr_train'][2:-2])
print('mse for lr=',mselr)
```

```
mse for lr= 345136026.02144533
```

2023-07-01	170728	NaN	170879.812415	NaN
2024-01-01	211542	NaN	207903.919716	207903.919716
2024-07-01	NaN	NaN	NaN	184476.239583
2025-01-01	NaN	NaN	NaN	216306.062980

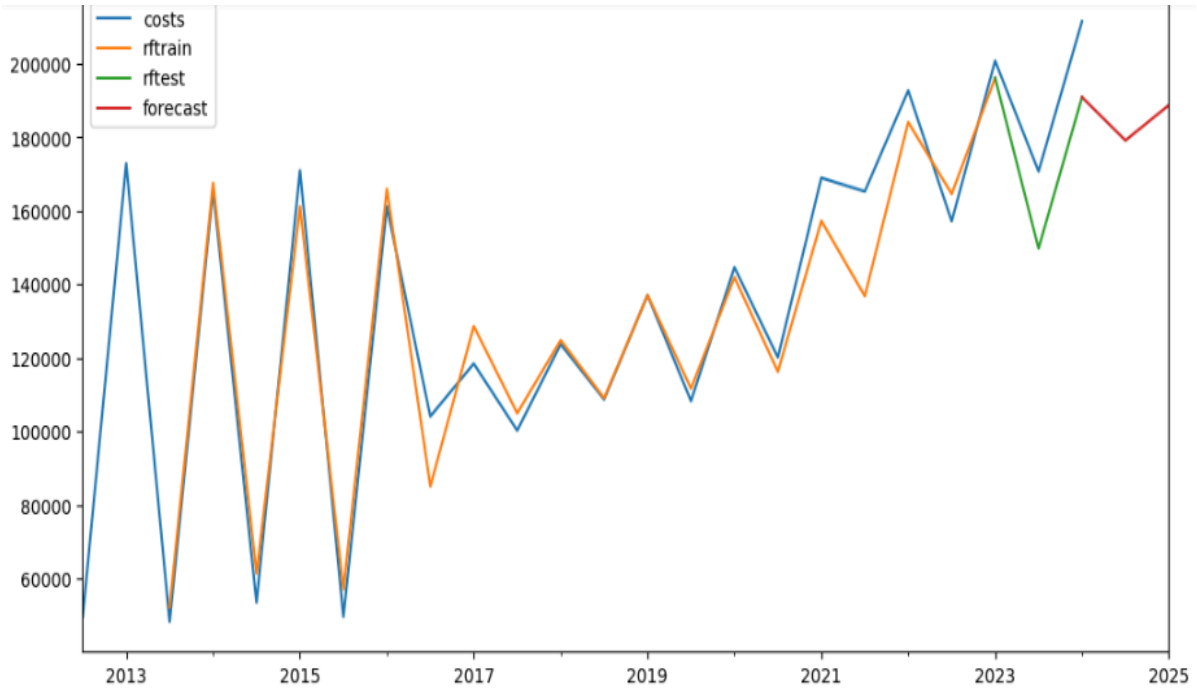
Εικονα 210: γραμμική παλινδρόμηση




```
Train MSE: 346311846.75225943
Test MSE: 105720171.63370265
```

2023-07-01	170728.0	None	162804.094092	NaN
2024-01-01	211542.0	None	199349.704956	NaN
2024-07-01	NaN	NaN	NaN	164513.336515
2025-01-01	NaN	NaN	NaN	202430.612170

Εικονα 211: svr

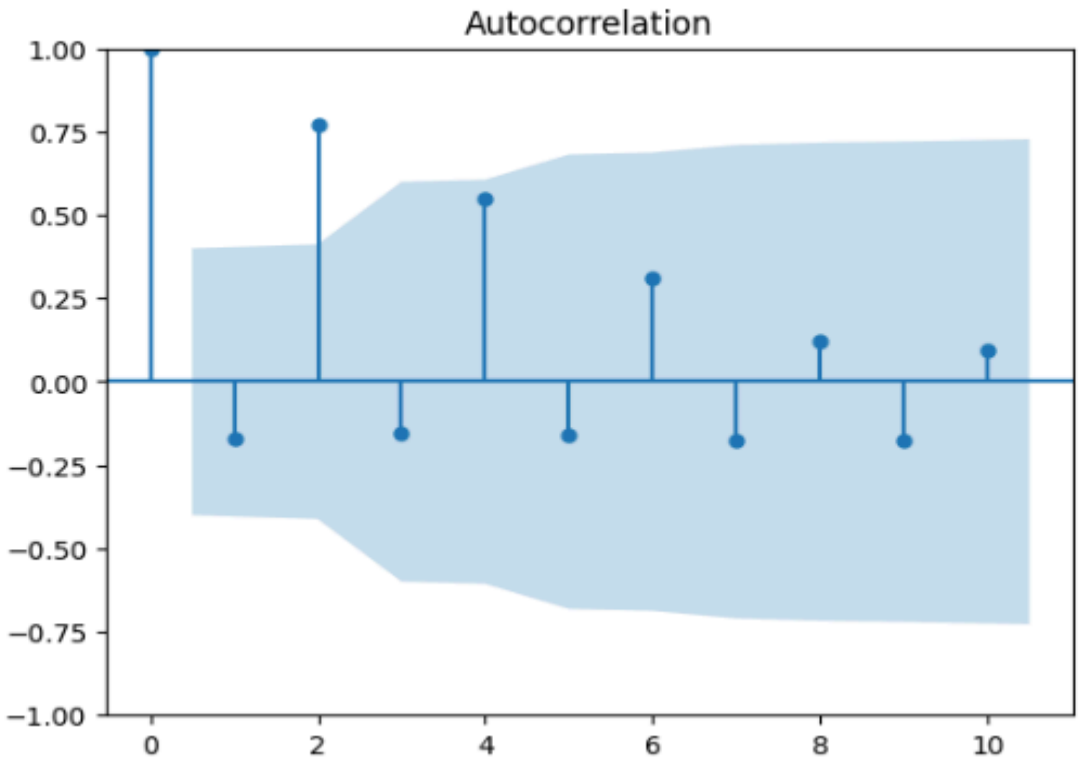



```
Train MSE (Random Forest): 93817920.59168516
Test MSE (Random Forest): 432547108.5514517
```

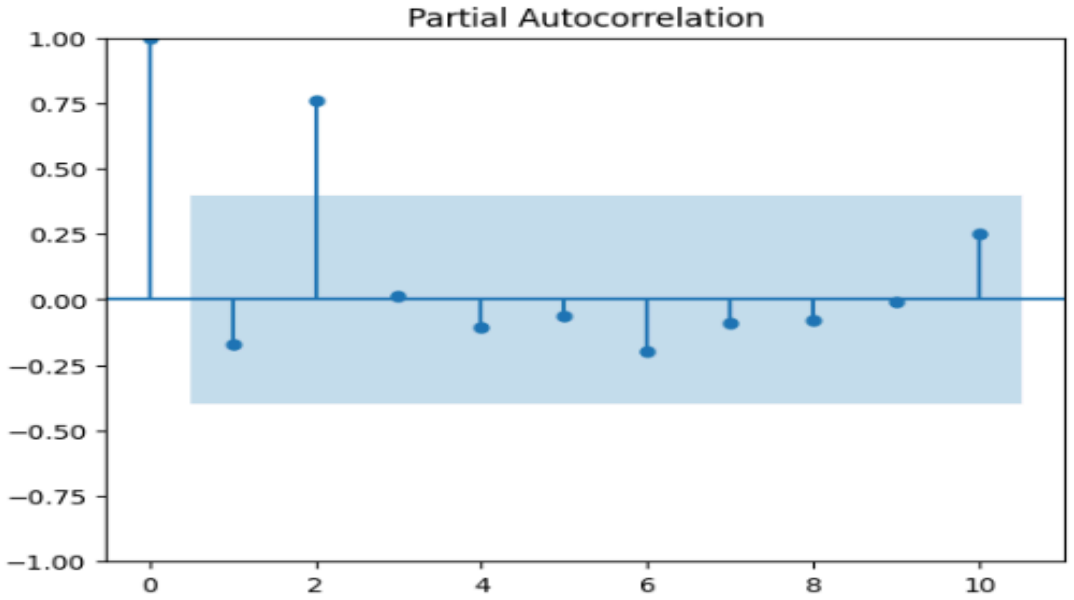
2023-07-01	170728.0	None	149779.27	NaN
2024-01-01	211542.0	None	190896.3	NaN
2024-07-01	NaN	NaN	NaN	179143.78
2025-01-01	NaN	NaN	NaN	188749.80

Εικονα 212: random forest regressor

```
plot_acf(dfcostsar,lags=10)
plt.show()
```

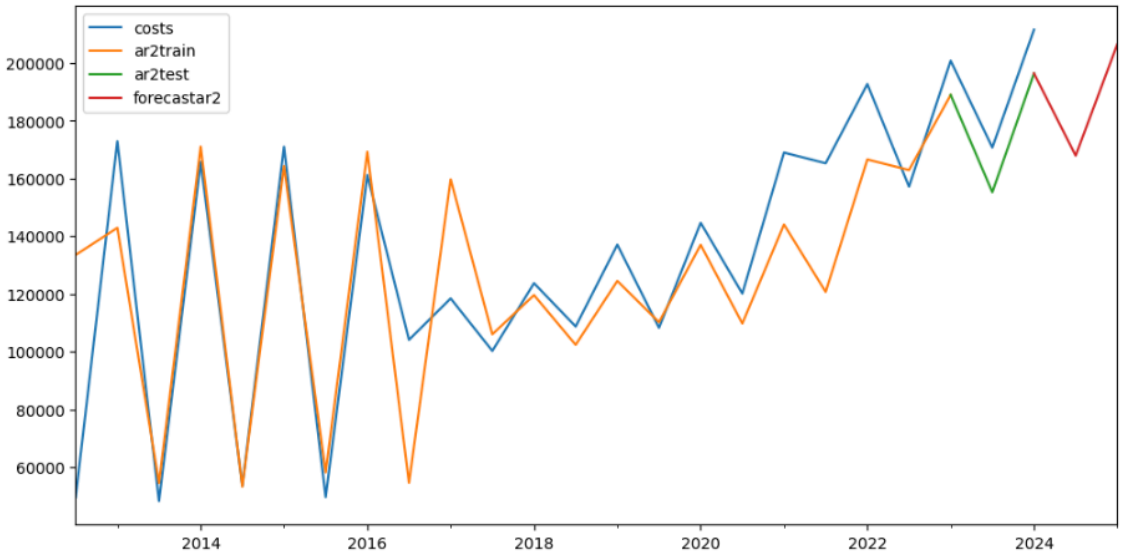



```
plot_pacf(dfcostsar,lags=10)
plt.show()
```



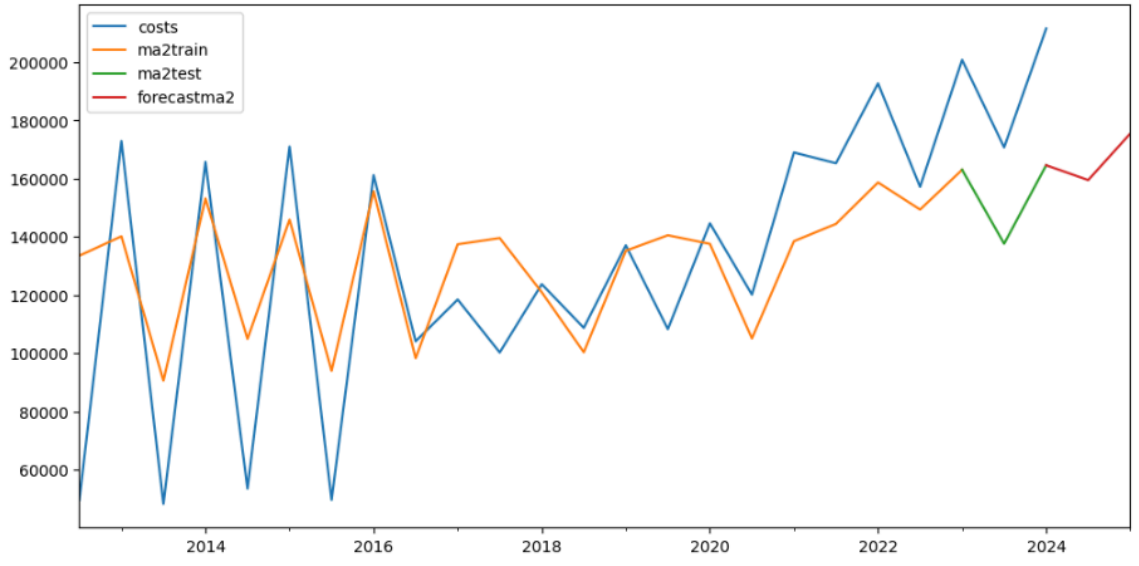
```
ADF Statistic: 0.5874707962270203
p-value: 0.9873026997856753
Critical Values:
        1% : -3.7883858816542486
        5% : -3.013097747543462
        10% : -2.6463967573696143
```

Εικονα 213: pacf , acf , adf test



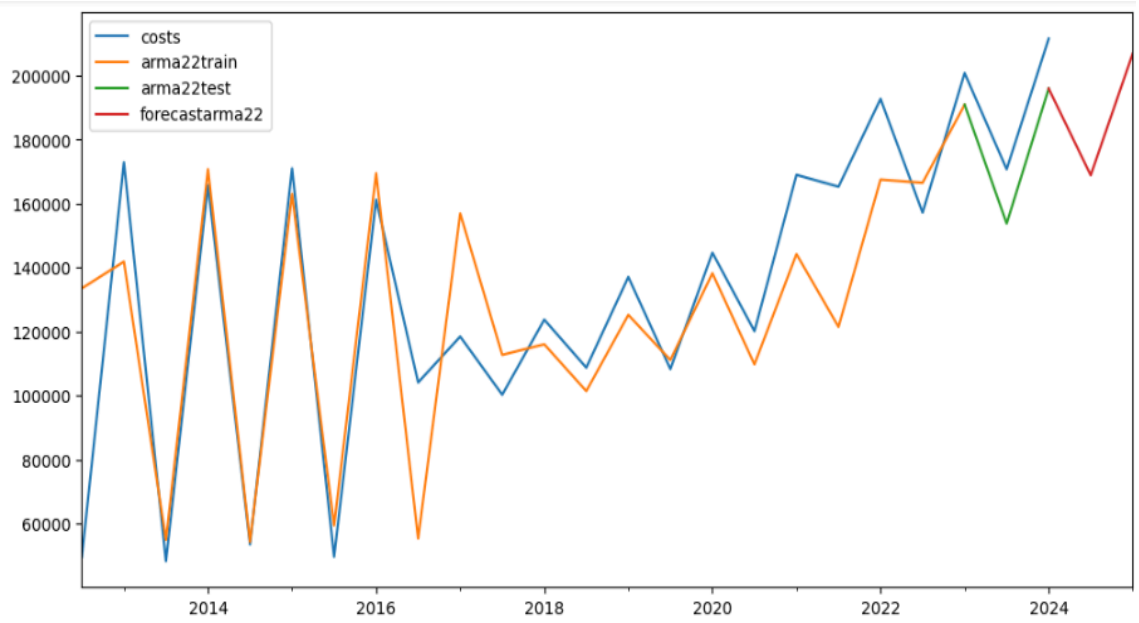

```
mse train ar2= 737692913.3600516
mse test ar2= 234516768.69122392
```

Εικονα 214: εφαρμογή ar(2)



```
mse train ma2= 1030828922.9064164
mse test ma2= 1650750954.2700849
```

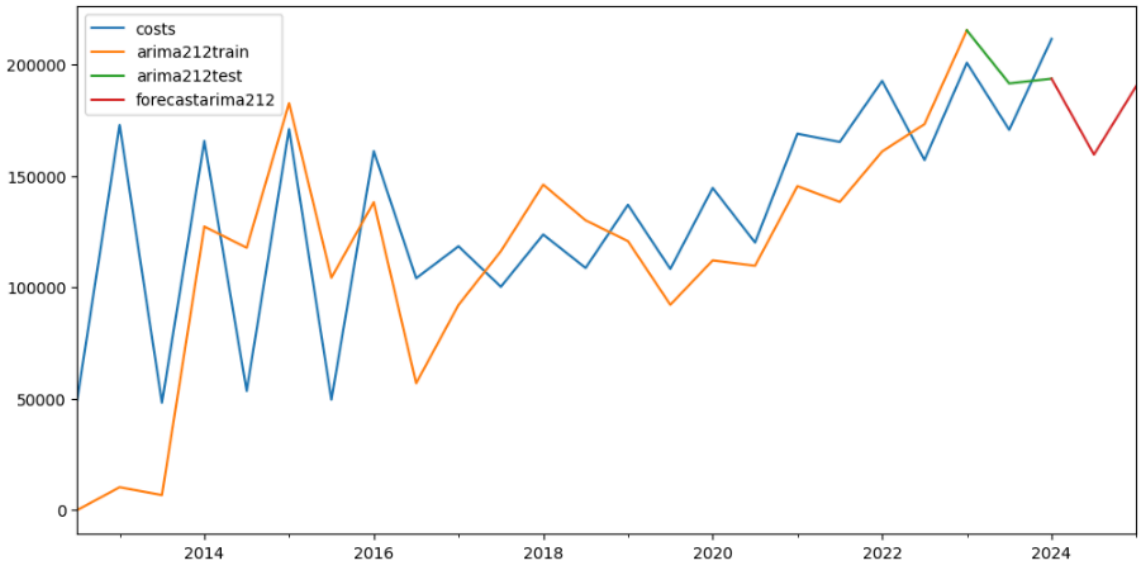
Εικονα 215: εφαρμογή ma(2)




```
mse train arma22= 731058377.4511694
mse test arma22= 265844046.47571173
```

2023-07-01	213170	170728	NaN	153766.011557	NaN
2024-01-01	255608	211542	NaN	195922.171544	NaN
2024-07-01	NaN	NaN	NaN	NaN	168859.905056
2025-01-01	NaN	NaN	NaN	NaN	206914.634231

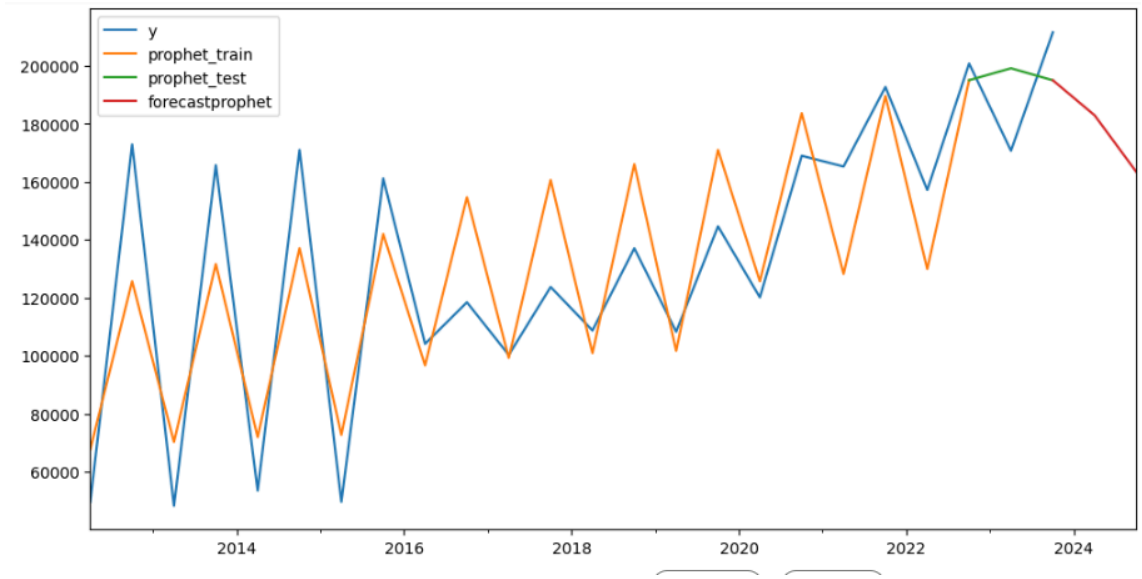
Εικονα 216: εφαρμογή arma(2,2)




```
mse train arima212= 2203335744.6675506
mse test arima212= 375860804.9407697
```

2023-07-01	213170	170728	NaN	191550.324115	NaN
2024-01-01	255608	211542	NaN	193705.172133	NaN
2024-07-01	NaN	NaN	NaN	NaN	159596.652941
2025-01-01	NaN	NaN	NaN	NaN	190394.675977

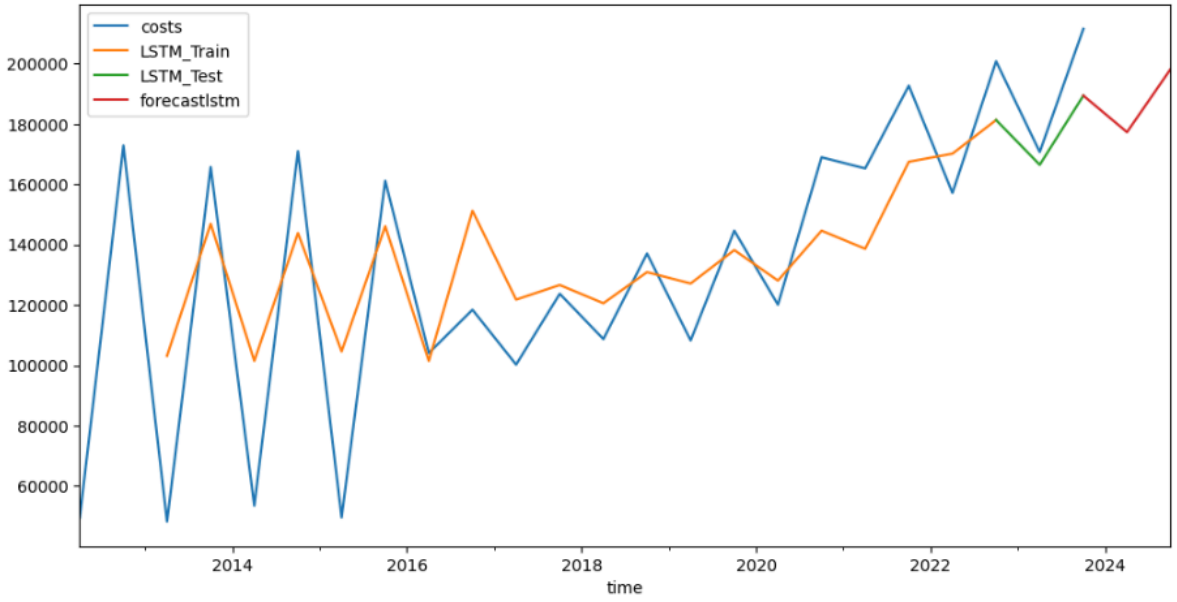
Εικονα 217: εφαρμογή arima(2,1,2)



2023-06-30	213170.0	170728.0	NaN	166474.703125	NaN
2023-12-31	255608.0	211542.0	NaN	189404.234375	NaN
2024-06-30	NaN	NaN	NaN	NaN	177292.43750
2024-12-31	NaN	NaN	NaN	NaN	198224.09375

```
Train MSE: 607708302.1492254
Test MSE: 540548911.2378215
```

Εικονα 214: εφαρμογή facebook prophet




```
Train MSE (LSTM): 717110850.3084565
Test MSE (LSTM): 254085600.5871582
```

2023-06-30	170728	None	199158.700515	NaN
2023-12-31	211542	None	195025.550904	195025.550904
2024-06-30	NaN	NaN	NaN	182908.270934
2024-12-31	NaN	NaN	NaN	163304.275188

Εικονα 215: αποτελέσματα lstm

8. ΣΥΜΠΕΡΑΣΜΑΤΑ

Η παρούσα εργασία ανέδειξε τον καίριο ρόλο που διαδραματίζει η ανάλυση της ζήτησης και η πρόβλεψή της στην αποτελεσματική λειτουργία της εφοδιαστικής αλυσίδας. Αρχικά, έγινε μια εκτενής εισαγωγή στις βασικές στατιστικές έννοιες και στις βάσεις δεδομένων, ώστε ο αναγνώστης να μπορεί να κατανοήσει τον τρόπο με τον οποίο συλλέγονται, οργανώνονται και αξιοποιούνται τα δεδομένα για τη λήψη αποφάσεων. Μέσα από τη σχεδίαση και την ανάλυση μιας σχεσιακής βάσης δεδομένων για ένα κατάστημα ηλεκτρονικών ειδών, αναδείχθηκε η σημασία της σωστής μοντελοποίησης και της χρήσης εννοιών όπως τα πρωτεύοντα και ξένα κλειδιά, καθώς και οι σχέσεις μεταξύ των πινάκων, προκειμένου να διασφαλίζεται η ακεραιότητα και η συνέπεια των πληροφοριών. Στην συνέχεια αναλυθήκαν απλοί μέθοδοι περιγραφικής στατιστικής όπως τα διαστήματα εμπιστοσύνης και οι έλεγχοι υποθέσεων ενώ ταυτόχρονα έγιναν αρκετά παραδείγματα πάνω σε προβλήματα που προκύπτουν στην εφοδιαστική αλυσίδα. Στην συνέχεια αναλυθήκαν θεωρητικά μοντέλα μηχανικής μάθησης όπως τα η γραμμική παλινδρόμηση και τα δέντρα αποφάσεων και αναδείχτηκε η σημασία της χρήσης τους στην εφοδιαστική αλυσίδα. Στο επόμενο κεφάλαιο έγινε μια σύντομη αναφορά σε βασικές βιβλιοθήκες της python όπως την NumPy και την pandas οι οποίες χρησιμοποιήθηκαν κατά κύριο λόγο για την ανάπτυξη μοντέλων μηχανικής μάθησης και ανάλυσης των δεδομένων όπως στο κεφάλαιο 6 για την ανάλυση της πιθανότητας επαναγοράς των πελατών ενός ηλεκτρονικού καταστήματος. Στην συνέχεια έγινε ανάλυση στα χαρακτηριστικά των χρονομέτρων όπως η τάση , η εποχικότητα , η κυκλικότητα και η αυτοσυσχετηση , ενώ αναλύθηκε δεξιώθηκα η χρονοσειρα της τιμής του Samsung galaxy s24 καθώς ένας από τους σημαντικότερους προσδιοριστικούς παράγοντες της ζήτησης είναι η τιμή του προϊόντος. Στο κεφαλαίο 8 αναλυθήκαν τα χαρακτηριστικά των λαπτοπ οπού είναι διακείμενοι οι πελάτες να πληρώσουν παραπάνω για να αποκτήσουν το λαπτοπ που τα συμπεριλαμβάνει. Τέλος στο κεφάλαιο 9 αναλύθηκε ο ισολογισμός της εταιρίας πλαίσιο οπού δραστηριοποιείται στον χώρο των λιανικών πωλήσεων ηλεκτρικών συσκευών και ειδών γραφείου , πιο συγκεκριμένα αναλυθήκαν αριθμοδείκτες και αναδειχτήκαν μελλοντικές ευκαιρίες αλλά και κίνδυνοι , ενώ τέλος με βάση την θεωρεία του κεφαλαίου 7 προβλεφτικαν οι μελλοντικές πωλήσεις του ομίλου.

Προφανώς θα μπορούσαν να είχαν χρησιμοποιηθεί για την ανάλυση της ζήτησης αλλά σύνολα δεδομένων πιο συγκεχυμένα η ανάλυση θα ήταν πιο βέλτιστη αν τα δεδομένα

προέρχονταν εσωτερικά από μια επιχείρηση. Επίσης θα μπορούσαν να είχαν χρησιμοποιηθεί και άλλες μέθοδοι ανάλυσης όπως εξωγενές μεταβλητές στο μοντέλο Arima. Επίσης μέσα από την παρούσα εργασία αναδείχτηκε ότι σύνθετα μοντέλα όπως το lstm δεν είναι πάντα καλυτέρα σε σχέση με τα πιο απλά όπως το holt winters.

Βιβλιογραφία

1) ΔΗΜΕΛΗ ΣΟΦΙΑ 2013 ΣΥΧΡΟΝΕΣ ΜΕΘΟΔΟΙ ΑΝΑΛΥΣΗΣ ΧΡΟΝΟΛΟΓΙΚΩΝ ΣΕΙΡΩΝ ΑΘΗΝΑ ΟΠΑ

2) SEAN J TAYLOR, BENJAMIN LETHAM FORCASTING AT SCALE https://peerj.com/preprints/3190/

3) Christopher Olah. (2015). Understanding lstm networks. http://colah.github.io/posts/2015-08-Understanding-LSTMs/

4) ΔΗΜΗΤΡΗΣ ΒΑΣΙΛΕΙΟΥ , ΝΙΚΟΛΑΟΣ ΗΡΕΙΩΤΗΣ ΑΝΑΛΥΣΗ ΕΠΕΝΔΥΣΕΩΝ ΚΑΙ ΔΙΑΧΕΙΡΙΣΗ ΧΑΡΤΟΦΥΛΑΚΙΟΥ ΕΔΟΣΕΙΣ ROSILI

5) H2o.ai CONFUSION MATRIX https://h2o.ai/wiki/confusion-matrix/

6) MATTHEW F.DIXON , IGOR HALPERIN , PAUL BILOKON MACHINE LEARNING IN FINANCE FROM THEORY TO PRACTICE SPRINGER

7) JAYDIP SEN, RAJDEEP SEN, ABHISHEK DUTTA MACHINE LERANING IN FINANCE-EMERGING TRENDS AND CHALLENGES

8) πλαισιο οικονομικα reports: https://www.plaisio.gr/IR/Financial-Information/Financial-Results

9) sap wms: https://www.sap.com/greece/products/scm/extended-warehouse-management/technical-information.html

10) sap cloud : https://www.sap.com/greece/products/erp/sales.html

11) numpy : https://numpy.org/doc/stable/index.html

12) pandas: https://pandas.pydata.org/docs/getting_started/index.html

13) matplotlib: https://matplotlib.org/stable/users/index.html

14) seaborn: https://seaborn.pydata.org/tutorial.html

15) stats models: https://seaborn.pydata.org/tutorial.html

16) scikit learn: https://scikit-learn.org/stable/user_guide.html

17) tensorflow: https://www.tensorflow.org/

18) sap crm: https://www.sap.com/products/crm.html

19) Kaggle datasets : https://www.kaggle.com/datasets

20) Ιωάννης Κονταράτος σημειώσεις μαθήματος αστικά logistics

21) Σωκράτης Μοσχούρης σημειώσεις μαθήματος Διοίκηση Εφοδιαστικής Αλυσίδας

22) PAUL Newbold , William L. Carlson . Betty M. Thorne Στατιστική για τη διοίκηση και τα οικονομικά εκδόσεις κριτική

23) MANKIW N. GREGORY, TAYLOR P. MARK οικονομική εκδόσεις ΤΖΙΟΛΑ

24) Λάμπρος Λάιος Διοίκηση Εφοδιασμού, Εκδόσεις Humantec.

25) Makridakis S Wheelwright, SC & Hyndman, RJ 1998, Forecasting. Methods and applications, Εκδόσεις Chichester.

26) GREWAL DHRUV, LEVY MICHAEL ΜΑΡΚΕΤΙΝΓΚ εκδόσεις κριτική

27) Ευάγγελος Κεχρής Σχεσιακές βάσεις δεδομένων εκδόσεις Κριτική