ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

ΤΜΗΜΑ ΨΗΦΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ

ΠΛΗΡΟΦΟΡΙΑΚΑ ΣΥΣΤΗΜΑΤΑ ΚΑΙ ΥΠΗΡΕΣΙΕΣ

ΚΑΤΕΥΘΥΝΣΗ: ΠΡΟΗΓΜΕΝΑ ΠΛΗΡΟΦΟΡΙΑΚΑ ΣΥΣΤΗΜΑΤΑ

ΠΡΟΒΛΕΨΗ ΟΙΚΙΑΚΟΥ ΚΟΣΤΟΥΣ ΗΛΕΚΤΡΙΚΗΣ ΕΝΕΡΓΕΙΑΣ ΒΑΣΕΙ ΔΗΜΟΓΡΑΦΙΚΩΝ ΚΡΙΤΗΡΙΩΝ

ΜΕΤΑΠΤΥΧΙΑΚΗ ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Ιωάννης Γιαβρής

Μιχαήλ Φιλιππάκης, Καθηγητής, Επιβλέπων

Πειραιάς, Σεπτέμβριος 2022

ΕΥΧΑΡΙΣΤΙΕΣ

Θα ήθελα να ευχαριστήσω θερμά τον επιβλέποντα καθηγητή κ. Μιχαήλ Φιλιππάκη για την συνεργασία και την οικογένεια μου για την συμπαράσταση και την στήριξή της. Επίσης θα ήθελα να ευχαριστήσω τη Δρ. Μαρία Ελένη Πούλου για την πολύτιμη βοήθεια της στην ανάλυση των δεδομένων και στην επίβλεψη της διπλωματικής.

ΠΕΡΙΛΗΨΗ

Σκοπός αυτής της μεταπτυχιακής εργασίας είναι ο πειραματισμός με μια σειρά από αλγορίθμους μηχανικής μάθησης για την εύρεση ενός αξιόπιστου μοντέλου πρόβλεψης οικιακού κόστους ηλεκτρικής ενέργειας. Το μοντέλο, εκπαιδευμένο σε δημογραφικά χαρακτηριστικά όπως η οικογενειακή κατάσταση, θα δύναται να παρέχει χρήσιμες πληροφορίες για το μελλοντικό κόστος κατανάλωσης ηλεκτρικής ενέργειας από τα νοικοκυριά. Οι πληροφορίες αυτές θα μπορούσαν να αξιοποιηθούν για ευρύτερη κατανόηση των αλλαγών στο κόστος της ηλεκτρικής ενέργειας λόγω μεταβολών σε δημογραφικά χαρακτηριστικά, καθώς και για λήψη καλύτερων αποφάσεων σε περίπτωση πιθανής μελλοντικής επάρκειας η ανεπάρκειας στο αγαθό αυτό. Στα πλαίσια της εργασίας αυτής, γίνεται μεταξύ άλλων συνοπτική βιβλιογραφική ανασκόπηση στην εξόρυξη δεδομένων και στους επιλεχθέντες αλγόριθμους μηχανικής μάθησης, περιγραφική ανάλυση του συνόλου δεδομένων και παρουσίαση των αποτελεσμάτων μετά την εφαρμογή των αλγορίθμων. Η ανάλυση στηρίχθηκε σε ένα σύνολο δεδομένων που διατίθεται δωρεάν στο διαδίκτυο, με την χρήση της γλώσσας προγραμματισμού Python.

ABSTRACT

Purpose of this master thesis is the experimentation with a number of machine learning algorithms for the discovery of an accurate predictive model for the cost of the household electric power consumption. The model, trained on demographic characteristics such as the family status, could provide valuable information for the future cost of electricity consumption by households. This kind of information could be utilized for wider knowledge of changes on electricity consumption due to variations on demographic characteristics as well as for taking appropriate decisions on possible future surplus or deficit on this commodity. In this thesis, among others a short bibliographic review on data mining and on the selected machine learning algorithms takes place, a descriptive analysis of the dataset and presentation of the results of the applied algorithms. The analysis is based on a free and available dataset on the web, with the use of the Python programming language.

ΠΕΡΙΕΧΟΜΕΝΑ

ΛΙΣΤΑ ΕΙΚΟΝΩΝ

ΛΙΣΤΑ ΠΙΝΑΚΩΝ

ΛΙΣΤΑ ΓΡΑΦΗΜΑΤΩΝ

ΕΙΣΑΓΩΓΗ ΚΑΙ ΔΟΜΗ ΕΡΓΑΣΙΑΣ

Η βελτίωση του βιοτικού επιπέδου των πολιτών είναι μια από τις σημαντικότερες υποχρεώσεις που καλείται να εκπληρώσει μια οργανωμένη κοινωνία. Υλική πτυχή του βιοτικού επιπέδου δε θα μπορούσε να μην αποτελεί η απρόσκοπτη παροχή ηλεκτρικής ενέργειας μιας από τις πλέον λειτουργικές και χρήσιμες μορφές ενέργειας για μια σειρά από καθημερινές λειτουργίες ενός νοικοκυριού.

Συνάμα, είναι ενδιαφέρουσα η αξιολόγηση της απόδοσης μοντέλων πρόβλεψης κόστους οικιακής ηλεκτρικής ενέργειας πόσω μάλλον όταν οι προβλέψεις αυτές στηρίζονται σε διάφορα δημογραφικά χαρακτηριστικά, παρέχοντας μια εικόνα για το αντίκτυπο των δημογραφικών μεταβολών στο συγκεκριμένο αγαθό, με μελλοντικές προεκτάσεις την ενδεχόμενη συμβολή στην γνώση για σχετικό πλεόνασμα ή έλλειμμα μιας πόλης στο αγαθό αυτό, συνοψίζοντας και την προσπάθεια αυτής της εργασίας για συνεισφορά.

Η εργασία αποτελείται από 4 κεφάλαια, τα οποία και περιγράφονται συνοπτικά ακολούθως:

Στο 1ο κεφάλαιο γίνεται βιβλιογραφική ανασκόπηση στην διαδικασία εξόρυξης δεδομένων και σε βασικές μεθόδους που είναι διαθέσιμες

Στο 2ο κεφάλαιο γίνεται βιβλιογραφική ανασκόπηση σε συγκεκριμένους αλγορίθμους μηχανικής μάθησης που χρησιμοποιούνται στα πλαίσια της παρούσας ανάλυσης και σε άλλα θέματα της διαδικασίας ανάλυσης όπως ο διαχωρισμός δεδομένων

Στο 3ο κεφάλαιο γίνεται η προεπεξεργασία των δεδομένων, η περιγραφική ανάλυση τους και η παραμετροποίηση των επιλεχθέντων αλγορίθμων

Στο 4ο κεφάλαιο παρουσιάζονται τα αποτελέσματα, ο πιο ακριβής αλγόριθμος πρόβλεψης και συμπεράσματα/σκέψεις πάνω στην ανάλυση

Τέλος, ολοκληρώνεται η μεταπτυχιακή εργασία με αναφορά στην βιβλιογραφία και ένα παράρτημα ενδεικτικού κώδικα που χρησιμοποιήθηκε.

ΚΕΦΑΛΑΙΟ 1

1.1 Εξόρυξη Δεδομένων

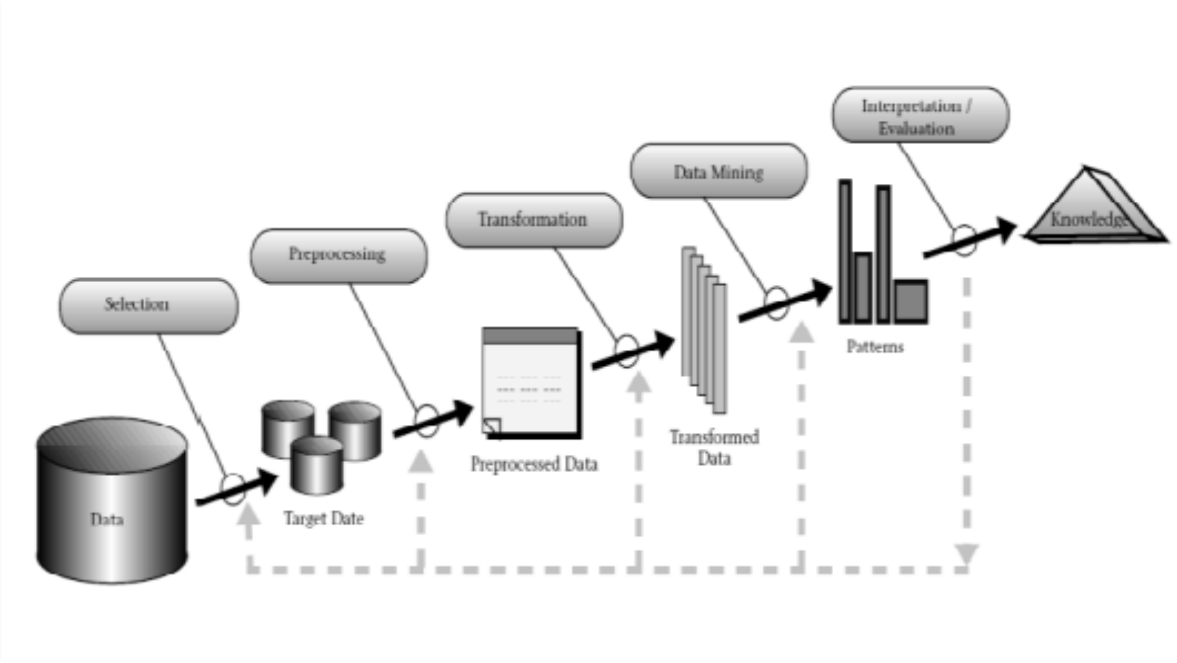
Η εξόρυξη δεδομένων (data mining) είναι μια διαδικασία που συνδυάζει παραδοσιακές μεθόδους ανάλυσης και πρόβλεψης με αλγορίθμους στοχεύοντας στην εξαγωγή χρήσιμων πληροφοριών από τα δεδομένα. Πληροφορίες όπως κρυμμένα μοτίβα, συσχετίσεις και προβλέψεις για μελλοντικές συμπεριφορές μπορούν να εξαχθούν και να χρησιμοποιηθούν από μια σειρά από οντότητες και οργανισμούς, όπως κράτη, εταιρείες και πανεπιστήμια, με μεγάλο εύρος εφαρμογής από τα χρηματοοικονομικά μέχρι την μετεωρολογία, με την συμβολή της εξόρυξης δεδομένων στην έρευνα, στην ανάλυση και στην λήψη αποφάσεων να μπορεί να χαρακτηριστεί ανεκτίμητη.

1.2 Διαδικασία Εξόρυξης Δεδομένων

Η ανακάλυψη γνώσης από τα δεδομένα είναι μια διαδικασία που αποτελείται από 5 φάσεις, όπως απεικονίζεται στην Εικόνα 1.1. Μέσα από αυτές τις φάσεις δημιουργείται το μοντέλο το οποίο περιγράφει συμπεριφορές ή προβλέπει μελλοντικές. Κάποιες φορές ένα περιγραφικό μοντέλο είναι απλά η εισαγωγή για περαιτέρω επεξεργασία και δημιουργία ενός μοντέλου πρόβλεψης. Οι 5 φάσεις της διαδικασίας εξόρυξης δεδομένων είναι οι εξής:

1. Συλλογή Δεδομένων

2. Προεπεξεργασία Δεδομένων
3. Μετασχηματισμός Δεδομένων
4. Εξόρυξη Δεδομένων
5. Διερμηνεία και Αξιολόγηση



Εικόνα 1.1, Διαδικασία Ανακάλυψης Γνώσης από τα Δεδομένα

Φάση 1η : Γίνεται η συλλογή των δεδομένων με διάφορους τρόπους όπως μέσα από μια βάση δεδομένων ή με την χρήση ερωτηματολογίων και άλλων μέσων. Σε αυτή τη φάση τα δεδομένα είναι πιθανό να περιέχουν ελλιπή δεδομένα, ακραίες τιμές και άλλα θέματα τα οποία θα αντιμετωπιστούν μετέπειτα.

Φάση 2η : Γίνεται η πρώτη επεξεργασία των δεδομένων με απώτερο στόχο την αύξηση της ποιότητας και της εγκυρότητας τους. Ορθά δεδομένα θα δώσουν αξιόπιστα και εγκυρότερα μοντέλα πρόβλεψης, αποφεύγοντας την παρουσίαση υπερπροσαρμογής (overfitting) σε αυτά. Σε αυτή την φάση καταναλώνεται μεγάλο μέρος της προσπάθειας ενός αναλυτή αντιμετωπίζοντας εσφαλμένες και άλλες τιμές.

Φάση 3η : Είναι η στιγμή στην οποία ο μετασχηματισμός των δεδομένων λαμβάνει χώρα και περιλαμβάνει ενέργειες όπως την δημιουργία δομών που εξυπηρετούν τον τελικό στόχο. Τεχνικές όπως η κανονικοποίηση (ένταξη δεδομένων σε συγκεκριμένο εύρος τιμών) ή η συσταδοποίηση (δημιουργία ομάδων με κοινά χαρακτηριστικά) μπορούν να αξιοποιηθούν σε αυτή την φάση.

Φάση 4η : Έχοντας επεξεργαστεί και μετασχηματίσει τα δεδομένα, πραγματοποιείται η εφαρμογή και εκπαίδευση των επιλεχθέντων αλγορίθμων με στόχο την προσπάθεια παραγωγής του μοντέλου που θα περιγράφει τα δεδομένα ή θα προβλέπει μελλοντικές τιμές, ανάλογα με τον στόχο που έχει τεθεί.

Φάση 5η : Οι φάσεις της εξόρυξης δεδομένων ολοκληρώνονται με την αξιολόγηση και ερμηνεία των αποτελεσμάτων. Η διαδικασία μπορεί να επαναληφθεί εξαρχής αλλά και να τεθούν ερωτήματα και σκέψεις για βελτίωση και μελλοντική έρευνα.

1.3 Μέθοδοι Εξόρυξης Δεδομένων

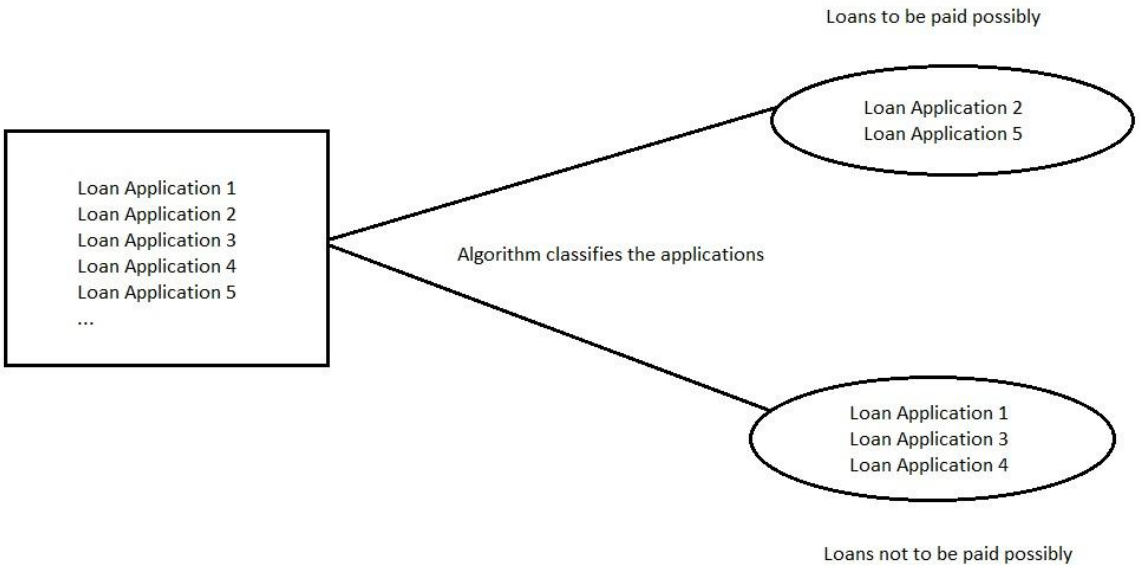
Μια σειρά από μεθόδους είναι διαθέσιμες για την εξόρυξη δεδομένων και η επιλογή επαφίεται στον τελικό στόχο. Σε γενικές γραμμές, οι μέθοδοι ανήκουν σε δύο μεγάλες κατηγορίες: προγνωστικές και περιγραφικές. Στις προγνωστικές μεθόδους, το μοντέλο εκπαιδεύεται για να προβλέπει την τιμή μιας μεταβλητής. Αντίθετα, στις περιγραφικές μεθόδους, το μοντέλο χρησιμοποιείται για ανάλυση και περιγραφή των δεδομένων με στόχο την εύρεση συσχετίσεων και μοτίβων που εξυπηρετούν στην καλύτερη κατανόηση τους.

Περαιτέρω, και προτού γίνει συνοπτική ανάλυση συγκεκριμένων μεθόδων εξόρυξης δεδομένων, 2 κύριες κατηγορίες εκπαίδευσης θα πρέπει να σημειωθούν: η επιβλεπόμενη μάθηση (supervised learning) και η μη επιβλεπόμενη μάθηση (non supervised learning). Η βασική διαφορά έγκειται στο γεγονός ότι η επιβλεπόμενη μάθηση απαιτεί συγκεκριμένες κατευθύνσεις πάνω στις οποίες θα εκπαιδευτεί το μοντέλο, παράδειγμα ότι μια τιμή θα

αντιστοιχεί σε συγκεκριμένη κλάση. Αντίθετα, στην μη επιβλεπόμενη μάθηση το μοντέλο προσπαθεί χωρίς καμία εξωτερική κατεύθυνση να ανακαλύψει δομές και κρυμμένα μοτίβα.

1.3.1 Κατηγοριοποίηση

Η κατηγοριοποίηση (classification) είναι μια προγνωστική μέθοδος, επιβλεπόμενης μάθησης, η οποία αντιστοιχεί δεδομένα σε προκαθορισμένες κλάσεις. Το μοντέλο που επιλέγεται εκπαιδεύεται στα δεδομένα και βάσει των προκαθορισμένων κατηγοριών που έχουν τεθεί μαθαίνει να αντιστοιχεί κάθε δεδομένο στην αντίστοιχη κατηγορία. Στην Εικόνα 1.2, παράδειγμα χρήσης αυτής της μεθόδου παρουσιάζεται όπου έχοντας εκπαιδεύσει το μοντέλο σε δεδομένα αιτήσεων δανείου, το μοντέλο μπορεί πια να κατηγοριοποιήσει τις αιτήσεις δανείων σε εκείνες που θα εξυπηρετηθούν ή θα μείνουν ανεκπλήρωτες.



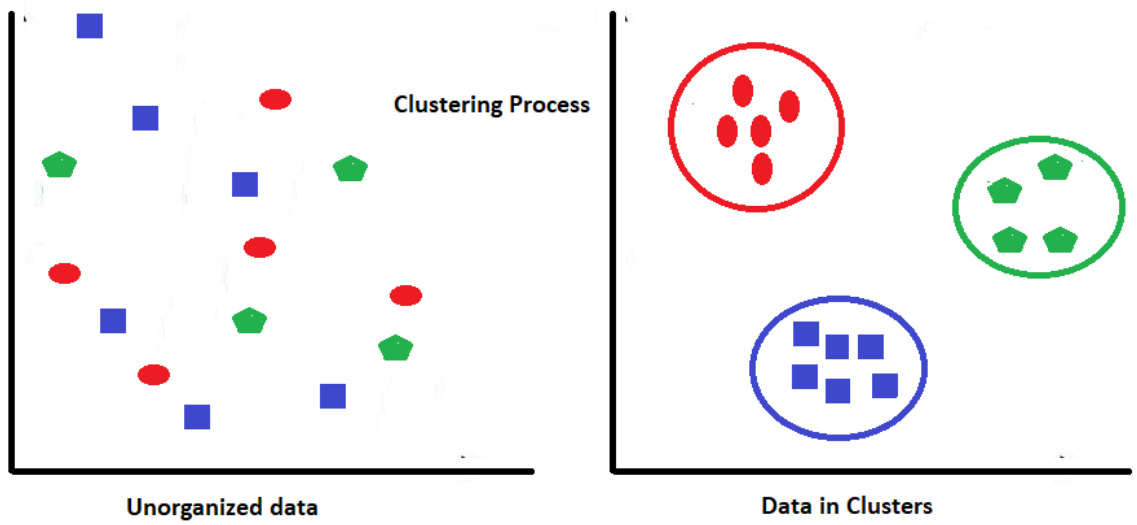
Εικόνα 1.2, Κατηγοριοποίηση Αιτήσεων Δανείου

1.3.2 Παλινδρόμηση

Η παλινδρόμηση (regression) είναι μια κλασική μέθοδος πρόγνωσης, επιβλεπόμενης μάθησης, η οποία κατανοεί την σχέση μεταξύ της εξαρτημένης μεταβλητής και των ανεξάρτητων μεταβλητών και δημιουργεί μια συνάρτηση που αντικατοπτρίζει την σχέση αυτή με την μικρότερη δυνατή απόκλιση. Ως επιβλεπόμενης μάθησης μέθοδος, είναι σημαντικό να δοθούν οι σωστές κατευθύνσεις κατά την διάρκεια της εκπαίδευσης του μοντέλου ώστε η πρόβλεψη να είναι όσο τον δυνατόν πιο ακριβής. Μια σειρά από τεχνικές δύναται να χρησιμοποιηθούν, όπως η γραμμική παλινδρόμηση, η λογιστική παλινδρόμηση ή η παλινδρόμηση με την χρήση διανυσμάτων υποστήριξης.

1.3.3 Συσταδοποίηση

Η συσταδοποίηση (clustering) θεωρείται μια περιγραφική μέθοδος, μη επιβλεπόμενης μάθησης, με κύριο στόχο την ανακάλυψη ομάδων δεδομένων που έχουν κοινά χαρακτηριστικά, εξυπηρετώντας στην καλύτερη κατανόηση και αξιοποίηση τους. Η διαδικασία αυτή έχει ως αποτέλεσμα την δημιουργία συστάδων - με τον αριθμό τους να μην είναι γνωστός εκ των προτέρων - που δίνουν περιγραφικές γνώσεις στον αναλυτή και πιθανόν μια βάση για περαιτέρω επεξεργασία από ένα μοντέλο πρόβλεψης. Στην Εικόνα 1.3, παρουσιάζεται ένα απλό παράδειγμα αυτής της μεθόδου στο οποίο γίνεται αντιληπτό η χρησιμότητα της μέσω της ανακάλυψης ομοιοτήτων και δημιουργίας ομοιογένειας, μετατρέποντας ένα ασύνδετο σύνολο σε 3 ομοιογενείς ομάδες που μοιράζονται κοινές ιδιότητες.



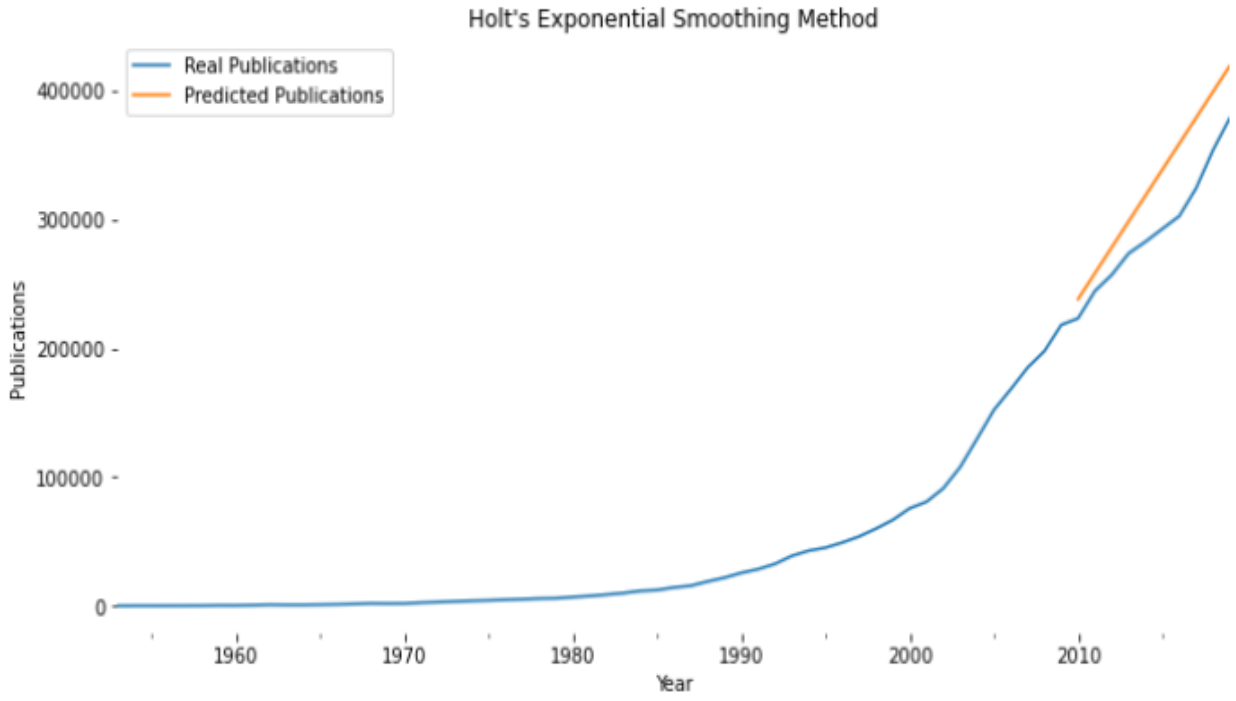
Εικόνα 1.3, Συσταδοποίηση Δεδομένων

1.3.4 Ανάλυση Συσχέτισης

Η ανάλυση συσχέτισης (association analysis), ως περιγραφική μέθοδος μη επιβλεπόμενης μάθησης, στοχεύει στην εύρεση συσχετίσεων και κανόνων σε ένα σύνολο δεδομένων, ανακαλύπτοντας την εξάρτηση κάποιων μεταβλητών με κάποιες άλλες. Θεωρείται μια ιδιαίτερα ενδιαφέρουσα πρακτική ανάλυσης με εφαρμογή σε μια σειρά από τομείς καθώς μπορεί να ανακαλύψει άγνωστα κρυμμένα μοτίβα, παράδειγμα ότι μια συγκεκριμένη αγορά προϊόντος συνδυάζεται συχνά με μια άλλη από τους καταναλωτές συμβάλλοντας με αυτόν τον τρόπο κομβικά στην λήψη ποιοτικότερων αποφάσεων. Αυτοί οι κανόνες που δημιουργούνται από τα σχετικά μοντέλα, όπως ο Apriori, είναι γνωστοί και ως κανόνες σχετικότητας (association rules).

1.3.5 Ανάλυση Χρονοσειράς

Η ανάλυση χρονοσειράς (time series analysis) είναι κατά βάση μία μέθοδος πρόβλεψης, επιβλεπόμενης μάθησης, και χρησιμοποιείται για την πρόγνωση μιας τιμής στον χρόνο. Η ανάλυση πραγματοποιείται πάνω σε δεδομένα που έχουν καταγραφεί σε σαφή χρονικά διαστήματα και όχι σε τυχαίες χρονικές στιγμές. Η μέθοδος αυτή εξυπηρετεί ακόμη και στην κατανόηση διαφόρων χαρακτηριστικών, όπως η τάση και η εποχικότητα. Μια σειρά από μοντέλα είναι διαθέσιμα για την ανάλυση μιας χρονοσειράς, όπως του Holt η οποία απεικονίζεται στην Εικόνα 1.4 και αφορά εκδόσεις δημοσιεύσεων στον χώρο της επιστήμης των υπολογιστών.



Εικόνα 1.4, Ανάλυση Χρονοσειράς σε Επιστημονικές Δημοσιεύσεις με την μέθοδο Holt

ΚΕΦΑΛΑΙΟ 2

2.1 Μηχανική Μάθηση και Αλγόριθμοι

Η μηχανική μάθηση (machine learning) είναι μια διαδικασία που προέρχεται από τον κλάδο της τεχνητής νοημοσύνης και εκπαιδεύει αλγόριθμους και μοντέλα σε δεδομένα. Μέσω της μηχανικής μάθησης τα μοντέλα αποκτούν την ικανότητα να αποδίδουν σε νέα δεδομένα και καινούργιες καταστάσεις παρέχοντας περιγραφικές και προγνωστικές γνώσεις. Η ανάλυση δεν χρειάζεται να στηρίζεται αποκλειστικά στον αναλυτή πλέον αλλά στα ίδια τα μοντέλα και στην δυνατότητα εκμάθησης τους. Στα πλαίσια αυτής της εργασίας θα γίνει αναφορά σε συγκεκριμένους αλγορίθμους, όπως τον Ridge που ανήκει στους αλγόριθμους (γραμμικής) παλινδρόμησης, τις μηχανές υποστήριξης διανυσμάτων (support vector machines) και τον Random Forest, οι οποίοι χρησιμοποιούνται στο πρακτικό μέρος της ανάλυσης.

2.1.1 Αλγόριθμος Ridge

Η παλινδρόμηση όπως έχει ήδη αναφερθεί είναι μια κλασική τεχνική στην προσπάθεια να βρεθεί η σχέση μεταξύ μιας ή περισσοτέρων ανεξάρτητων μεταβλητών με την εξαρτημένη μεταβλητή. Στην μηχανική μάθηση η μέθοδος της παλινδρόμησης χρησιμοποιείται ως τεχνική πρόβλεψης με τα μοντέλα να εκπαιδεύονται στο να προβλέπουν μελλοντικές τιμές με την χρήση διαφόρων τεχνικών παλινδρόμησης, όπως της γραμμικής. Ο αλγόριθμος Ridge είναι μια βελτιωμένη έκδοση της γραμμικής παλινδρόμησης με κύριο χαρακτηριστικό ότι ο

τιμωρεί το μοντέλο. Ενώ στην γραμμική παλινδρόμηση το βάρος μιας παραμέτρου μπορεί να αυξηθεί αρκετά εάν το μοντέλο κρίνει ότι έχει μεγάλη επίδραση, ο αλγόριθμος Ridge τιμωρεί την επίδραση αυτή παρέχοντας μεγαλύτερη ισορροπία στην επίδραση των ανεξάρτητων μεταβλητών και αποφεύγοντας την υπερπροσαρμογή στα δεδομένα.

Πλεονεκτήματα - Μειονεκτήματα

Στα πλεονεκτήματα του αλγόριθμου Ridge, και γενικότερα των αλγορίθμων παλινδρόμησης, είναι η ευκολία στην χρήση και στην κατανόηση. Όπως έχει ήδη αναφερθεί, στα θετικά σημειώνεται ότι ο αλγόριθμος Ridge αποδίδει καλά και αποφεύγει την υπερπροσαρμογή που οδηγεί σε εσφαλμένα αποτελέσματα. Στα μειονεκτήματα μπορεί να γίνει αναφορά στην αδυναμία διαχείρισης δεδομένων που δεν έχουν επεξεργαστεί ορθά για εσφαλμένες και χαμένες τιμές, ένα χαρακτηριστικό που αφορά τους περισσότερους αλγόριθμους παλινδρόμησης. Στα αρνητικά πρέπει να σημειωθεί ακόμη η ανάγκη να δοθεί η κατάλληλη τιμή στην παράμετρο alpha που αφορά την τιμωρία του μοντέλου ώστε να αποδώσει τα καλύτερα αποτελέσματα.

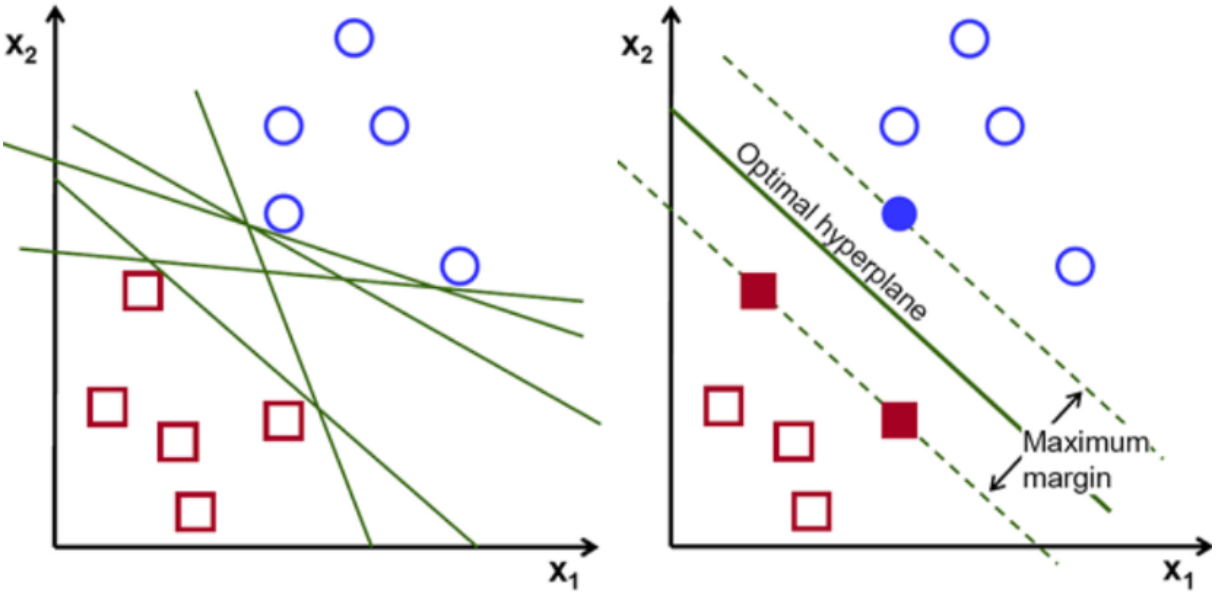
2.1.2 Μηχανές Υποστήριξης Διανυσμάτων

Ο αλγόριθμος SVM, επιβλεπόμενης μάθησης, είναι δημοφιλής για λύσεις τόσο σε θέματα παλινδρόμησης όσο και κατηγοριοποίησης. Η μέθοδος αυτή αφορά την δημιουργία μιας διαχωριστικής γραμμής ή υπερεπέδου (hyperplane) που διαχωρίζει τα δεδομένα με βέλτιστο τρόπο, όπως απεικονίζεται στην Εικόνα 2.1. Αρκετές διαχωριστικές γραμμές μπορεί να προκύπτουν ωστόσο ο αλγόριθμος SVM διατηρεί εκείνη με το μεγαλύτερο περιθώριο.

Πλεονεκτήματα - Μειονεκτήματα

Στα πλεονεκτήματα της μεθόδου αυτής είναι η δυνατότητα να λύνει πολύπλοκα θέματα και η μειωμένη τάση για υπερπροσαρμογή στα δεδομένα. Στα μειονεκτήματα μπορεί να σημειωθεί

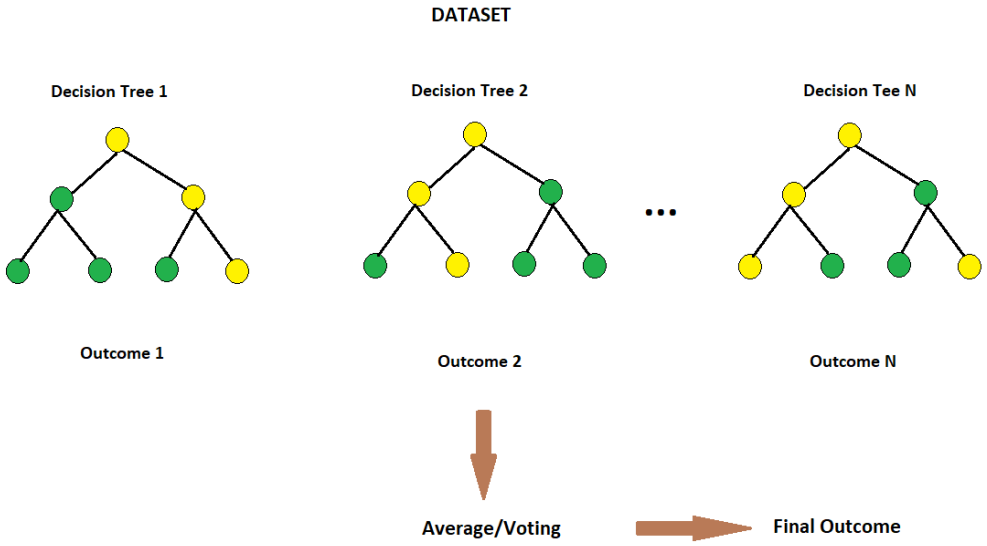
ότι ο αλγόριθμος δεν θεωρείται κατάλληλος για πολύ μεγάλα σύνολα δεδομένων και ότι δεν αποδίδει καλά όταν υπάρχει θόρυβος στα δεδομένα.



Εικόνα 2.1, Αλγόριθμος Μηχανής Υποστήριξης Διανυσμάτων

2.1.3 Αλγόριθμος Random Forest

Ο αλγόριθμος Random Forest, επιβλεπόμενης μάθησης, θεωρείται εξαιρετικά δημοφιλής τόσο για αναλύσεις κατηγοριοποίησης όσο και παλινδρόμησης. Η μέθοδος αυτή αφορά την δημιουργία πολλών δέντρων αποφάσεων (decision trees) και βασισμένη στις επιμέρους προβλέψεις καταλήγει στα τελικά αποτελέσματα, όπως παρουσιάζεται στην Εικόνα 2.2. Παρόλο που κάποια δέντρα απόφασης είναι πιθανό να μην οδηγήσουν σε έγκυρες προβλέψεις, θεωρείται ότι η πλειοψηφία των δέντρων θα καταλήξει σε σωστές προβλέψεις και έτσι η μέθοδος μέσω της σύνοψης θα οδηγήσει σε βέλτιστα αποτελέσματα. Για την εφαρμογή του συγκεκριμένου αλγόριθμου, θα πρέπει να πληρούνται κάποια χαρακτηριστικά, με σημαντικότερη την χαμηλή συσχέτιση μεταξύ των δέντρων αποφάσεων.



Εικόνα 2.2, Αλγόριθμος Random Forest

Πλεονεκτήματα - Μειονεκτήματα

Στα πλεονεκτήματα της μεθόδου αυτής είναι η υψηλή απόδοση στις προβλέψεις λόγω της πολυπλοκότητας και η μειωμένη υπερπροσαρμογή στα δεδομένα που οδηγεί σε εσφαλμένα αποτελέσματα. Υπερπροσαρμογή μπορεί να εμφανιστεί σε μεμονωμένα δέντρα απόφασης, αλλά είναι πιο δύσκολο να εμφανιστεί όταν δεκάδες δέντρα έχουν δημιουργηθεί με το τελικό αποτέλεσμα να είναι προϊόν συνδυασμένων προβλέψεων. Στα θετικά μπορεί να σημειωθεί επίσης η δυνατότητα αντιμετώπισης χαμένων τιμών και η εφαρμογή σε αναλύσεις κατηγοριοποίησης και παλινδρόμησης. Ως μειονέκτημα της μεθόδου είναι η αργή απόδοση του αλγορίθμου λόγω της πολυπλοκότητας που εμπεριέχει και των υπολογισμών που πρέπει να εκτελεστούν.

2.2 Επεξεργασία Δεδομένων

Η επεξεργασία των δεδομένων είναι αρκετές φορές μια χρονοβόρα διαδικασία καθώς μια σειρά από τιμές του συνόλου δεδομένων πρέπει να αντιμετωπιστούν ώστε να θεωρούνται

ορθές και έγκυρες και να μπορούν να χρησιμοποιηθούν από τα μοντέλα σε μεταγενέστερο στάδιο παρέχοντας τις βάσεις για πιο αξιόπιστη απόδοση.

Χαμένες Τιμές

Σημαντικό μέρος της επεξεργασίας δεδομένων είναι η αντιμετώπιση των χαμένων τιμών η οποία παρατηρείται σε πολλά σύνολα δεδομένων και δεν μπορεί να διαχειριστεί από αρκετά μοντέλα. Η απουσία των δεδομένων έγκειται σε διάφορες αιτίες όπως το ανθρώπινο λάθος, την κακή συντήρηση ή την μη καταγραφή εξαρχής. Για την επεξεργασία αυτών των δεδομένων, υπάρχει η δυνατότητα διαγραφής όλης της εγγραφής που εμπεριέχει την χαμένη τιμή, ωστόσο η αντικατάσταση της με μια από τις ακόλουθες επιλογές θεωρείται πιο ορθή:

- Αντικατάσταση με τον μέσο όρο, την επικρατούσα τιμή ή την διάμεσο
- Αντικατάσταση με μια τυχαία τιμή από το εύρος της μεταβλητής
- Αντικατάσταση με μια τιμή υπολογισμένη από άλλο μοντέλο

Ακραίες τιμές

Οι ακραίες τιμές είναι παρατηρήσεις που βρίσκονται μακριά από τα υπόλοιπα δεδομένα. Μια ακραία τιμή δεν σημαίνει απαραίτητα ότι είναι εσφαλμένη και θεωρείται σημαντικό να αναλύεται προτού αφαιρεθεί ή αντικατασταθεί. Παρόλα αυτά καλό είναι οι ακραίες τιμές να αντιμετωπίζονται καθώς μπορούν επηρεάσουν την απόδοση των μοντέλων. Η αναγνώριση τους μπορεί να γίνει εύκολα μέσω της χρήσης γραφημάτων, όπως τα θηκογράμματα (boxplots).

Άλλες τιμές

Εκτός από τις προαναφερθείσες περιπτώσεις, υπάρχουν και άλλες τιμές που χρήζουν αντιμετώπισης για να θεωρηθούν τα δεδομένα έτοιμα για χρήση από τα μοντέλα, παράδειγμα οι μη λογικές τιμές και οι διπλοεγγραφές. Με την ολοκλήρωση της επεξεργασίας, μπορεί να

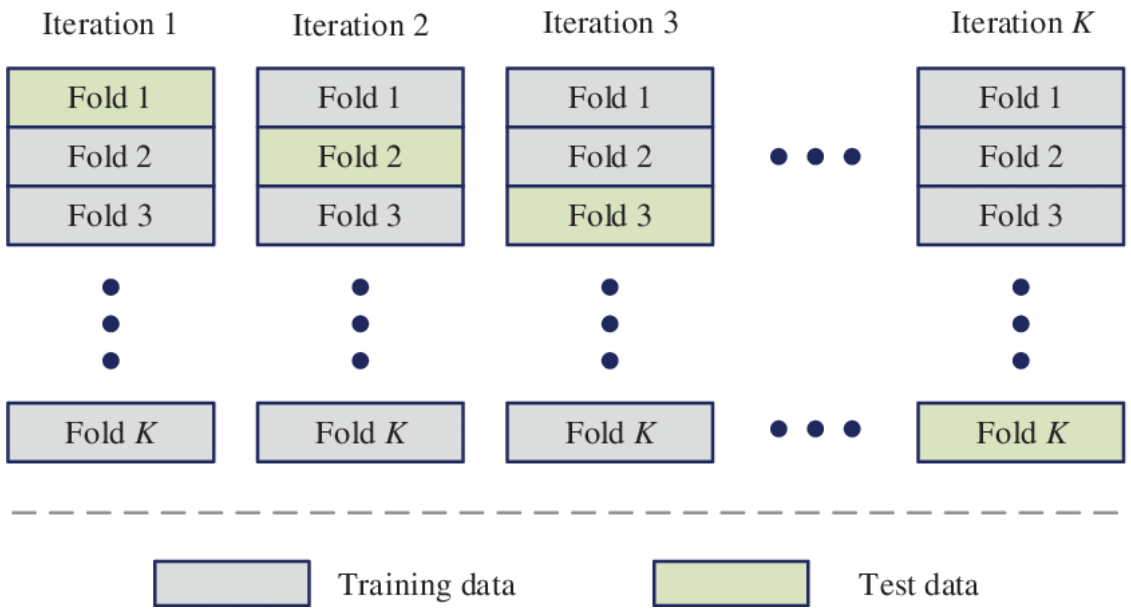
ακολουθήσει και ενδεχόμενος μετασχηματισμός των δεδομένων, παράδειγμα η κανονικοποίηση τους, ανάλογα με τον στόχο που έχει τεθεί.

2.3 Διαχωρισμός Δεδομένων

Για να πραγματοποιηθεί η εφαρμογή των μοντέλων απαιτείται αρχικά ο διαχωρισμός των δεδομένων σε δύο μέρη, ένα μέρος αφορά τα δεδομένα εκπαίδευσης (training set) και ένας μέρος τα δεδομένα που χρησιμοποιούνται για έλεγχο και αξιολόγηση (test set). Τα δεδομένα εκπαίδευσης συνήθως περιέχουν από 70% έως 80% του συνόλου ενώ το υπόλοιπο περιέχει από 20% εώς 30%, συνοψίζοντας τον πιο κλασικό τρόπο διαχωρισμού πριν την εφαρμογή οποιουδήποτε μοντέλου. Τα δεδομένα εκπαίδευσης χρησιμοποιούνται για να εκπαιδεύσουν το μοντέλο να κάνει προβλέψεις και τα υπόλοιπα δεδομένα χρησιμοποιούνται για να μετρηθεί κατά πόσο το μοντέλο εκπαιδεύτηκε σωστά και μπορεί να κάνει αξιόπιστες προβλέψεις. Θα πρέπει να σημειωθεί ότι αυτός ο διαχωρισμός θεωρείται συνήθως καταλληλότερος για μεγάλα σύνολα δεδομένων.

Άλλοι Μέθοδοι

Η μέθοδος K – Fold Cross Validation είναι μια τεχνική που χωρίζει το σύνολο δεδομένων σε k υποσύνολα, μια τιμή που καθορίζεται πριν την εκτέλεση. Αυτά τα υποσύνολα έχουν κοινό μέγεθος και κάθε ένα υποσύνολο χρησιμοποιείται από μια φορά ως δεδομένο ελέγχου και αξιολόγησης, με τα υπόλοιπα σε ρόλο εκπαίδευσης. Το μοντέλο σε κάθε επανάληψη αξιολογείται και εμφανίζει την τελική αξιολόγηση όταν όλα τα υποσύνολα έχουν χρησιμοποιηθεί ως δεδομένα ελέγχου. Σε αντίθεση με τον κλασικό τρόπο διαχωρισμού, αυτή η τεχνική θεωρείται καταλληλότερη για μικρά σύνολα δεδομένων και έχει χρησιμοποιηθεί από την παρούσα ανάλυση με τα αποτελέσματα να παρουσιάζονται σε επόμενο κεφάλαιο. Μια σύνοψη της τεχνικής παρουσιάζεται στην Εικόνα 2.3.



Εικόνα 2.3, Εφαρμογή τεχνικής K - Fold Cross Validation

2.4 Μέτρα Αξιολόγησης

Για την αξιολόγηση ενός μοντέλου πρόβλεψης μια σειρά από δείκτες αξιολόγησης είναι διαθέσιμοι με στόχο να αποτυπώσουν το σφάλμα πρόβλεψης ανάμεσα στις προβλεπόμενες τιμές και στις πραγματικές. Οι βασικότεροι δείκτες αξιολόγησης είναι οι ακόλουθοι:

- Μέσο Τετραγωνικό Σφάλμα (MSE)
- Ρίζα Μέσου Τετραγωνικού Σφάλματος (RMSE)
- Μέσο Απόλυτο Σφάλμα (MAE)

Ο δείκτης MSE υπολογίζει την διαφορά μεταξύ πραγματικής και προβλεπόμενης τιμής και την υψώνει στο τετράγωνο. Θεωρείται σημαντικός δείκτης αξιολόγησης ωστόσο λόγω του υπολογισμού στο τετράγωνο έχει την τάση να διογκώνει τα αποτελέσματα σε μοντέλα με

μεγάλα σφάλματα. Με μικρή διαφοροποίηση, ο δείκτης RMSE υπολογίζει την ρίζα της υψωμένης στο τετράγωνο διαφοράς μεταξύ πραγματικής και προβλεπόμενης τιμής. Είναι επίσης ένας σημαντικός δείκτης αξιολόγησης με το πλεονέκτημα να διατηρεί τα αποτελέσματα στην ίδια κλίμακα με τα δεδομένα διευκολύνοντας την κατανόηση και την αξιολόγηση. Ολοκληρώνοντας με τον δείκτη MAE, ο δείκτης αυτός υπολογίζει την απόλυτη τιμή της διαφοράς μεταξύ πραγματικής και προβλεπόμενης τιμής και με αυτό τον τρόπο δεν χάνονται σφάλματα με αρνητικές τιμές. Διατηρεί επίσης τα αποτελέσματα στην ίδια κλίμακα με τα δεδομένα, όπως ο RMSE, εξυπηρετώντας την κατανόηση και την ερμηνεία.

2.5 Περιβάλλον Εφαρμογής και Ανάλυσης

Για την υλοποίηση αυτής της ανάλυσης χρησιμοποιήθηκε η γλώσσα προγραμματισμού Python. Η Python είναι μια αντικειμενοστραφής γλώσσα προγραμματισμού η οποία δημιουργήθηκε από τον ολλανδικής καταγωγής προγραμματιστή Guido van Rossum και κυκλοφόρησε το 1991. Η Python αναπτύσσεται ως ελεύθερο λογισμικό και θεωρείται μια από τις δημοφιλέστερες και ταχεία αναπτυσσόμενες γλώσσες προγραμματισμού σε παγκόσμια κλίμακα. Το εργαλείο προγραμματισμού που χρησιμοποιήθηκε είναι το Jupyter Notebook, μια εφαρμογή εύκολη στην χρήση και ιδανική για αναλύσεις και παρουσιάσεις.

Όσον αφορά τις βιβλιοθήκης που χρησιμοποιήθηκαν για την ανάλυση, οι βασικότερες παρουσιάζονται ακολούθως:

1. Pandas: Η βιβλιοθήκη αυτή παρέχει την δυνατότητα καλού χειρισμού και προετοιμασίας των δεδομένων ώστε να αποκτήσουν την κατάλληλη μορφή για την μετέπειτα χρήση τους στα επιλεγμένα μοντέλα, αποφεύγοντας εσφαλμένες τιμές που μπορούν να επηρεάσουν την απόδοση των αλγορίθμων.

2. Numpy: Η συγκεκριμένη βιβλιοθήκη του οικοσυστήματος της Python αφορά τους αριθμητικούς υπολογισμούς που λαμβάνουν χώρα και παρέχει το κατάλληλο έδαφος για την ανάπτυξη και την λειτουργία άλλων βιβλιοθηκών, όπως της Scikit-learn, μέσω πολυδιάστατων πινάκων και μαθηματικών συναρτήσεων.

3. Scikit-Learn: Η βιβλιοθήκη αυτή είναι κομβική για την ανάλυση στο περιβάλλον της Python αφού παρέχει τους αλγόριθμους μηχανικής μάθησης που χρησιμοποιούνται κατά την διάρκεια της εξόρυξης δεδομένων.

4. Matplotlib: Η βιβλιοθήκη αυτή θεωρείται ευκολόχρηστη και παρέχει στους χρήστες μια σειρά από δυνατότητες σχεδιασμού γραφημάτων στην Python, όπως θηκογράμματα, ιστογράμματα (histograms) και άλλα.

ΚΕΦΑΛΑΙΟ 3

3.1 Επεξεργασία και Περιγραφή Συνόλου Δεδομένων

Το σύνολο δεδομένων που χρησιμοποιήθηκε στα πλαίσια αυτής της διπλωματικής εργασίας βρέθηκε από την ιστοσελίδα της Kaggle. Τα δεδομένα περιέχουν 1000 εγγραφες, αποτελούνται από 10 παραμέτρους και αφορούν την μηνιαία πληρωμή ηλεκτρικού ρεύματος από νοικοκυριά. Οι μεταβλητές παρουσιάζονται στον Πίνακα 3.1.

ΜΕΤΑΒΛΗΤΕΣ	ΕΠΕΞΗΓΗΣΗ
num_rooms	Αριθμός Δωματίων
num_people	Αριθμός Ανθρώπων
house_area	Επιφάνεια Κατοικίας
is_ac	Εξοπλισμός με Air Condition?
is_tv	Εξοπλισμός με Τηλεόραση?
is_flat	Είναι Διαμέρισμα?
ave_monthly_income	Μηνιαίο Εισόδημα
num_children	Αριθμός παιδιών
is_urban	Βρίσκεται Εντός Πόλεως?
amount_paid	Μηνιαίος Λογαριασμός

Πίνακας 3.1, Παρουσίαση Μεταβλητών

3.1.1 Επεξεργασία Δεδομένων

Σε αυτό τα στάδιο έγινε η επεξεργασία των δεδομένων προτού αυτά χρησιμοποιηθούν για τα μοντέλα πρόβλεψης και την εύρεση του πιο αξιόπιστου μοντέλου. Η επεξεργασία αφορά

κυρίως την αντιμετώπιση των δεδομένων από τυχόν χαμένες, ακραίες ή γενικότερα εσφαλμένες τιμές ώστε να αποφευχθούν παραπλανητικές προβλέψεις σε μεταγενέστερο στάδιο.

Στον Πίνακα 3.2, παρουσιάζονται βασικά χαρακτηριστικά των δεδομένων, όπως η ελάχιστη/μέγιστη τιμή, ο μέσος όρος και η τυπική απόκλιση. Γίνεται αντιληπτό με μια πρώτη επαφή η ύπαρξη αρνητικών τιμών για κάποιες μεταβλητές, όπως τον αριθμό δωματίων και τον αριθμό ανθρώπων της κατοικίας, καθώς και για το μηνιαίο εισόδημα, οι οποίες πρέπει να αντικατασταθούν ως εσφαλμένες.

ΜΕΤΑΒΛΗΤΕΣ	**ΕΛΑΧΙΣΤΗ/ ΜΕΓΙΣΤΗ ΤΙΜΗ**	**ΜΕΣΟΣ ΟΡΟΣ**	**ΤΥΠΙΚΗ ΑΠΟΚΛΙΣΗ**
num_rooms	-1/5	2.0	1.0
num_people	-1/11	5.0	2.0
house_area	244/1189	795	145
is_ac	0/1	0.4	0.5
is_tv	0/1	0.8	0.4
is_flat	0/1	0.5	0.5
ave_monthly_income	-1576/56531	24685	9678
num_children	0/4	1.0	0.9
is_urban	0/1	0.6	0.5
amount_paid	88/1103	600	181

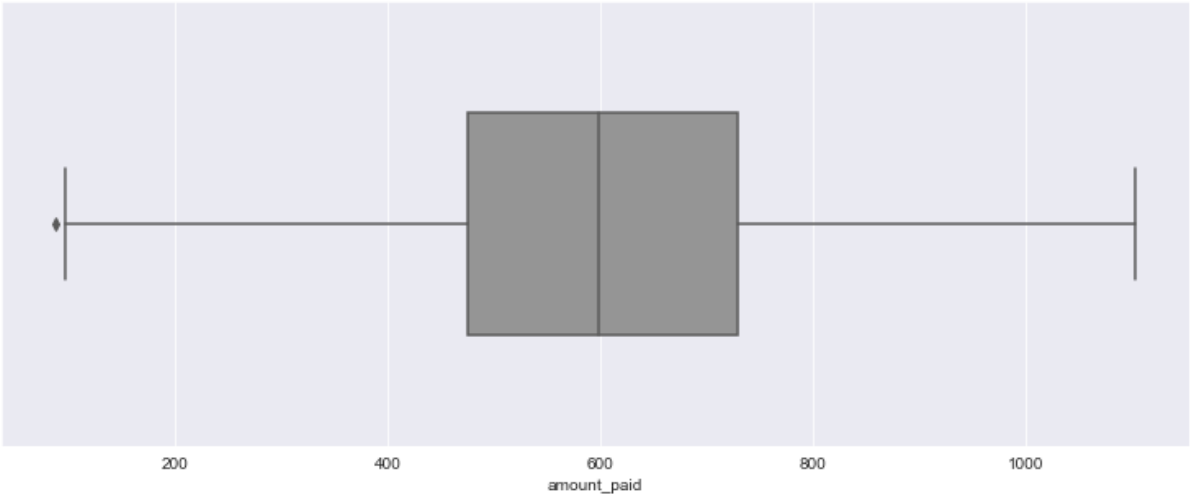
Πίνακας 3.2, Βασικά Χαρακτηριστικά Παραμέτρων

Για την περαιτέρω επεξεργασία των δεδομένων αυτών, επιλέχθηκε να γίνει η αντικατάσταση τους με την χρήση του μέσου όρου. Βάσει της μεθόδου αυτής οι ζητούμενες τιμές αντικαθίστανται από το μέσο όρο των τιμών της συγκεκριμένης μεταβλητής, με τον Πίνακα 3.3 να παρουσιάζει τις νέες τιμές.

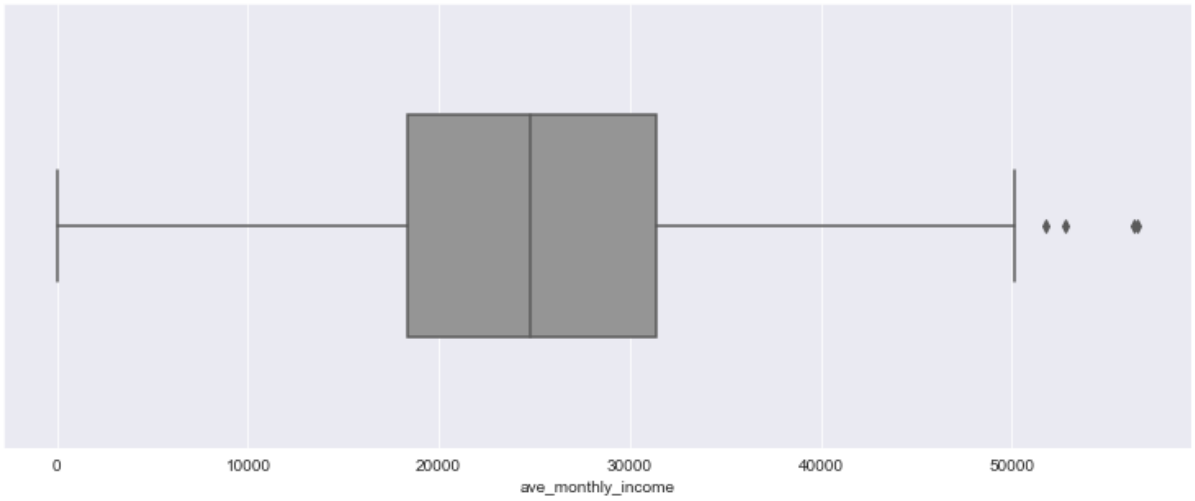
ΜΕΤΑΒΛΗΤΕΣ	ΕΛΑΧΙΣΤΗ/ ΜΕΓΙΣΤΗ ΤΙΜΗ	ΜΕΣΟΣ ΟΡΟΣ	ΤΥΠΙΚΗ ΑΠΟΚΛΙΣΗ
num_rooms	**1/5**	**2.0**	**1.0**
num_people	**1/11**	**5.0**	**2.0**
house_area	244/1189	795	145
is_ac	0/1	0.4	0.5
is_tv	0/1	0.8	0.4
is_flat	0/1	0.5	0.5
ave_monthly_income	**38/56531**	**24839**	**9472**
num_children	0/4	1.0	0.9
is_urban	0/1	0.6	0.5
amount_paid	88/1103	600	181

Πίνακας 3.3, Βασικά Χαρακτηριστικά Παραμέτρων μετά την χρήση του Μέσου Όρου

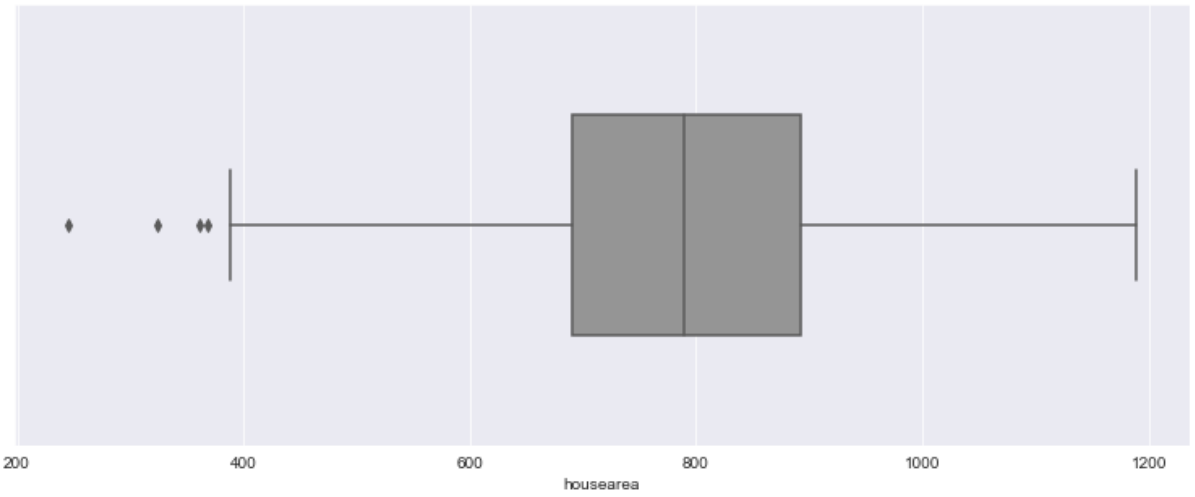
Στα δεδομένα έγινε ακόμη έλεγχος για χαμένες ή ακραίες τιμές, και για διπλές εγγραφές δεδομένων. Εντοπίστηκαν μόνο ακραίες τιμές, όπως απεικονίζονται στα Γραφήματα 3.1, 3.2 και 3.3, και αντικαταστάθηκαν με την χρήση του μέσου όρου.



Γράφημα 3.1 Θηκόγραμμα Μηνιαίου Λογαριασμού



Γράφημα 3.2, Θηκόγραμμα Μηνιαίου Εισοδήματος

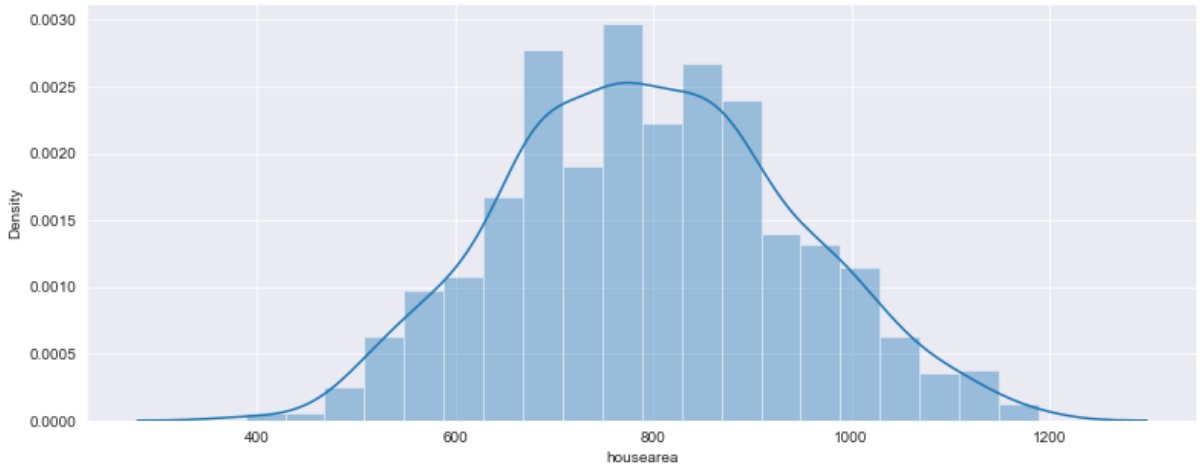


Γράφημα 3.3, Θηκόγραμμα Επιφάνειας Κατοικίας

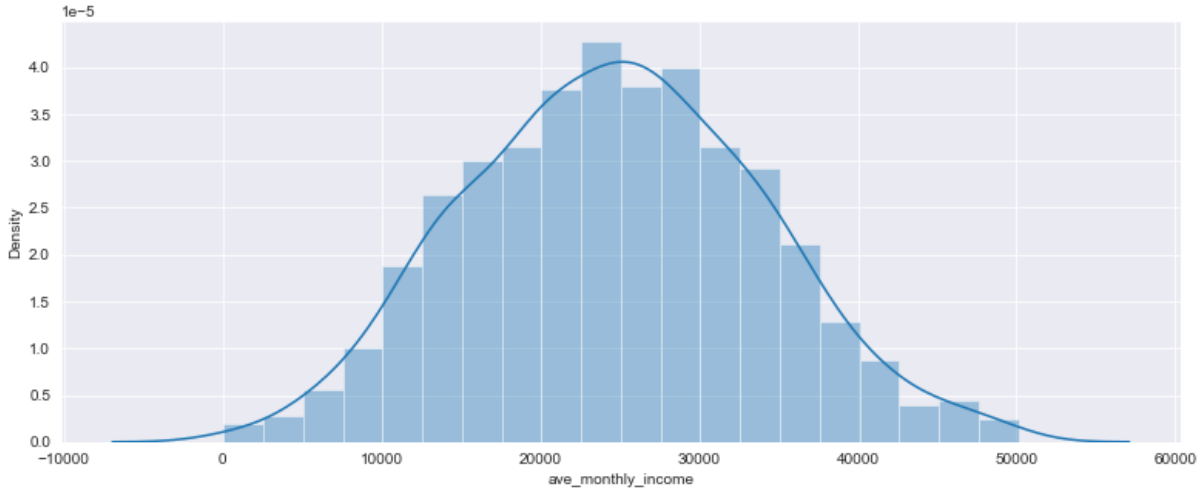
3.1.2 Περιγραφή Δεδομένων

Σε αυτό τα στάδιο πραγματοποιείται η περιγραφή των δεδομένων προτού αυτά χρησιμοποιηθούν για τα μοντέλα πρόβλεψης και την εύρεση του πιο αξιόπιστου αλγορίθμου.

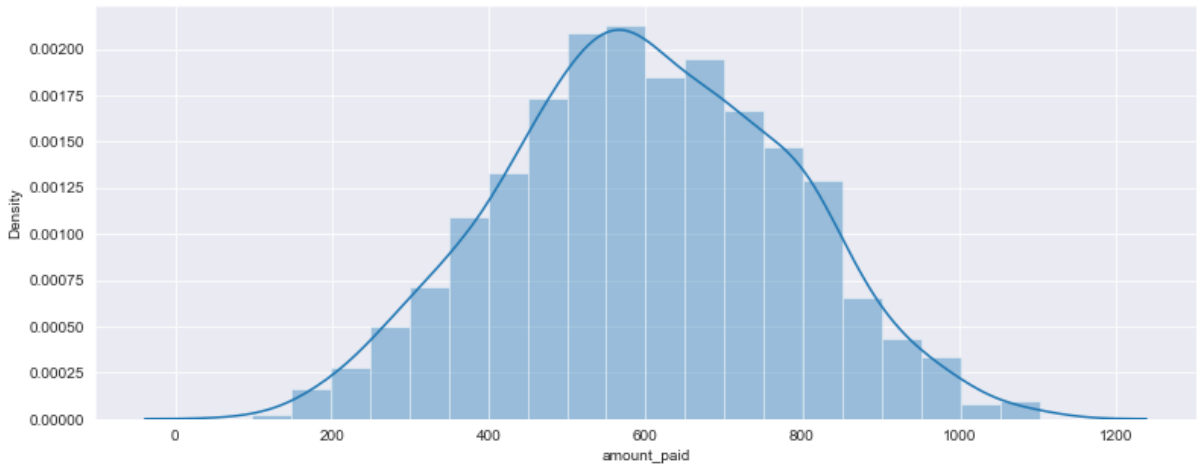
Στόχός είναι η καλύτερη κατανόηση των δεδομένων και η ανακάλυψη τυχόν σημαντικών πτυχών τους. Αρχικά, γίνεται η οπτικοποίηση της κατανομής για τις παραμέτρους της επιφάνειας του σπιτιού, του μηνιαίου εισοδήματος και του μηνιαίου λογαριασμού, όπως απεικονίζονται στα Γράφημα 3.4, 3.5 και 3.6.



Γράφημα 3.4, Κατανομή Επιφάνειας Κατοικίας



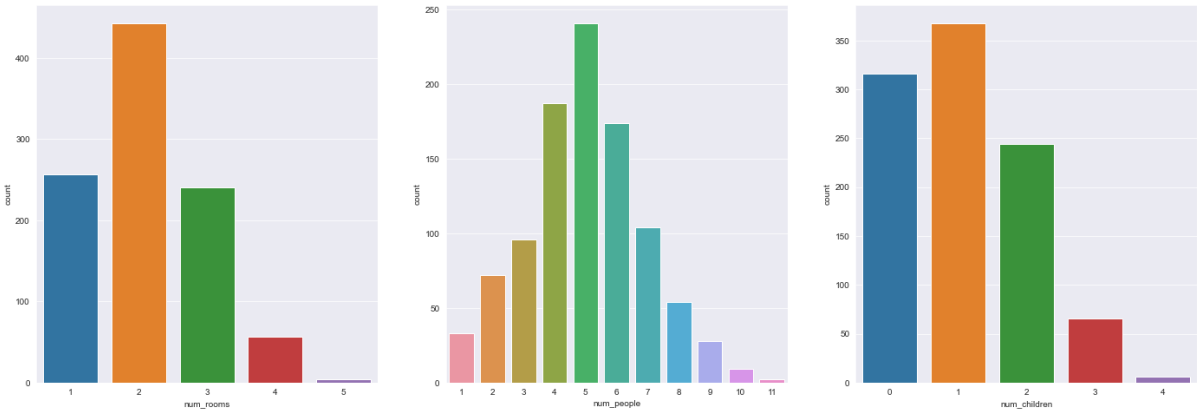
Γράφημα 3.5, Κατανομή Μηνιαίου Εισοδήματος



Γράφημα 3.6, Κατανομή Μηνιαίου Λογαριασμού

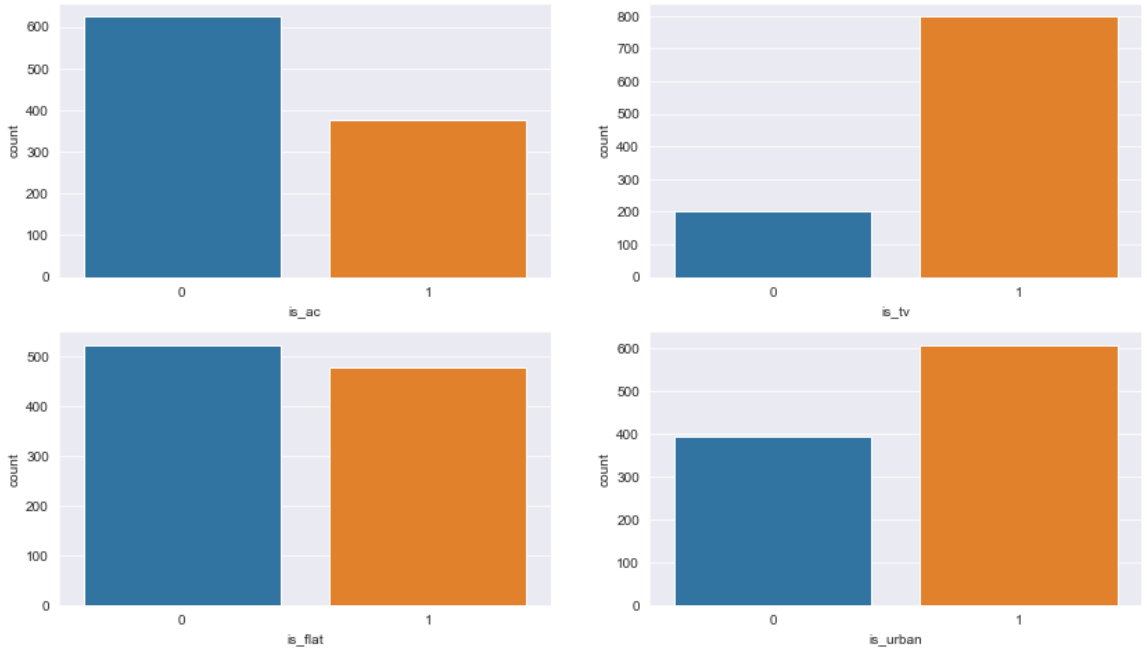
Για την παράμετρο της επιφάνειας της κατοικίας δεν είναι αρχικά ευδιάκριτη κάποια ασυμμετρία, ωστόσο υπολογίζοντας τον συντελεστή ασυμμετρίας καταλήγουμε στην τιμή 0.11 η οποία υποδηλώνει μια αμυδρά θετική ασυμμετρία για τα συγκεκριμένα δεδομένα. Στο ίδιο συμπέρασμα καταλήγουμε και για τις άλλες δύο παραμέτρους με συντελεστές ασυμμετρίας 0.07 και 0.01 για τον μηνιαίο εισόδημα και μηνιαίο λογαριασμό αντίστοιχα, δείγμα ότι η κατανομή είναι αρκετά συμμετρική με τα δεδομένα να κατανέμονται ομοιόμορφα.

Ακολουθεί η οπτικοποίηση των υπόλοιπων μεταβλητών του συνόλου δεδομένων, όπως απεικονίζονται στα Γραφήματα 3.7 και 3.8. Για τον αριθμό δωματίων παρατηρείται ότι οι περισσότερες οικίες έχουν εώς 3 δωμάτια, αντίθεται λίγες οικίες έχουν 4 ή 5 δωμάτια. Για τον αριθμό ανθρώπων που διαμένουν σε αυτές, παρατηρείται ότι οι περισσότερες οικίες αποτελούνται από 4 εώς 6 ενοίκους, και ότι υπάρχουν μερικές οικίες που κατοικούνται από 9 εώς 11 ενοίκους. Για την παράμετρο που αφορά τον αριθμό των παιδιών, η πλειοψηφία των κατοικιών αποτελείται μέχρι 2 παιδιά, και σε λίγες οικίες παρατηρείται μεγαλύτερος αριθμός με τον μέγιστο αριθμό να φτάνει τα 4 παιδιά.



Γράφημα 3.7, Ραβδόγραμμα για Αριθμό Δωματίων, Ανθρώπων και Παιδιών

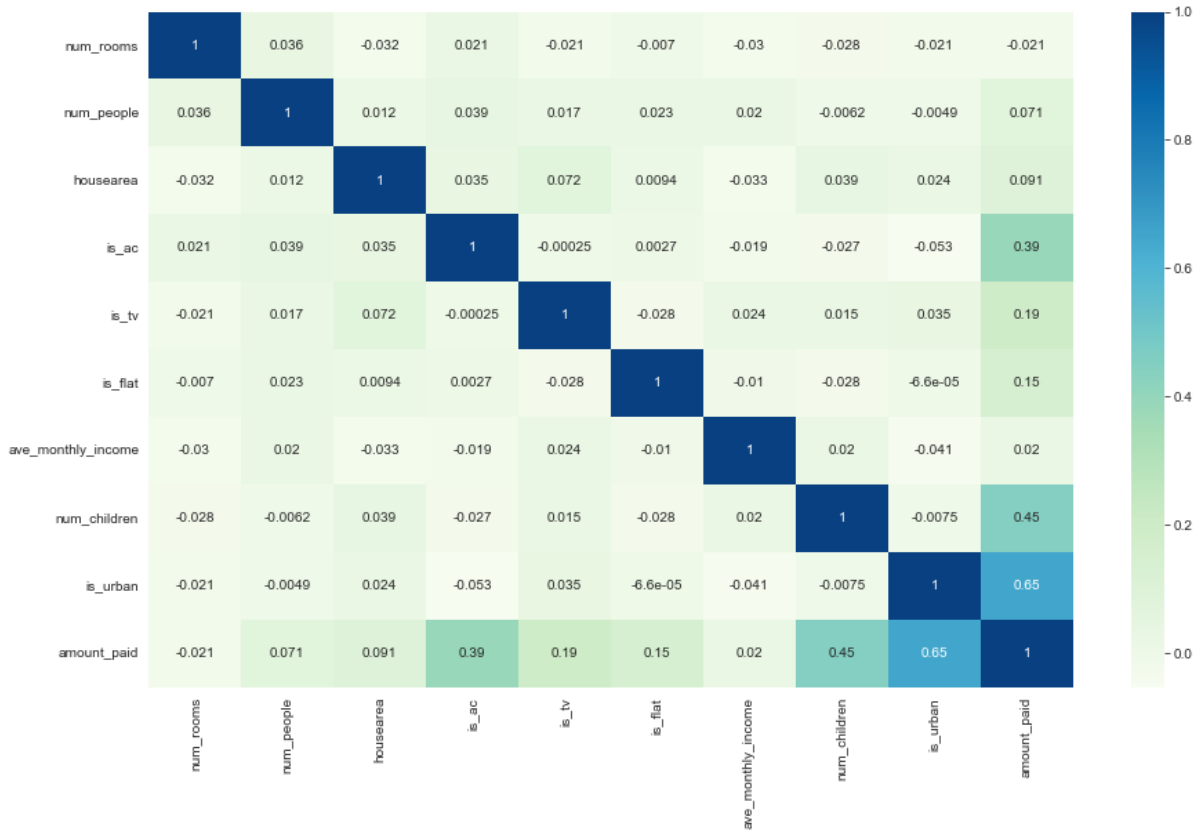
Συνεχίζοντας με τις υπόλοιπες παραμέτρους, για την κατοχή air condition παρατηρείται σχετική ισορροπία στα δεδομένα, με τις οικίες που δεν έχουν air condition να υπερέχουν αριθμητικά από εκείνες που έχουν. Για την παράμετρο που καταγράφει την κατοχή τηλεόρασης παρατηρείται ότι η πλειοψηφία του δείγματος, σχεδόν το 80%, κατέχει τηλεόραση.



Γράφημα 3.8, Ραβδόγραμμα για Air Condition, Τηλεόραση, Διαμέρισμα και Κατοικία Εντός Πόλεως

Ικανοποιητική ισορροπία εμφανίζεται στα δεδομένα που καταγράφουν εάν η οικία είναι διαμέρισμα, με τα διαμερίσματα να υπολείπονται αριθμητικά οριακά. Ολοκληρώνοντας, παρατηρείται μια σχετική ισορροπία στις κατοικίες που ανήκουν ή όχι στον αστικό ιστό με τις περισσότερες οικίες του δείγματος ωστόσο να βρίσκονται εντός πόλεως.

Η φάση της περιγραφής των δεδομένων ολοκληρώνεται με την απεικόνιση των συντελεστών συσχέτισης Pearson, όπως παρουσιάζεται στο Γράφημα 3.9, μια μέθοδος η οποία εξυπηρετεί στην αναγνώριση πιθανών συσχετίσεων μεταξύ των μεταβλητών.



Γράφημα 3.9, Απεικόνιση Συντελεστών Συσχέτισης Pearson

Βάσει του γραφήματους, διαπιστώνεται μια ισχυρή θετική συσχέτιση της εξαρτημένης μεταβλητής που καταγράφει την τιμή του μηνιαίου λογαριασμού με την ανεξάρτητη

μεταβλητή που καταγράφει εάν οι οικίες είναι εντός πόλεως. Θετική συσχέτιση παρατηρείται ακόμη με τον αριθμό παιδιών που διαμένουν στο σπίτι και με την ύπαρξη air condition. Αντίθετα, φαίνεται να μην επηρεάζεται ιδιαίτερα η εξαρτημένη μεταβλητή από άλλες παραμέτρους όπως την επιφάνεια της οικίας, τον αριθμό των δωματίων ή το σύνολο των ενοίκων της οικίας.

3.2 Επιλογή Αλγορίθμων και Παραμετροποίηση

Για την υλοποίηση αυτής της ανάλυσης χρησιμοποιήθηκαν 3 αλγόριθμοι αφού πρώτα μέρος των δεδομένων μετασχηματίστηκε μέσω της τεχνικής της κανονικοποίησης με εξαίρεση τα δεδομένα που λαμβάνουν τιμές 0 και 1. Οι αλγόριθμοι που επιλέχθηκαν είναι οι Ridge, SVM και Random Forest.

Για τον αλγόριθμο Ridge, επεξεργάστηκε η παράμετρος alpha η οποία και καθορίζει το βαθμό τιμωρίας για το μοντέλο. Επιλέχθηκε η προκαθορισμένη τιμή 1.0 η οποία και είναι και η απόλυτη τιμωρία. Για τον αλγόριθμο SVM, επεξεργάστηκαν οι παράμετροι C και kernel οι οποίες καθορίζουν αντίστοιχα τον βαθμό τιμωρίας και τις μαθηματικές εξισώσεις που θα χρησιμοποιηθούν. Για την παράμετρο C έγινε χρήση της τιμής 500 και για την παράμετρο kernel η γραμμική εξίσωση.

Για τον αλγόριθμο Random Forest, επεξεργάστηκαν η παράμετρος n estimators (=100) που καθορίζει τον αριθμό των δέντρων, η παράμετρος max depth (=50) που καθορίζει το βάθος τους, η παράμετρος max features (=4) που αφορά τον μέγιστο αριθμό των μεταβλητών, η παράμετρος min samples split (=8) που αφορά τον ελάχιστο αριθμό που επιτρέπεται διαχωρισμός, η παράμετρος min samples leaf (=4) που καθορίζει τον ελάχιστο αριθμό που επιτρέπεται να προκύψει μετά από διαχωρισμό και η παράμετρος bootstrap (=True) που αφορά εάν ένα δείγμα που επιλέγεται είναι διαφορετικό από ένα άλλο δείγμα.

3.3 Εκπαίδευση και Αξιολόγηση

Σε πρώτο στάδιο, τα δεδομένα χωρίστηκαν με την κλασική μέθοδο Train/Split και αναλογία 80/20, όπου το 80% των δεδομένων χρησιμοποιήθηκε για εκπαίδευση και το 20% για έλεγχο και αξιολόγηση. Σε μετέπειτα στάδιο, επιλέχθηκε να εφαρμοστεί μια διαφορετική μέθοδος εκπαίδευσης από τον κλασικό διαχωρισμό των δεδομένων, η τεχνική του K – Fold Cross Validation η οποία και βελτιώνει την απόδοση των αλγορίθμων. Η τεχνική αυτή χρησιμοποιήθηκε και για την εύρεση των τιμών των παραμέτρων που μεγιστοποιούν την απόδοση των αλγορίθμων, με την εκπαίδευση και αξιολόγηση να επαναλαμβάνεται με τις βέλτιστες τιμές. Ανεξάρτητα από την επιλογή εκπαίδευσης, τα αποτελέσματα αξιολογήθηκαν με κοινούς δείκτες αξιολόγησης και συγκεκριμένα τους δείκτες MSE, RMSE και MAE.

ΚΕΦΑΛΑΙΟ 4

4.1 Αποτελέσματα Μοντέλων με Διαχωρισμό Train/Split

Η αξιολόγηση των μοντέλων ξεκινάει με την παρουσίαση των αποτελεσμάτων μετά τον διαχωρισμό των δεδομένων σε Train/Split με αναλογία 80/20. Όπως παρουσιάζεται στον Πίνακα 4.1, ο αλγόριθμος Ridge δείχνει να έχει τα καλύτερα αποτελέσματα και στους 3 δείκτες, με τον αλγόριθμο SVR να πλησιάζει στην απόδοση. Αντίθετα, ο αλγόριθμος Random Forest δείχνει να έχει τα χειρότερα αποτελέσματα και να κατατάσσεται τελευταίος στην απόδοση ανάμεσα στους χρησιμοποιηθέντες αλγόριθμους.

ΔΕΙΚΤΕΣ ΑΞΙΟΛΟΓΗΣΗΣ ---------- ΑΛΓΟΡΙΘΜΟΙ	RMSE	MSE	MAE
Ridge	**67.79**	**4595.06**	**59.21**
SVR	68.54	4697.08	59.69
Random Forest	74.57	5560.46	62.51

Πίνακας 4.1, Αποτελέσματα με διαχωρισμό Train/Split

4.2 Αποτελέσματα Μοντέλων με τεχνική K – Fold Cross Validation

Η αξιολόγηση των μοντέλων συνεχίζεται με την παρουσίαση των αποτελεσμάτων μετά την χρήση της τεχνικής K – Fold Cross Validation. Όπως παρουσιάζεται στον Πίνακα 4.2, ο αλγόριθμος Ridge δείχνει να έχει ξανά τα καλύτερα αποτελέσματα σε όλους τους δείκτες, με τον αλγόριθμο SVR να ακολουθεί με μικρή απόκλιση. Αντίθετα, ο αλγόριθμος Random Forest δείχνει να έχει τα χειρότερα αποτελέσματα και να τα κατατάσσεται ξανά τελευταίος στην απόδοση. Ωστόσο, βελτίωση παρατηρείται σε όλους τους αλγόριθμους.

ΔΕΙΚΤΕΣ ΑΞΙΟΛΟΓΗΣΗΣ ---------- ΑΛΓΟΡΙΘΜΟΙ	RMSE	MSE	MAE
Ridge	**64.73**	**4217.5**	**54.78**
SVR	65.24	4286.15	54.93
Random Forest	71.0	5062.78	58.97

Πίνακας 4.2, Αποτελέσματα με τεχνική K – Fold Cross Validation

Για να βελτιστοποιήσουμε την απόδοση αλγορίθμων γίνεται αναζήτηση εκείνων των τιμών που επιτρέπουν να επιτευχθεί η υψηλότερη απόδοση για τα μοντέλα. Οι παρακάτω τιμές παρουσιάζονται ως βέλτιστες:

- Ridge: alpha = 0.99
- SVR: kernel = linear, C= 60
- Random Forest: estimators = 560, bootstrap = False, max features = 4, max depth = 125, min samples leaf = 5, min samples split = 11

Με την χρήση της τεχνικής K – Fold Cross Validation, τα μοντέλα εκπαιδεύονται με τις νέες παραμέτρους και τα αποτελέσματα παρουσιάζονται στον Πίνακα 4.3.

ΔΕΙΚΤΕΣ ΑΞΙΟΛΟΓΗΣΗΣ ---------- ΑΛΓΟΡΙΘΜΟΙ	RMSE	MSE	MAE
Ridge	**64.73**	**4217.49**	**54.78**
SVR	65.21	4283.06	54.9
Random Forest	69.66	4875.83	57.66

Πίνακας 4.3, Αποτελέσματα με Βέλτιστες Παραμέτρους

Παρατηρείται ελάχιστη βελτίωση των αποτελεσμάτων για τους αλγόριθμους Ridge και SVR, δείγμα ότι οι τιμές που είχαν χρησιμοποιηθεί ήδη ήταν κοντά στις βέλτιστες ή δεν είχαν μεγάλη επίδραση στην τελική απόδοση. Αντίθετα, βελτίωση στην απόδοση σημειώνεται για τον αλγόριθμο Random Forest. Παρατηρείται επίσης ότι ο αλγόριθμος Ridge συνεχίζει να παρουσιάζει τα καλύτερα αποτελέσματα, με τον αλγόριθμο SVR να ακολουθεί και τον αλγόριθμο Random Forest να κατατάσσεται τελευταίος βάσει των δεικτών αξιολόγησης που χρησιμοποιούνται.

4.3 Συμπεράσματα και Προτάσεις

Οι αλγόριθμοι Ridge και SVR παρουσίασαν υψηλότερη απόδοση στο να προβλέψουν το κόστος ηλεκτρικού ρεύματος που θα πληρωθεί από τα νοικοκυριά, με τον αλγόριθμο Ridge να παρέχει ελαφρώς καλύτερα αποτελέσματα. Ο αλγόριθμος Random Forest δεν απέδωσε αντίστοιχα με πιθανή την ανάγκη μεγαλύτερου συνόλου δεδομένων για βελτιστοποίηση της απόδοσης του στην πρόβλεψη του κόστους ηλεκτρικού ρεύματος. Επομένως, η συλλογή

περισσότερων δεδομένων πιθανόν να διαφοροποιήσει τα αποτελέσματα και να παρέχει μια διαφορετική εικόνα για τον πιο αξιόπιστο αλγόριθμο.

Η περιγραφική ανάλυση έδειξε ότι οι κυριότεροι παράμετροι του συνόλου δεδομένων που επηρεάζουν την εξαρτημένη μεταβλητή είναι εάν τα νοικοκυριά βρίσκονται εντός πόλεως, ο αριθμός των παιδιών και η ύπαρξη air condition. Για να μπορέσουν οι προβλέψεις να γίνουν πιο αξιόπιστες, θα ήταν σημαντική η συλλογή περισσότερων παραμέτρων που θεωρητικά μπορούν να επηρεάσουν το κόστος πληρωμής, όπως η περίοδος κατανάλωσης ηλεκτρικού ρεύματος με πιθανή μεγαλύτερη κατανάλωση τους καλοκαιρινούς μήνες.

Ολοκληρώνοντας, θα πρέπει να σημειωθεί ότι η πρόβλεψη τους κόστους κατανάλωσης ενέργειας είναι μια πολύπλοκη διαδικασία η οποία μπορεί να επηρεαστεί από μια σειρά από παράγοντες, όπως η υπάρχουσα ενεργειακή κρίση σε παγκόσμιο επίπεδο, επομένως η πρόβλεψη με βάση δημογραφικά χαρακτηριστικά δεν αποτελεί παρά ένα κομμάτι στην διαδικασία κατανόησης και πρόβλεψης.

ΒΙΒΛΙΟΓΡΑΦΙΑ

1. Grus, J. (2015). Data Science from Scratch, First Principles with Python, O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472.
2. Han, J., Kamber, M. and Pei, J. (2012). Data mining Concepts and Techniques. Amsterdam, Elsevier
3. Muller, A. C., Guido, S. (2017). Introduction to Machine Learning with Python, A Guide for Data Scientists, O'Reilly Media, Inc., 1005 Gravenstein Highway, Sebastopol, CA 95472
4. Mitchell, T. M. (1997). Machine Learning, McGraw-Hill Science/Engineering/Math
5. Awad, M., Khanna R. (2015). Efficient Learning Machines, Theories, Concepts and Applications for Engineers and System Designers, Apress Open
6. Kantardzic, M. (2011). Data mining, Concepts, Models, Methods, and Algorithms, Wiley-IEEE Press
7. Ye, N. (2014). Data Mining, Theories, Algorithms and Examples, CRC Press

ΔΙΑΔΙΚΤΥΑΚΟΙ ΙΣΤΟΤΟΠΟΙ

1. https://www.wikipedia.org/
2. https://machinelearningmastery.com/
3. https://towardsdatascience.com/
4. https://www.analyticsvidhya.com/blog/
5. https://www.geeksforgeeks.org/
6. https://stackoverflow.com/
7. https://www.kaggle.com
8. https://scikit-learn.org/

ΠΑΡΑΡΤΗΜΑ

```
#Standardization of non binary data

data.num_rooms = preprocessing.scale(data.num_rooms)

data.num_people = preprocessing.scale(data.num_people)

data.housearea = preprocessing.scale(data.housearea)

data.ave_monthly_income = preprocessing.scale(data.ave_monthly_income)

data.num_children = preprocessing.scale(data.num_children)


#Training models on split 80/20

train_data, test_data = train_test_split (data, test_size = 0.2, random_state = 1)


#Ridge training

ridge = Ridge(alpha=1.0)

ridge.fit(train_data.iloc[:,:9], train_data.amount_paid)

predicted_amount_paid = ridge.predict(test_data.iloc[:,:9])

rmse = math.sqrt(mean_squared_error(test_data.amount_paid, predicted_amount_paid))

mse = np.square(np.subtract(test_data.amount_paid, predicted_amount_paid)).mean()

mae = mean_absolute_error(test_data.amount_paid, predicted_amount_paid)
```

```
#SVR training

svr = SVR (kernel='linear', C= 500)

svr.fit(train_data.iloc[:,:9], train_data.amount_paid)

predicted_amount_paid = svr.predict(test_data.iloc[:,:9])

rmse = math.sqrt(mean_squared_error(test_data.amount_paid, predicted_amount_paid))

mse = np.square(np.subtract(test_data.amount_paid, predicted_amount_paid)).mean()

mae = mean_absolute_error(test_data.amount_paid, predicted_amount_paid)


#Random Forest training

rfr = RandomForestRegressor (random_state=1, n_estimators=100, max_depth=50,
max_features=4, min_samples_leaf=4, min_samples_split=8, bootstrap = True)

rfr.fit(train_data.iloc[:,:9], train_data.amount_paid)

predicted_amount_paid = rfr.predict(test_data.iloc[:,:9])

rmse = math.sqrt(mean_squared_error(test_data.amount_paid, predicted_amount_paid))

mse = np.square(np.subtract(test_data.amount_paid, predicted_amount_paid)).mean()

mae = mean_absolute_error(test_data.amount_paid, predicted_amount_paid)


#Training models on K-Fold cross validation


#Ridge training by setting alpha 1.0

input_data, target_data = data.iloc[:,:9], data.amount_paid

model = Ridge(alpha=1.0)

cv = RepeatedKFold(n_splits=10, n_repeats=3, random_state=1)
```

```
score1 = cross_val_score(model, input_data, target_data,
scoring='neg_root_mean_squared_error', cv=cv, n_jobs=-1)

score1 = absolute(score1)

score2 = cross_val_score(model, input_data, target_data, scoring='neg_mean_squared_error',
cv=cv, n_jobs=-1)

score2 = absolute(score2)

score3 = cross_val_score(model, input_data, target_data, scoring='neg_mean_absolute_error',
cv=cv, n_jobs=-1)

score3 = absolute(score3)


#Finding best alpha value for Ridge

model = Ridge()

cv = RepeatedKFold(n_splits=10, n_repeats=3, random_state=1)

grid = dict()

grid['alpha'] = arange(0, 1, 0.01)

search = GridSearchCV(model, grid, cv=cv, n_jobs=-1)

results = search.fit(input_data, target_data)


#Ridge training by setting best alpha value

model = Ridge(alpha=0.99)

cv = RepeatedKFold(n_splits=10, n_repeats=3, random_state=1)

score1 = cross_val_score(model, input_data, target_data,
scoring='neg_root_mean_squared_error', cv=cv, n_jobs=-1)

score1 = absolute(score1)
```

```
score2 = cross_val_score(model, input_data, target_data, scoring='neg_mean_squared_error', cv=cv, n_jobs=-1)

score2 = absolute(score2)

score3 = cross_val_score(model, input_data, target_data, scoring='neg_mean_absolute_error', cv=cv, n_jobs=-1)

score3 = absolute(score3)


#SVR training by setting linear kernel and 500 for C

model = SVR (kernel='linear',  C = 500)

cv = RepeatedKFold(n_splits=10, n_repeats=3, random_state=1)

score1 = cross_val_score(model, input_data, target_data,
scoring='neg_root_mean_squared_error', cv=cv, n_jobs=-1)

score1 = absolute(score1)

score2 = cross_val_score(model, input_data, target_data, scoring='neg_mean_squared_error', cv=cv, n_jobs=-1)

score2 = absolute(score2)

score3 = cross_val_score(model, input_data, target_data, scoring='neg_mean_absolute_error', cv=cv, n_jobs=-1)

score3 = absolute(score3)


#Finding best kernel and C value for SVR

model = SVR()

cv = RepeatedKFold(n_splits=10, n_repeats=3, random_state=1)

grid = dict()
```

```
grid['C'] = np.arange(5, 70, 5)

grid['kernel'] = ['linear', 'poly', 'sigmoid', 'rbf']

search = GridSearchCV(model, grid, cv=cv, n_jobs=-1)

results = search.fit(input_data, target_data)


#SVR training by setting best values for kernel and C

model = SVR (kernel='linear', C = 60)

cv = RepeatedKFold(n_splits=10, n_repeats=3, random_state=1)

score1 = cross_val_score(model, input_data, target_data,
scoring='neg_root_mean_squared_error', cv=cv, n_jobs=-1)

score1 = absolute(score1)

score2 = cross_val_score(model, input_data, target_data, scoring='neg_mean_squared_error',
cv=cv, n_jobs=-1)

score2 = absolute(score2)

score3 = cross_val_score(model, input_data, target_data, scoring='neg_mean_absolute_error',
cv=cv, n_jobs=-1)

score3 = absolute(score3)


#Random Forest training by setting parameters

model = RandomForestRegressor(random_state=1, n_estimators=100, max_depth=50,
max_features=4, min_samples_leaf=4, min_samples_split=8, bootstrap = True)

cv = RepeatedKFold(n_splits=10, n_repeats=3, random_state=1)

score1 = cross_val_score(model, input_data, target_data,
scoring='neg_root_mean_squared_error', cv=cv, n_jobs=-1)
```

```
score1 = absolute(score1)

score2 = cross_val_score(model, input_data, target_data, scoring='neg_mean_squared_error', cv=cv, n_jobs=-1)

score2 = absolute(score2)

score3 = cross_val_score(model, input_data, target_data, scoring='neg_mean_absolute_error', cv=cv, n_jobs=-1)

score3 = absolute(score3)


#Finding best parameters for Random Forest Regressor

model = RandomForestRegressor()

cv = RepeatedKFold(n_splits= 10, n_repeats=3, random_state=1)

grid = dict()

grid['bootstrap'] = [True, False]

grid['max_features'] = np.arange(4, 6, 1)

grid['min_samples_leaf'] =  np.arange(4, 6, 1)

grid['max_depth'] = np.arange(120, 130, 5)

grid['n_estimators'] = np.arange(560, 580, 10)

grid['min_samples_split'] =  np.arange(11, 12, 1)

search = GridSearchCV(model, grid, cv=cv, n_jobs=-1)

results = search.fit(input_data, target_data)

print(results.best_params_)


#Random Forest Regressor training by setting best values
```

```
model = RandomForestRegressor(random_state=1, n_estimators = 560, max_depth= 125,
max_features = 4, min_samples_leaf = 5, min_samples_split = 11, bootstrap = False)

cv = RepeatedKFold(n_splits=10, n_repeats=3, random_state=1)

score1 = cross_val_score(model, input_data, target_data,
scoring='neg_root_mean_squared_error', cv=cv, n_jobs=-1)

score1 = absolute(score1)

score2 = cross_val_score(model, input_data, target_data, scoring='neg_mean_squared_error',
cv=cv, n_jobs=-1)

score2 = absolute(score2)

score3 = cross_val_score(model, input_data, target_data, scoring='neg_mean_absolute_error',
cv=cv, n_jobs=-1)

score3 = absolute(score3)
```