ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

Σχολή Χρηματοοικονομικής και Στατιστικής

Τμήμα Στατιστικής και Ασφαλιστικής Επιστήμης

ΜΕΤΑΠΤΥΧΙΑΚΟ ΠΡΟΓΡΑΜΜΑ ΣΠΟΥΔΩΝ
ΣΤΗΝ **ΕΦΑΡΜΟΣΜΕΝΗ ΣΤΑΤΙΣΤΙΚΗ**

ΕΦΑΡΜΟΓΕΣ ΜΗΧΑΝΙΚΗΣ ΜΑΘΗΣΗΣ ΣΤΗΝ ΝΑΥΤΙΛΙΑ

Παναγιώτης Μπίρης

Διπλωματική Εργασία

που υποβλήθηκε στο Τμήμα Στατιστικής και Ασφαλιστικής Επιστήμης του Πανεπιστημίου Πειραιώς ως μέρος των απαιτήσεων για την απόκτηση του Μεταπτυχιακού Διπλώματος Ειδίκευσης στην *Εφαρμοσμένη Στατιστική*

Πειραιάς,
Ιούλιος 2023

Π Α Ν Ε Π Ι Σ Τ Η Μ Ι Ο Π Ε Ι Ρ Α Ι Ω Σ

Σχολή Χρηματοοικονομικής και Στατιστικής

Τμήμα Στατιστικής και Ασφαλιστικής Επιστήμης

ΜΕΤΑΠΤΥΧΙΑΚΟ ΠΡΟΓΡΑΜΜΑ ΣΠΟΥΔΩΝ
ΣΤΗΝ **ΕΦΑΡΜΟΣΜΕΝΗ ΣΤΑΤΙΣΤΙΚΗ**

ΕΦΑΡΜΟΓΕΣ ΜΗΧΑΝΙΚΗΣ ΜΑΘΗΣΗΣ ΣΤΗΝ ΝΑΥΤΙΛΙΑ

Παναγιώτης Μπίρης

Διπλωματική Εργασία

που υποβλήθηκε στο Τμήμα Στατιστικής και Ασφαλιστικής Επιστήμης του Πανεπιστημίου Πειραιώς ως μέρος των απαιτήσεων για την απόκτηση του Μεταπτυχιακού Διπλώματος Ειδίκευσης στην *Εφαρμοσμένη Στατιστική*

Πειραιάς,
Ιούλιος 2023

Η παρούσα Διπλωματική Εργασία εγκρίθηκε ομόφωνα από την Τριμελή Εξεταστική Επιτροπή που ορίσθηκε από τη Συνέλευση του Τμήματος Στατιστικής και Ασφαλιστικής Επιστήμης του Πανεπιστημίου Πειραιώς στην υπ' αριθμ. ..... συνεδρίασή της σύμφωνα με τον Εσωτερικό Κανονισμό Λειτουργίας του Προγράμματος Μεταπτυχιακών Σπουδών στην Εφαρμοσμένη Στατιστική

Τα μέλη της Επιτροπής ήταν:

- Σωτήριος Μπερσίμης (Αναπληρωτής καθηγητής) (Επιβλέπων)
- Σωτήριος Τασουλής (Επίκουρος καθηγητής)
- Αθανάσιος Ρακιντζής (Επίκουρος καθηγητής)

Η έγκριση της Διπλωματικής Εργασίας από το Τμήμα Στατιστικής και Ασφαλιστικής Επιστήμης του Πανεπιστημίου Πειραιώς δεν υποδηλώνει αποδοχή των γνωμών του συγγραφέα.

UNIVERSITY OF PIRAEUS

School of Finance and Statistics

Department of Statistics and Insurance Science

POSTGRADUATE PROGRAM IN
APPLIED STATISTICS

APPLICATIONS OF MACHINE LEARNING IN SHIPPING INDUSTRY

By

Panagiotis Biris

MSc Dissertation

submitted to the Department of Statistics and Insurance Science of the University of Piraeus in partial fulfilment of the requirements for the degree of Master of Science in Applied Statistics

Piraeus, Greece
July 2023

Στην οικογένειά μου

Ευχαριστίες

Ολοκληρώνοντας την παρούσα διπλωματική εργασία, θα ήθελα να ευχαριστήσω την οικογένεια μου για όλη την κατανόηση και συμπαράσταση που μου έδειξε και την στήριξη που μου παρείχε όλα αυτά τα χρόνια σπουδών μου.

Θα ήθελα επίσης να ευχαριστήσω θερμά και να εκφράσω την ευγνωμοσύνη μου στον καθηγητή μου κ. Σωτήρη Μπερσίμη για την καθοδήγησή του, την στήριξη και την εμπιστοσύνη που μου έδειξε σε όλα αυτά τα χρόνια σπουδών μου στο Πανεπιστήμιο του Πειραιά.

Περίληψη

Η ταχύτητα με την οποία εξελίσσεται η βιομηχανία της ναυτιλίας καθώς και το διογκωμένο πλήθος δεδομένων που ρέει καθημερινά προς τις ναυτιλιακές εταιρίες, έχει κάνει την ανάγκη για εξόρυξη γνώσης μέσα από αυτά τα δεδομένα ακόρεστη. Οι αποφάσεις και το στρατηγικό πλάνο των ναυτιλιακών οργανισμών βασίζεται πλέον, σχεδόν εξ 'ολοκλήρου, στην χρήσιμη εξαγωγή γνώσης και πληροφορίας που υπάρχει ακατέργαστη, μέσα σε αυτόν τον τεράστιο όγκο δεδομένων. Για την εξόρυξη και την ανάλυση των διαθέσιμων δεδομένων απαιτείται η χρήση στατιστικής, μαθηματικών και πληροφορικής αλλά παράλληλα απαιτείται και υπολογιστική ισχύς. Ο συνδυασμός αυτών των τριών επιστημονικών κλάδων και η αναγκαιότητα ύπαρξης μεγάλης υπολογιστικής ισχύος, έχει φέρει στην επιφάνεια τα τελευταία χρόνια την επιστήμη της στατιστικής μηχανικής μάθησης, όπου τα μοντέλα και οι αλγόριθμοι που περιέχει, εκτός από το να επεξεργάζονται, να αναλύουν και τελικά να εξάγουν χρήσιμα συμπεράσματα και κρυφή γνώση μέσα από τα δεδομένα, προσπαθούν με ένα ευφυή και καινοτόμο τρόπο να εκπαιδευτούν πάνω στα διαθέσιμα δεδομένα των ναυτιλιακών εταιρών και να προβλέψουν μελλοντικές, αβέβαιες καταστάσεις όταν νέα δεδομένα εισέρχονται στους ναυτιλιακούς οργανισμούς. Στην παρούσα διπλωματική εργασία γίνεται μια λεπτομερής και εκτενής αναφορά στις πολλές και διαφορετικές κατηγορίες και μεθόδους στατιστικής μηχανικής μάθησης και στατιστικής ανάλυσης των δεδομένων και παρουσιάζονται μερικές μελέτες και εφαρμογές της στατιστικής μηχανικής μάθησης στον χώρο της ναυτιλίας.

Abstract

The speed at which the shipping industry is evolving, as well as the burgeoning amount of data that flows daily to shipping companies, has created an insatiable need to mine knowledge from this data. The decisions and the strategic plan of shipping organizations are now based, almost entirely, on the useful extraction of knowledge and information that exists raw, within this huge volume of data. The use of statistics, mathematics and informatics as well as computing power is used for the analysis of the available data and to draw conclusions. In recent years, the combination of these three disciplines and the necessity of having a lot of computing power, has brought to the surface the scientific field of statistical machine learning, where its models and algorithms, in addition to processing, analyzing and ultimately extracting useful conclusions and hidden insights from the data, try in an intelligent and innovative way to train on the available data of shipping companies in order to predict future, uncertain situations when new data enter the shipping organizations. In this thesis, a detailed and extensive reference is made to many different categories and methods of statistical machine learning and statistical analysis of data, and some studies and applications of statistical machine learning in the shipping field are presented.

Περιεχόμενα

Κατάλογος πινάκων

Κατάλογος σχημάτων

ΚΕΦΑΛΑΙΟ 1^{Ο}

1. Εισαγωγή

1.1 Η βιομηχανία της ναυτιλίας και ο τεράστιος όγκος δεδομένων

Ο τεράστιος όγκος δεδομένων που ρέει καθημερινά προς τις ναυτιλιακές εταιρίες τα τελευταία χρόνια και οι αυξημένες ανάγκες της ναυτιλιακής αγοράς, έχουν καταστήσει μεγάλη πρόκληση για την ναυτιλιακή βιομηχανία την εξαγωγή χρήσιμης γνώσης από τέτοιου είδους αδόμητα δεδομένα. Όπως σε σχεδόν όλες τις βιομηχανίες, έτσι και η ναυτιλία έχει επηρεαστεί από τον τεράστιο όγκο δεδομένων και πληροφοριών που είναι διαθέσιμα και τα οποία απαιτούν ειδική μεταχείριση και ανάλυση ώστε να αποκαλυφθεί η «κρυφή» γνώση μέσα τους. Ο ανταγωνιστικός χαρακτήρας της συγκεκριμένης βιομηχανίας έχει στρέψει τις ναυτιλιακές εταιρίες στο να αναζητούν πιο αποδοτικές μεθόδους για υλοποίηση των δρομολογίων των πλοίων και με το όσο το δυνατόν λιγότερο κόστος. Η ανάγκη για γρήγορη και ασφαλή μεταφορά προϊόντων και αγαθών στα διάφορα μέρη του κόσμου με την ελάχιστη δυνατή κατανάλωση καυσίμων και συνεπώς με ελάχιστο κόστος, έχει κάνει απαραίτητη για την ναυτιλιακή βιομηχανία την ανάγκη για εύρεση νέων, καινοτόμων, οικονομικών τρόπων ταξιδιού μέσω θαλάσσης. Στον ορίζοντα νέοι στόχοι και προβλήματα εμφανίζονται συνεχώς και πλέον την λύση στα πολλά ζητήματα του ναυτιλιακού κλάδου έρχεται να δώσει η επιστήμη της πληροφορικής και της στατιστικής. Δεδομένα που αφορούν τα χαρακτηριστικά των πλοίων, τις καταναλώσεις τους, τις εκπομπές ρύπων τους αλλά και την λειτουργικότητά τους βοηθούν αναλύοντάς τα, τις ναυτιλιακές εταιρίες, να οργανώσουν ένα στρατηγικό πλάνο το οποίο πρόκειται να κάνει τα ταξίδια μέσω θαλάσσης πιο κερδοφόρα, περισσότερο ασφαλή αλλά και πιο φιλικά προς το περιβάλλον μειώνοντας αρκετά τις εκπομπές διοξειδίου του άνθρακα που εκπέμπονται από τα πλοία.

1.2 Αξιοποίηση των διαθέσιμων δεδομένων

Για την ανακάλυψη σημαντικών πληροφοριών και κρυφής γνώσης μέσα από το υπέρογκο πλήθος των διαθέσιμων δεδομένων τρία επιστημονικά πεδία συνδυάζονται για την καλύτερη δυνατή επεξεργασία, ανάλυση και παρουσίασή αυτών των δεδομένων. Τα τρία επιστημονικά πεδία είναι αυτά των μαθηματικών, της πληροφορικής και της στατιστικής. Ο συνδυασμός αυτών των επιστημονικών κλάδων, έχει διαμορφώσει έναν νέο σχετικά τομέα, αυτόν της στατιστικής μηχανικής μάθησης. Με την είσοδο της στατιστικής μηχανικής μάθησης στις βιομηχανίες και

φυσικά στην βιομηχανία της ναυτιλίας πολλά πολύπλοκα προβλήματα που ήταν σχεδόν αδύνατον να επιλυθούν στο παρελθόν λόγω της περιορισμένης υπολογιστικής ισχύος, μπορούν πλέον να προσεγγιστούν και να επιλυθούν μέσω της μηχανικής μάθησης και πιο συγκεκριμένα μέσω των διαφόρων τεχνικών, αλγορίθμων και των μοντέλων προβλέψεων που περιέχει. Αυτές οι τεχνικές και τα μοντέλα προβλέψεων της στατιστικής μηχανικής μάθησης μπορούν να εξάγουν ακριβή, αξιόπιστα και γρήγορα συμπεράσματα με έναν ευφυή τρόπο, αναλύοντας τα υπάρχοντα δεδομένα, μαθαίνοντας μέσα από αυτά και κάνοντας προβλέψεις όταν νέα δεδομένα εισέρχονται στους ναυτιλιακούς οργανισμούς. Πολλά στοιχεία, από διαφορετικές και πολλές πηγές ήταν πολύ δύσκολο και πολύπλοκο για τον ανθρώπινο νου να τα επεξεργαστεί και να τα αναλύσει μέσω της παρατήρησης, οπότε η στατιστική μηχανική μάθηση και ειδικότερα οι τεχνικές και τα μοντέλα που διαθέτει, λειτουργώντας κατά μία έννοια όπως ο ανθρώπινος νους, προσφέρουν πολύτιμη βοήθεια για την επίτευξη των στόχων των ναυτιλιακών εταιριών και την επίλυση καθημερινών πολύπλοκων προβλημάτων που εμφανίζονται.

ΚΕΦΑΛΑΙΟ 2^{Ο}

2. Ναυτιλία

2.1 Η εξέλιξη της ναυτιλίας

Η ανάγκη των εθνών ανά τους αιώνες για την μεταφορά αγαθών και εμπορευμάτων από χώρα σε χώρα και από ήπειρο σε ήπειρο με όσο το δυνατόν λιγότερο κόστος, με περισσότερη ασφάλεια αλλά και σε μικρότερο χρονικό διάστημα, ήταν το έναυσμα για την ανάπτυξη του κλάδου της ναυτιλίας. Η χρονοβόρα διαδικασία μεταφοράς προϊόντων μέσω ξηράς, το αυξημένο οικονομικό κόστος, η αδυναμία εμπορικών συναλλαγών με απομακρυσμένα μέρη της γης, και οι κίνδυνοι που παραμόνευαν σε τέτοια μεγάλα ταξίδια, οδήγησαν τα κράτη με τον καιρό, να αναπτύσσουν συνεχώς τον στόλο τους. Με την άνθιση της ναυτιλίας και φυσικά την εξέλιξή της με την πάροδο των χρόνων, τα κράτη κατάφεραν να διευρύνουν τους εμπορικούς τους ορίζοντες και να έρθουν σε επαφή με άλλες απομακρυσμένες χώρες και ηπείρους πράγμα που βοήθησε σημαντικά στην περαιτέρω ανάπτυξη και βελτίωση της οικονομίας τους και του πολιτισμού τους. Οι καλύτερες συνθήκες που υπάρχουν στα κατάλληλα διαμορφωμένα πλοία ανάλογα με την φύση του κάθε αγαθού και εμπορεύματος, η κατά πολύ μειωμένη διάρκεια των ταξιδιών καθώς και η εξοικονόμηση χρημάτων, έχουν καταστήσει πλέον την ναυτιλία τον κύριο κλάδο για εμπορικές συναλλαγές μεταξύ των χωρών και φυσικά έναν από τους βασικότερους κλάδους για την εισροή κερδών για μια χώρα. Ξεκινώντας από τα αρχαία χρόνια, η Αθήνα και η Κόρινθος φαίνεται να ήταν τα κυρίαρχα κέντρα του θαλάσσιου εμπορίου. Με τον καιρό, και με το πέρασμα των αιώνων, ήρθαν στην επιφάνεια και απέκτησαν δύναμη στον χώρο της ναυτιλίας και άλλα κράτη όπως η Ιαπωνία, το Ηνωμένο Βασίλειο, οι Η.Π.Α, η Νότια Κορέα αλλά και η Κίνα. Φυσικά, ο στόλος που άρχισαν να δημιουργούν και να αναπτύσσουν τα κράτη δεν είχαν μόνο ως σκοπό το θαλάσσιο εμπόριο αλλά και την άμυνά τους σε περίοδο πολέμων. Αξίζει να σημειωθεί ότι από την εποχή της βιομηχανικής επανάστασης όπου και έγιναν κοινωνικοοικονομικές και γενικά ριζικές αλλαγές στην καθημερινότητα των ανθρώπων, η ναυτιλία αναπτύχθηκε ραγδαία λόγω και της κατασκευής ατμόπλοιων. Αυτό το γεγονός έδωσε την ευκαιρία στην Ευρώπη να είναι κυρίαρχη στον χώρο της ναυτιλίας, ενώ η μεγαλύτερη δύναμη μέχρι και τα τέλη του 19^{ου} αιώνα θεωρούταν το Ηνωμένο Βασίλειο καθώς διέθετε σχεδόν το μισό του παγκόσμιου στόλου. Διαχρονικά υπήρχαν σημαντικές απώλειες για τον στόλο πολλών κρατών συμπεριλαμβανομένου και της Ελλάδας, λόγω διαφόρων κρίσεων όπως της μεγάλης ύφεσης του 1930 και εξαιτίας φυσικά των παγκοσμίων πολέμων που έγιναν. Το θαλάσσιο εμπόριο και οι θαλάσσιες μεταφορές κατ' επέκταση έμελλαν να γίνουν αναπόσπαστο κομμάτι των περισσοτέρων αναπτυσσόμενων και ήδη ανεπτυγμένων χωρών στον κόσμο. Πλέον σχεδόν το 90% των εξαγωγών πραγματοποιείται μέσω θαλάσσης. Συνεπώς καταλαβαίνει κανείς το πόσο σημαντική χρίζεται για την οικονομία κάθε χώρας και γενικά για την παγκόσμια οικονομία η ναυτιλία. Όπως ήταν αναμενόμενο, θα έπρεπε να συνυπάρχουν όλα τα έθνη στην θάλασσα

μέσα σε ένα πλαίσιο διεθνούς δικαίου και για τον λόγο αυτό, έχουν υπογραφτεί παρόμοιες συμφωνίες και νομοθεσίες με σκοπό την ειρηνική και χωρίς προβλήματα διέλευση των πλοίων από τα διάφορα διεθνή ύδατα και χερσονήσους του πλανήτη. Οι πιο σημαντικές νομοθεσίες στον χώρο της ναυτιλίας θα αναφερθούν σε επόμενη υπό ενότητα.

2.2 Νομοθεσίες και κανονισμοί στην ναυτιλία

Η αναγκαιότητα να συνυπάρχουν και να είναι συμφιλιωμένοι οι λαοί οδήγησε στην θέσπιση νόμων και κανόνων μεταξύ των κρατών. Έτσι λοιπόν και στην ναυτιλία, για να διεκπεραιώνονται ειρηνικά μέσα σε ένα πλαίσιο ευγενούς άμιλλας και με όσο το δυνατόν περισσότερη ασφάλεια τα δρομολόγια των πλοίων, θεσπίστηκαν κάποιες νομοθεσίες. Οι νομοθεσίες αυτές αποτελούν κομμάτι του ναυτικού δικαίου το οποίο στην ουσία είναι εκείνο που διασφαλίζει την ορθή και συναινετική διεξαγωγή του θαλάσσιου εμπορίου ανάμεσα στα έθνη. Αυτή η ορθή διεξαγωγή των θαλάσσιων μεταφορών μεταξύ των χωρών και η ειρηνική πλεύση των πλοίων σε διεθνή ύδατα επιτυγχάνεται πρωτίστως όταν τα πλοία και οι υπεύθυνοι αυτών, ακολουθούν και τηρούν πιστά τις θεμελιώδες αρχές και τους κανόνες που έχουν συμφωνηθεί. Οι σημαντικότεροι από αυτούς τους κανόνες και αρχές αφορούν τα έγγραφα που πρέπει να διαθέτουν τα πλοία κατά την διάρκεια των ταξιδιών, την νηολόγηση τους, καθώς και την σημαία που φέρουν.

Σε κάθε ταξίδι τα πλοία πρέπει να διαθέτουν κάποια έγγραφα και βιβλία τα οποία χρησιμοποιούνται σε διάφορους ελέγχους που γίνονται είτε σε εγχώρια ύδατα είτε σε διεθνή. Τέτοιοι έλεγχοι αφορούν συνήθως τις δραστηριότητές του καραβιού, την ταυτότητά του, το κατά πόσο ασφαλές είναι και στοιχεία σχετικά με το εμπόρευμα που μεταφέρεται. Τα βιβλία και τα έγγραφα αυτά έχουν μια ειδική ονομασία η οποία είναι «Ναυτιλιακά έγγραφα του πλοίου». Σύμφωνα με το άρθρο 46 του Κώδικα Διεθνούς Ναυτικού Δικαίου τα ναυτιλιακά έγγραφα του πλοίου είναι τα: (Τα ναυτιλιακά έγγραφα του πλοίου)

1. **Έγγραφο εθνικότητας**
2. **Ναυτολόγιο**
3. **Πιστοποιητικό ασφάλειας**
4. **Πιστοποιητικό καταμετρήσεως**
5. **Πιστοποιητικό γραμμής φόρτωσης**
6. **Ποινολόγιο**
7. **Ημερολόγιο Μηχανής**
8. **Ημερολόγιο γέφυρας**
9. **Ημερολόγιο ασυρμάτου**
10. **Βιβλίο πετρελαίου**
11. **Πιστοποιητικό εξαρτισμού**
12. **Φορτωτικά έγγραφα**

Άλλα σημαντικά έγγραφα που οφείλουν να έχουν διαθέσιμα τα πλοία με βάση την ναυτιλιακή πρακτική και τακτική είναι τα ακόλουθα:

- **Πιστοποιητικό μυοκτονίας**
- **Πιστοποιητικό υγειονομικής κατάστασης**

- **Πιστοποιητικό Υποθηκών**
- **Πιστοποιητικό πυρασφάλειας**
- **Πιστοποιητικό τελών διέλευσης Διωρύγων**
- **Ασφαλιστήριο κινδύνου ρύπανσης**
- **Διατακτική φορτίου**
- **Ναυλοσύμφωνο**
- **Άδεια απόπλου**

Όσον αφορά την νηολόγηση του πλοίου, είναι απαραίτητο να εγγράφονται τα πλοία στα νηολόγια τα οποία δεν είναι τίποτα άλλο από δημόσια βιβλία τα οποία τηρούνται από τα λιμεναρχεία. Τα νηολόγια περιέχουν τα μητρώα των πλοίων και μπορεί όποιος ενδιαφέρεται να έχει πρόσβαση σε αυτά. Το περιεχόμενό τους περιέχει το όνομα του πλοιοκτήτη καθώς και την ιθαγένειά του, το όνομα του πλοίου, τον τίτλο κτήσης και τις μεταβολές κυριότητας του πλοίου, την χωρητικότητα του πλοίου καθώς και άλλα εμπράγματα δικαιώματα. Για την νηολόγηση ενός πλοίου πρέπει να κατατεθούν στον υπεύθυνο τήρησης των νηολογίων, τον νηολόγο δηλαδή, τα ακόλουθα δικαιολογητικά:

1. **Αίτηση όπου ζητείται η νηολόγηση του πλοίου**
2. **Τίτλος κτήσεων κυριότητας**
3. **Πιστοποιητικό καταμετρήσεως**
4. **Πιστοποιητικό ιθαγένειας ιδιοκτητών**
5. **Δήλωση ιδιοκτησίας**
6. **Πιστοποιητικό διαγραφής**
7. **Έγγραφα από εκτελωνισμό του πλοίου**

Όταν ολοκληρωθεί η νηολόγηση του πλοίου χορηγείται ύστερα στο πλοίο, το λεγόμενο έγγραφο εθνικότητας το οποίο αντιπροσωπεύει ουσιαστικά τον τίτλο κυριότητάς του. Σχετικά με το έγγραφο εθνικότητας, συντελεί ένα από τα πιο σημαντικά ναυτιλιακά έγγραφα ενός πλοίου. Περιλαμβάνει χαρακτηριστικά του πλοίου όπως το όνομά του, το διεθνές διακριτικό σήμα και την χωρητικότητά του, καθώς και το όνομα και την κατοικία του πλοιοκτήτη. Επιπλέον, περιλαμβάνει και άλλα χαρακτηριστικά του πλοίου όπως η ιπποδύναμή της μηχανής του και το υλικό κατασκευής του.

Γενικά η εθνικότητα του πλοίου συνδέεται μοναδικά με την σημαία που φέρει επάνω του. Με βάση το διεθνές δίκαιο όταν τα καράβια βρίσκονται σε διεθνή ύδατα, η σημαία που φέρουν, δηλώνει την επέκταση του εδάφους της αντίστοιχης χώρας. Είναι λοιπόν σαφές ότι όταν τα πλοία πλέουν σε ανοικτή θάλασσα πρέπει να έχουν υψωμένη την σημαία της πολιτείας στην οποία είναι νηολογημένα αλλιώς υπονοείται ότι δεν συμβαδίζουν με γνώμονα το διεθνές και ναυτικό δίκαιο και θεωρούνται άνομα.

2.3 Τα είδη των πλοίων αναλόγως το είδος μεταφοράς

Η ραγδαία εξέλιξη της ναυτιλίας, η αυξημένη ποσότητα και το είδος των φορτίων που μεταφέρονται, επέφεραν αλλαγές στην κατασκευή των πλοίων. Παλιότερα δεν υπήρχαν κατηγορίες πλοίων αναλόγως του τύπου εμπορεύματος. Στα δεύτερα μισά του 20^{ου} αιώνα κρίθηκε αναγκαία, λόγω και των αυξημένων αναγκών και του διογκωμένου όγκου φορτίων, η

κατασκευή και η κατηγοριοποίηση των πλοίων ανάλογα με κάποια χαρακτηριστικά. Τα βασικά χαρακτηριστικά αυτά, με τα οποία έγινε η κατηγοριοποίηση των πλοίων ήταν το είδος μεταφοράς και προορισμού. (E nautilia; Κατάταξη πλοίων)

- Στην πρώτη κατηγορία ανήκουν τα φορτηγά πλοία (***Cargo Ships***) τα οποία διαχωρίζονται σε φορτηγά πλοία ξηρών φορτίων, σε φορτηγά πλοία υγρών φορτίων και σε φορτηγά συνδυασμένων μεταφορών. Τα πλοία ξηρών φορτιών μεταφέρουν χύμα ομοειδή φορτία (***Bulk Carrier***) όπως ζάχαρη και κάρβουνο. Τα πλοία υγρών φορτίων που είναι τα δεξαμενόπλοια (***Tankers***) ανάλογα με την κατασκευή τους μπορούν να μεταφέρουν πετρέλαιο, βενζίνη και υγροποιημένο φυσικό αέριο.
- Στην δεύτερη κατηγορία ανήκουν τα πλοία για ειδικούς προορισμούς που εξελίχθηκαν κυρίως λόγω της ιλιγγιώδους εξέλιξης της τεχνολογίας όπως τα αλιευτικά, τα ωκεανογραφικά και τα εκπαιδευτικά.
- Η Τρίτη κατηγορία είναι εκείνη των επιβατηγών πλοίων (***passenger ships***). Τα επιβατηγά πλοία κατηγοριοποιούνται σε υπερωκεάνια τα οποία υλοποιούν πολύ μακρινά ταξίδια, σε επιβατικά κλειστών θαλασσών, καθώς και σε ακτοπλοϊκά τα οποία κάνουν ταξίδια θαλάσσης εντός των εγχώριων συνόρων.
- Η τέταρτη και τελευταία κατηγορία είναι να πλοία βοηθητικής ναυτιλίας (***Auxiliary ships***) τα οποία βοηθούν τα υπόλοιπα πλοία ώστε να υλοποιήσουν με ασφάλεια τα είτε κοντινά είτε μακρινά ταξίδια τους. Τέτοιου είδους πλοία χαρακτηρίζονται τα ρυμουλκά (***Tug boats***), τα φαρόπλοια (***Light vessels***) και τα ναυαγοσωστικά (***Salvage boats***).

2.4 Τα είδη φορτίων που μεταφέρονται, η προστασία τους και παράγοντες που επηρεάζουν την προστασία τους

Τα πλοία όπως αναφέρθηκε και σε προηγούμενη υπό ενότητα χωρίζονται σε κατηγορίες ανάλογα με το είδος κατασκευής τους και το είδος εμπορεύματος που μεταφέρουν. Γενικά πέρα από τα χύδην φορτία στα οποία αναφερθήκαμε και προηγουμένως τα οποία μεταφέρουν τα φορτηγά πλοία, υπάρχουν και άλλα είδη φορτίων που μεταφέρονται σε καθημερινή βάση από τους διάφορους τύπους πλοίων και αξίζει να αναφερθούν. (Μεταφορά Φορτίων; Κατάταξη πλοίων) Τα φορτία που μεταφέρει λοιπόν ένα πλοίο μπορεί να είναι:

- **Χύδην φορτία:** Ονομάζονται τα φορτία τα οποία στην ουσία μεταφέρονται έτσι όπως είναι χωρίς να είναι συσκευασμένα και διακρίνονται σε χύδην ξηρά φορτία και χύδην υγρά φορτία. Τέτοια ξηρά φορτία είναι οι γαιάνθρακες, τα μεταλλεύματα, τα δημητριακά, τα σιτηρά και η ζάχαρη. Τα υγρά φορτία που μεταφέρονται είναι κυρίως το πετρέλαιο καθώς και παράγωγα αυτού. Πραγματοποιούνται μεταφορές αερίων πετρελαίου όπως το βουτάνιο καθώς και φυσικών αερίων όπως το μεθάνιο. Φυσικά για να γίνει η μεταφορά αερίων πετρελαίου και φυσικών αερίων απαιτείται πρώτα η υγροποίησή τους.
- **Γενικά φορτία:** Κυρίως αφορούν τα γεωργικά και τα βιομηχανικά προϊόντα που μεταφέρονται μέσα σε συσκευασίες
- **Χημικά προϊόντα:** Τα χημικά προϊόντα χρίζονται ειδικής μεταχείρισης καθώς απαιτούν εξειδικευμένο πλήρωμα, το πλοίο που τα μεταφέρει πρέπει να έχει κατάλληλα διαμορφωμένες εγκαταστάσεις και να πληρούνται όλοι οι κανόνες ασφαλείας για να

αποφευχθούν ατυχήματα που μπορεί να προκαλέσουν πέρα από τραυματισμούς και καταστροφές στο πλοίο, μόλυνση και στο περιβάλλον. Το πιο σύνηθες χημικό προϊόν που μεταφέρεται είναι το υγρό θείο. Τέτοιου είδους φορτία μεταφέρουν τα ειδικώς εξοπλισμένα χημικά δεξαμενόπλοια (***chemical tankers***).

- **Ευπαθή φορτία:** Είναι κατά κύριο λόγο τροφές όπως λαχανικά, κρέατα και φρούτα τα οποία πρέπει να μεταφερθούν σε μακρινές περιοχές διασχίζοντας πολλά ναυτικά μίλια. Αυτό έχει σαν αποτέλεσμα να αλλοιώνεται η ποιότητά τους εξαιτίας των συνθηκών που επικρατούν. Για την ασφάλειά τους λοιπόν, μεταφέρονται με ειδικά, εξειδικευμένα πλοία τα οποία αναλόγως και το είδος του ευπαθούς φορτίου διαχωρίζονται για να διατηρούν αντίστοιχες συνθήκες συντήρησης. Έτσι διασφαλίζεται η ποιότητά τους.
- **Οχήματα:** Πολλές φορές είναι αναγκαία η μεταφορά οχημάτων από περιοχή σε περιοχή. Τις περισσότερες φορές τα πλοία που μεταφέρουν οχήματα είναι κλειστού τύπου, δηλαδή τα οχήματα προστατεύονται με κατάστρωμα. Υπάρχουν και τα πλοία ανοικτού τύπου που μεταφέρουν οχήματα, όμως πραγματοποιούν πολύ κοντινά ταξίδια και τα οχήματα δεν έχουν την προστασία καταστρώματος.

Κάθε είδος φορτίου θέλει ειδική μεταχείριση και προσοχή έτσι ώστε να μην υπάρξουν ζημιές και ατυχήματα. Από την στιγμή φόρτωσης των εμπορευμάτων μέχρι και την εκφόρτωσή τους πρέπει να τηρούνται πιστά όλοι οι κανονισμοί ασφαλείας. Πολλά ατυχήματα και αλλοιώσεις στην ποιότητα των φορτίων συμβαίνουν γιατί γίνονται λανθασμένα, πρόχειρα και άκομψα τόσο η φόρτωση των προϊόντων όσο και η εκφόρτωσή τους. Οι υπεύθυνοι για τα εμπορεύματα κατά την τοποθέτησή των εμπορευμάτων στο πλοίο, πολλές φορές εξαιτίας των αδέξιων χειρισμών κατανέμουν ανορθόδοξα το φορτίο στο πλοίο με αποτέλεσμα κατά την διάρκεια του ταξιδιού τα φορτία να επιδέχονται ζημιές. Συνεπώς για την προστασία των εμπορευμάτων και του ίδιου του πλοίου, είναι αναγκαίο καθ' όλη την διάρκεια του ταξιδιού, ανά τακτά χρονικά διαστήματα, να ελέγχεται το πλοίο αν έχει υποστεί παραμορφώσεις ή αν δεν είναι ζυγοσταθμισμένο δηλαδή να εξετάζεται αν παρουσιάζει κάποια ασυνήθιστη κλίση εξαιτίας του όγκου του φορτίου και της ορθής ή μη τοποθέτησής του. Με αυτόν τον τρόπο πιθανόν να εντοπιστούν άμεσα τυχόν λανθασμένοι χειρισμοί κατά την φόρτωση των φορτίων και να διορθωθούν πριν επέλθουν ζημιές. Μπορούμε να πούμε και συνοπτικά ότι οι παράγοντες που επηρεάζουν την προστασία των φορτίων και την ασφαλή μεταφορά τους είναι:

- η ασφάλιση των φορτίων και ο τρόπος που τοποθετούνται στο πλοίο
- η αντοχή του πλοίου και η κατάλληλη ζυγοστάθμισή του
- η εκ των προτέρων κατάλληλη προετοιμασία του πλοίου για την φόρτωση των εμπορευμάτων για το ταξίδι
- η εμπειρία και η επιδεξιότητα του πληρώματος του εκάστοτε πλοίου

2.5 Η ναυλαγορά και η διαφοροποίησή της

Η ναυλαγορά αναφέρεται στο πως ορίζεται η αξία των ναύλων στα ταξίδια. Στο θαλάσσιο εμπόριο, ο ναύλος αντιπροσωπεύει το χρηματικό ποσό που ορίζει ο μεταφορέας για να υλοποιήσει μια μεταφορά είτε φορτίων είτε ατόμων. Τον ναύλο και συγκεκριμένα την αξία του, φαίνεται πως την επηρεάζουν πολλοί παράγοντες όπως η περιοχή που λαμβάνει χώρα η

μεταφορά των φορτίων καθώς και τα πρόσωπα που συμμετέχουν κατά μία έννοια στο ταξίδι όπως νομικά πρόσωπα ή φυσικά πρόσωπα. Ανάλογα λοιπόν, με τα συμφέροντά τους αυτά τα άτομα διαμορφώνουν την τελική αξία του ναύλου. Ακόμη ένας παράγοντας που διαμορφώνει την τελική αξία των ναύλων είναι και το είδος φορτίου που μεταφέρεται, δηλαδή οι ναύλοι και εν γένει η ναυλαγορά διαφοροποιείται ως προς το επίπεδο τιμών των ναύλων. Συνεπώς η διαφοροποίηση της ναυλαγοράς μπορεί να γίνει ως προς τους επόμενους παράγοντες:

- Ανάλογα με τον τύπο του διακινούμενου φορτίου, δηλαδή διαφορετική τιμή έχει ο ναύλος όταν μεταφέρονται για παράδειγμα χύδην ξηρά φορτία όπως τα δημητριακά και ο άνθρακας και άλλη τιμή έχει ο ναύλος όταν μεταφέρονται χύδην υγρά φορτία όπως το πετρέλαιο.
- Ανάλογα με το είδος του πλοίου που χρησιμοποιείται, δηλαδή ο ναύλος μπορεί να διαφέρει όταν χρησιμοποιούνται πλοία ξηρών φορτίων (bulk carriers) για μεταφορά χύδην ξηρών φορτίων και όταν χρησιμοποιούνται πλοία τύπου Tankers για μεταφορά υγρών φορτίων.
- Αναλόγως την γεωγραφική περιοχή που υλοποιεί τις μεταφορές το πλοίο. Κάθε περιοχή υπόκειται σε διαφορετικούς κανονισμούς και νομοθεσίες οπότε η είσοδος και η πλεύση στα διάφορα μέρη του πλανήτη, αποτελούν ξεχωριστά κομμάτια της ναυλαγοράς.
- Ανάλογα με την μορφή ναυλώσεως που διαλέγεται. Υπάρχουν διάφοροι τύποι ναυλώσεως στην αγορά όπου ο καθένας ξεχωριστά διαμορφώνεται από ξεχωριστούς όρους και συμφωνίες. Γενικά οι τύποι ναυλώσεως διαχωρίζονται σε αγορά ναυλώσεων ταξιδιού (**voyage charter market**), σε αγορά χρονοναυλώσεων (**time charter market**), σε αγορά μισθώσεων γυμνού πλοίου (**bare boat charter market**) και τέλος σε αγορά συμβολαίων εργολαβικής μεταφοράς (**contracts of affreightment market**).

2.6 Βασικά χαρακτηριστικά και ιδιότητες των πλοίων που μεταφέρουν χύδην ξερά φορτία (Bulk Carriers)

Τα πλοία που μεταφέρουν χύδην φορτία (Buk Carriers), επειδή οι συνθήκες της παγκόσμιας αγοράς συνεχώς άλλαζαν όπως και συνεχώς αλλάζουν με το πέρασμα των χρόνων, και επειδή εμφανίστηκε στο διεθνές εμπόριο η ανάγκη για την μεταφορά μεγαλύτερου όγκου εμπορεύματος, χρειάστηκε να εξελιχθούν και αναβαθμιστούν. Αυτή η εξέλιξη και αναβάθμισή τους αφορά την ασφάλεια μεταφοράς των φορτίων, την μέγιστη χωρητικότητα που διαθέτουν καθώς την ανθεκτικότητά τους όσον αφορά τον όγκο του φορτίου που μπορούν να μεταφέρουν χωρίς να υπάρχουν κίνδυνοι και φυσικά ζημιές. Πλέον λοιπόν, ανάλογα με την χωρητικότητα τους, τα πλοία μεταφοράς χύδην φορτίων έχουν διαχωριστεί σε κάποιες κατηγορίες. Οι κατηγορίες αυτές είναι οι εξής (Πλοία μεταφοράς χύδην ξηρών φορτίων):

- **Mini Handy**: Πρόκειται για πλοία μεταφοράς χύδην φορτίου τα οποία έχουν χωρητικότητα μεταξύ λιγότερη από 10000 dwt (deadweight).

- **Handysize**: Η χωρητικότητα που διαθέτουν αυτά τα πλοία κυμαίνεται μεταξύ 10000 και 35000 dwt.

- **Handymax**: Τα συγκεκριμένα bulk carriers διαθέτουν χωρητικότητα μεταξύ 35000 και 50000 dwt. Η συγκεκριμένη κατηγορία έχει μειωμένο κόστος κτήσης που σε συνδυασμό με το επίσης αρκετά μειωμένο κόστος συντήρησης καθιστά αυτά τα πλοία να είναι εκείνα που χρησιμοποιούνται ως επί το πλείστων για την μεταφορά των χύδην φορτίων.

- **Supramax**: Είναι ο τύπος πλοίων μεταφοράς χύδην φορτίων που έχουν χωρητικότητα 50000 με 60000 dwt. Τα πέντε κύτη (αμπάρια) καθώς και οι τέσσερις γερανοί με κοντά στους 30 τόνους φορτίου εργασίας, που διαθέτουν αυτά τα πλοία, κάνουν πιο εύκολη την χρήση τους σε διάφορα λιμάνια που έχουν περιορισμένη υποδομή.

- **Panamax**: Αυτά τα πλοία έχουν χωρητικότητα 60000 με 80000 dwt και όπως προκύπτει από το όνομά τους σχεδιάστηκαν με γνώμονα την όσο δυνατόν μεγαλύτερη χωρητικότητα για την ασφαλή πρόσβαση και διέλευση στην διώρυγα του Παναμά. Η ασφαλής διέλευση στην διώρυγα του Παναμά καθιστά πολύ οικονομική την πλεύση τους από τον Ατλαντικό ωκεανό προς τον Ειρηνικό ωκεανό ή και το αντίστροφο.

- **Post Panamax**: Το συγκεκριμένο μέγεθος πλοίων περιλαμβάνει πλοία χωρητικότητας 80000 με 100000 dwt.

- **Capesize**: Τα πλοία αυτά έχουν χωρητικότητα από 100000 dwt μέχρι και 200000 dwt.

- **VLBC (Very Large Bulk Carriers)**: Η μεγαλύτερη κατηγορία πλοίων μεταφοράς χύδην φορτίων είναι αυτή των VLBC τα οποία διαθέτουν πάνω από 200000 dwt μέχρι και 400000 dwt και ειδικεύονται στην μεταφορά σιδηρομεταλλεύματος μόνο. Είναι μια κατηγορία η οποία ήρθε στην επιφάνεια τα τελευταία χρόνια και αυτό λόγω της ταχύτατης ανάπτυξης της σιδηροβιομηχανίας στην Κίνα συγκεκριμένα αλλά και στην Άπω Ανατολή γενικότερα.

Εκτός από τις κατηγορίες τους, τα πλοία Bulk Carriers (Εικόνα 1) έχουν και κάποια κύρια χαρακτηριστικά τα οποία είναι σχετικά με το φορτίο που μεταφέρουν, τον αποθηκευτικό τους χώρο και τον σχεδιασμό τους. Ένα βασικό χαρακτηριστικό αυτού του τύπου πλοίων είναι τα κύτη τους ή με άλλα λόγια τα αμπάρια τους. Για την καλύτερη χωρητικότητα του φορτίου τα πλοία αυτά έχουν τα απαραίτητα και κατάλληλα κύτη. Επιπλέον τα κύτη δεν έχουν υποφράγματα που στην ουσία είναι η οριζόντια υποδιαίρεση στα κύτη τους και κλείνουν με την βοήθεια μηχανικών καλυμμάτων. Ο αριθμός των κυτών σε ένα πλοίο μεταφοράς χύδην φορτίου γενικά διαφοροποιείται αναλόγως το μέγεθός του. Δηλαδή μπορεί ένα μικρό πλοίο bulk carrier να έχει ένα κύτος, ενώ ένα τεράστιο πλοίο μπορεί να διαθέτει δέκα κύτη. Ένα ακόμα σημαντικό χαρακτηριστικό των συγκεκριμένων πλοίων είναι οι δεξαμενές με τις οποίες είναι εφοδιασμένα. Πιο συγκεκριμένα, έχουν δεξαμενές έρματος, και στους λεγόμενους χώρους χοάνης ή με την αγγλική ορολογία (hoppers), δίπλα και κάτω από την βάση. Οι δεξαμενές χοάνης και οι αντίστοιχες κορυφής, συνεισφέρουν στην βελτίωση του χειρισμού των φορτίων και συμβάλλουν σημαντικά στην μείωση του κόστους μεταφοράς.

Εικόνα 1: Πλοίο μεταφοράς χύδην φορτίων (Bulk Carrier ship)
Πηγή εικόνας: (https://www.intercargo.org/topics/cargoes-liquefaction/)

2.7 Η ναυτιλία σε παγκόσμιο επίπεδο

Η ναυτιλία παγκοσμίως πλέον, θεωρείται ίσως ο πιο σημαντικός κλάδος αν όχι σίγουρα ο πιο σημαντικός, για την οικονομική ευημερία και ευρυθμία των λαών καθώς το 90% περίπου των μεταφορών πραγματοποιείται μέσω θαλάσσης. Για να μπορούν όμως τα κράτη μεταξύ τους, να κάνουν εμπορικές συναλλαγές μέσω της θάλασσας έπρεπε να υπάρχουν κάποια θαλάσσια δίκτυα μέσω των οποίων θα μεταφέρονταν όσο το δυνατόν περισσότερα φορτία από την μία άκρη της γης ως την άλλη και θα συνδέονταν πιο άμεσα κάποιες περιοχές. Στον χώρο της ναυτιλίας λοιπόν, υπάρχουν κάποια ευρέως γνωστά θαλάσσια δίκτυα που δημιουργήθηκαν είτε στα τέλη του 19^{ου} αιώνα είτε στις αρχές του 20^{ου} αιώνα, που με την βοήθειά τους, διευκόλυναν το εμπόριο και την μεταφορά δισεκατομμυρίων τόνων φορτίων σε καθημερινή βάση (E nautilia). Το πρώτο από τα πιο σημαντικά περάσματα για το θαλάσσιο εμπόριο είναι η Διώρυγα του Σουέζ της οποία η διάνοιξη ολοκληρώθηκε το 1869. Η διώρυγα του Σουέζ είναι υψίστης σημασίας διότι δημιουργήθηκε μέσω αυτής ένα δίκτυο θαλάσσιων μεταφορών μεταξύ της Μεσογείου και της Ερυθράς θάλασσας. Στοιχεία αναφέρουν πως κάθε μέρα διέρχονται από την διώρυγα πάνω από 100 πλοία και η αξία των φορτίων που μεταφέρουν ανέρχεται περίπου στα 9,6 δισεκατομμύρια δολάρια. Το 2020 σύμφωνα με έρευνα, διέσχισαν την διώρυγα του Σουέζ πάνω από 18000 με 19000 πλοία μεταφέροντας συνολικά περίπου το 12% του πετρελαίου παγκοσμίως. Βέβαια έχει υποστεί αρκετές κρίσεις με την πιο πρόσφατη να είναι τον Μάρτιο του 2021 όπου το φορτηγό πλοίο «Ever Given» είχε φράξει επί έξι μέρες την διώρυγα. Ως συνέπεια το γεγονός αυτό είχε, να χάνονται καθημερινά 10 δισεκατομμύρια δολάρια. Σύμφωνα μάλιστα με οικονομικούς αναλυτές αν η διώρυγα είχε μπλοκαριστεί για πάνω από δύο βδομάδες τότε θα υπήρχαν σημαντικές οικονομικές συνέπειες σε παγκόσμιο επίπεδο. Μία ακόμη πολύ σημαντική θαλάσσια οδός είναι αυτή της διώρυγας του Παναμά. Η διώρυγα του Παναμά συνδέει τον Ατλαντικό ωκεανό με τον Ειρηνικό μέσω της Καραϊβικής και με την διάνοιξή της το 1914 διευκολύνθηκε περαιτέρω το θαλάσσιο εμπόριο μεταξύ των χωρών. Ακόμη δύο σπουδαίας σημασίας θαλάσσια δίκτυα που υπάρχουν είναι αυτά του στενού του Βοσπόρου και τα στενά της Δανίας. Χάρη στο στενό του Βοσπόρου συνδέεται η Μαύρη θάλασσα με την Μεσόγειο μέσω της θάλασσας Μαρμαρά. Με αυτόν τον τρόπο επιτυγχάνεται σύνδεση του ευρωπαϊκού θαλάσσιου δικτύου με της Ασίας. Από την άλλη, τα στενά της Δανίας είναι σημαντικά για την ναυτιλία διότι συνδέουν την βόρεια θάλασσα με την Βαλτική θάλασσα. Μέσω των στενών της Δανίας έρχονται δεξαμενόπλοια από

την ανατολική Ευρώπη και κυρίως της Ρωσίας και οδεύουν προς την δύση. Συνεπώς όλες αυτοί οι θαλάσσιες οδοί έχουν συμβάλλει στην ανάπτυξη και διευκόλυνση του θαλάσσιου εμπορίου και στην μεταφορά χιλιάδων φορτίων καθημερινά.
Τα τελευταία χρόνια έχουν τεθεί νέοι στόχοι σε παγκόσμια κλίμακα για την ναυτιλία. Αυτοί οι στόχοι αφορούν την μείωση εκπομπής διοξειδίων του άνθρακα από τα πλοία. Γίνονται εδώ και αρκετά έτη προσπάθειες να βρεθούν εναλλακτικές πηγές ενέργειας και να κατασκευαστούν ενεργειακά αποδοτικά πλοία έτσι ώστε να μεταβεί η ναυτιλία σε ένα καθαρό περιβάλλον χωρίς ρύπους ή όσο το δυνατόν λιγότερους γίνεται. Όμως αυτή η μετάβαση μόνο εύκολη δεν θεωρείται και αναζητούνται ασφαλείς και βιώσιμες λύσεις.
Όσον αφορά την παγκόσμια δύναμη στην ναυτιλία τα ηνία έχει η Ελλάδα καθώς αντιπροσωπεύει το 21% του παγκόσμιου στόλου και το 59% του Ευρωπαϊκού στόλου. Την Ελλάδα ακολουθούν σαν παγκόσμιες δυνάμεις η Ιαπωνία, η Κίνα, η Σιγκαπούρη, το Χονγκ Κονγκ οι οποίες διαθέτουν περισσότερο από το 50% της χωρητικότητας σε τόνους (dwt).

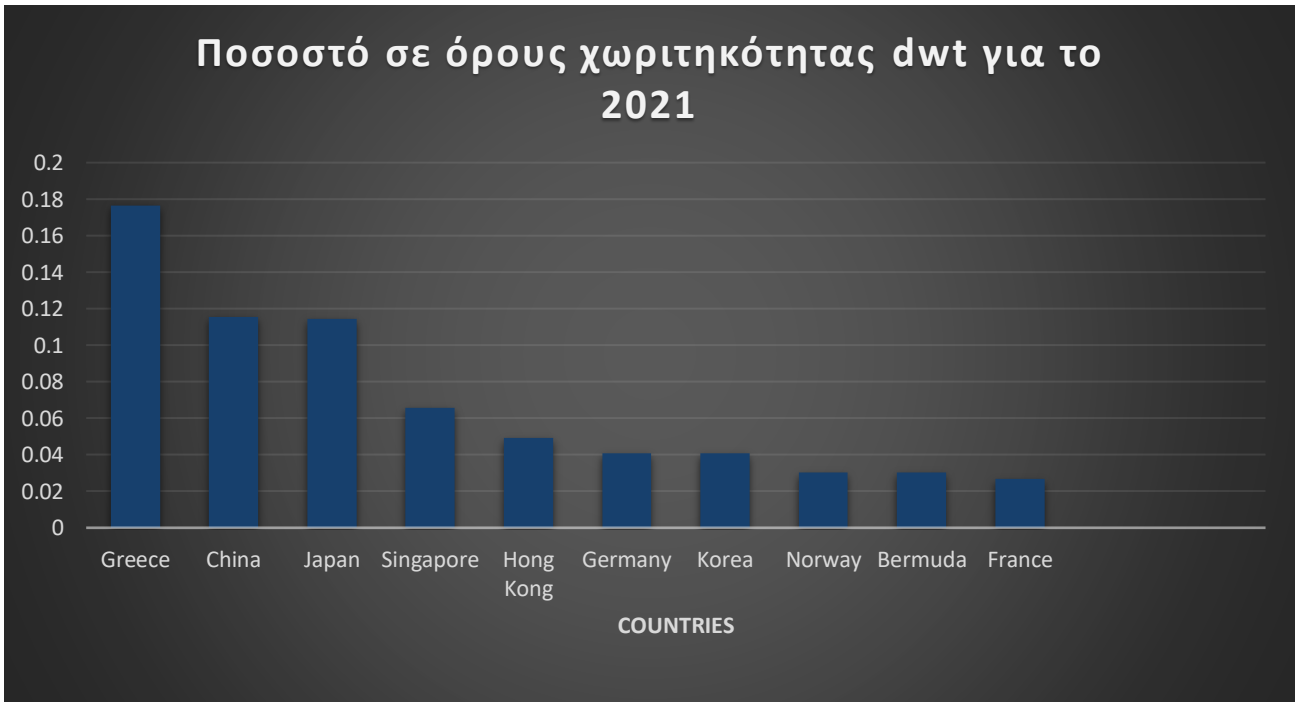
2.8 Το πρόβλημα της εκπομπής ρύπων από τα πλοία και η αντιμετώπισή του από την βιομηχανία της Ναυτιλίας

Η ναυτιλία παρότι είναι ο κλάδος που συμβάλλει σε ποσοστό περίπου 90% στην μεταφορά του διεθνές εμπορίου, συμβάλλει εξίσου σημαντικά στο αρνητικό και επικίνδυνο φαινόμενο του θερμοκηπίου εκπέμποντας σχεδόν το 3% των παγκοσμίων ρύπων διοξειδίου του άνθρακα. Το φαινόμενο του θερμοκηπίου επειδή συνεχώς με τα χρόνια διογκώνεται, αυτό έχει σαν αποτέλεσμα να γίνει αισθητή η κλιματική αλλαγή παγκοσμίως. Επειδή η ναυτιλία σαν κλάδος έχει γνωρίσει τεράστια και ραγδαία ανάπτυξη με την πάροδο των χρόνων αλλά και επειδή συνεχώς εξελίσσεται, μέσα σε λίγα χρόνια εάν δεν παρθούν αυστηρά και κατάλληλα μέτρα για την μείωση των εκπομπών ρύπων από τα πλοία τότε, το ποσοστό εκπομπής ρύπων ίσως και να τριπλασιαστεί σε μερικές δεκαετίες. Οι θαλάσσιες εκπομπές ρύπων αυξάνονται πολύ γρήγορα και είναι πολύ πιο έντονες σε σχέση με άλλους τομείς σύμφωνα με τον Syed-Asif Ansar ο οποίος είναι επιστήμονας στο γερμανικό αεροδιαστημικό κέντρο «DLR» (European Commission). Η πλειοψηφία των πλοίων χρησιμοποιούν κινητήρες diesel για να παράγουν ενέργεια με σκοπό να εκτελέσουν τα δρομολόγιά τους όμως αυτή η χρήση diesel κινητήρων οδηγεί και στην εκπομπή πολλών ρύπων. Σκοπός λοιπόν της ναυτιλίας και πρωταρχικός στόχος του Διεθνή Ναυτιλιακού Οργανισμού (IMO) είναι να βρεθούν νέες πηγές καυσίμου για τα πλοία, με άλλα λόγια δηλαδή, να οδηγηθεί η ναυτιλία σε νέα, μεταβατικά και καθαρότερα καύσιμα. Ακόμη ένας στόχος για τον οργανισμό «IMO» είναι να μειωθούν οι εκπομπές ρύπων διοξειδίου του άνθρακα από τα πλοία στο μισό, μέχρι το 2050. Γενικά έχουν προταθεί αρκετές εναλλακτικές λύσεις ώστε να στραφεί ο ναυτιλιακός τομέας σε άλλες πηγές καυσίμου όπως το υγροποιημένο φυσικό αέριο LNG το οποίο η Ευρωπαϊκή Ένωση το ταξινομεί ως μεταβατικό καύσιμο που πιθανόν να βοηθήσει την μετάβαση σε ανανεώσιμες πηγές ενέργειας. Μάλιστα ήδη σε μία έρευνα με όνομα «Nautilus» που ξεκίνησε το 2020, γίνεται προσπάθεια να κατασκευαστεί ένας νέος τύπος κινητήρα με βάση το υγροποιημένο φυσικό αέριο το οποίο θα οδηγούσε ενδεχομένως σε μείωση των επιβλαβών ρύπων στο μισό. Φυσικά η πρόκληση αυτή για την εύρεση νέων ανανεώσιμων πηγών ενέργειας είναι τεράστια και ο ναυτιλιακός κλάδος ήδη όπως έχει σημειωθεί σε προηγούμενο κεφάλαιο προσπαθεί αρκετά χρόνια για την εύρεση και ανάπτυξη και έλεγχο τέτοιων νέων, μεταβατικών και καθαρών από επιβλαβείς ρύπους καυσίμων.

2.9 Η ναυτιλία στην Ελλάδα με την πάροδο των χρόνων

Όπως αναφέρθηκε και προηγουμένως, η Ελλάδα αποτελεί την κυρίαρχη δύναμη στην ναυτιλία, ανεξαρτήτου σημαίας. Όμως αυτή η δύναμη και η κυριαρχία της Ελλάδας σε παγκόσμιο επίπεδο, αν κοιτάξει κάποιος το παρελθόν, θα αντιληφθεί πως μόνο εύκολο δεν ήταν να επιτευχθεί καθώς υπήρξαν αρκετές κρίσεις και πόλεμοι.
Η Ελλάδα διέθετε πάντοτε από τα αρχαία χρόνια έναν δυνατό στόλο. Στην σύγχρονη ιστορία όμως, οι πιο δύσκολες στιγμές για τους Έλληνες επί τουρκοκρατίας είχαν σαν αποτέλεσμα να δυσχεράνουν και την ελληνική ναυτιλία και το θαλάσσιο εμπόριο κατ' επέκταση. Από τα τέλη όμως του 18ου αιώνα και με την συνθήκη «Κιουτσούκ – Καϊναρτζή» (Ναυτικό μουσείο Αιγαίου) η οποία έδωσε το δικαίωμα στους Έλληνες υπηκόους της Οθωμανικής αυτοκρατορίας να φέρουν στα πλοία τους την Ρωσική σημαία, ανέτρεψε την εις βάρος κατάσταση της ελληνικής ναυτιλίας, ο εμπορικός στόλος των Ελλήνων γνώρισε μεγάλα άλματα και θεαματική εξέλιξη και έδωσε την ευκαιρία στα ελληνικά πλοία να συνεχίσουν ελεύθερα τις εμπορικές τους συναλλαγές μέσω θάλασσας κυρίως στον Εύξεινο Πόντο. Φτάνοντας στα τέλη του 19ου αιώνα και στις αρχές του 20ου η Ελλάδα φαίνεται να είναι μια μεγάλη δύναμη στον ναυτιλιακό τομέα με ισχυρότερη δύναμη τότε το Ηνωμένο Βασίλειο. Ενώ όλα κυλούσαν ομαλά και άνθιζε και εξελισσόταν με ιλιγγιώδεις ταχύτητες η ελληνική ναυτιλία, πραγματοποιείται ο Α' Παγκόσμιος πόλεμος. Με το τέλος του Α' Παγκοσμίου πολέμου η Ελλάδα είχε σημαντικές απώλειες, καθώς της είχαν απομείνει τα μισά περίπου ατμόπλοια από τα 400 περίπου που είχε στην διάθεσή της. Οι Έλληνες πλοιοκτήτες όμως, δουλεύοντας σκληρά, άρχισαν εμπορικές συναλλαγές και εκτός της μεσογείου με αποτέλεσμα ο στόλος της Ελλάδας να αναπτυχθεί και να ισχυροποιηθεί. Έπειτα, λαμβάνει χώρα, ο Β' παγκόσμιος πόλεμος όπου η Ελλάδα και συγκεκριμένα η ναυτιλία της θα υποστεί τρομερές απώλειες που την οδήγησαν σχεδόν στο να διαλυθεί. Ενώ λοιπόν, δεν υπήρχε φως στον ορίζοντα για τον στόλο της Ελλάδας, την λύση έρχονται να δώσουν οι Η.Π.Α. Χάρη στις Η.Π.Α και τον εφεδρικό στόλο που διέθεταν, οι Έλληνες πλοιοκτήτες, σύμφωνα με ιστορικά στοιχεία, αγόρασαν 100 πλοία τύπου «Liberty» και με αυτόν τον τρόπο ξεκίνησε και πάλι η ελληνική ναυτιλία να ανθίζει. Η χρυσή εποχή για την ελληνική ναυτιλία όπου και εκτοξεύθηκε η δύναμή της ήταν η δεκαετία του 60'. Οι Έλληνες πλοιοκτήτες συνεχώς αγόραζαν πλοία με αποτέλεσμα να αυξάνεται όλο ένα και περισσότερο ο ελληνικός στόλος. Μάλιστα, από εκείνη την εποχή, το λιμάνι του Πειραιά άρχισε να θεωρείται ένα από τα πιο μεγάλα και σημαντικά λιμάνια του κόσμου. Τα ελληνικά πλοία και περνώντας τα χρόνια, συνέχισαν να διασχίζουν τις θάλασσες και να κυριαρχούν και στον Ειρηνικό ωκεανό και να μεταφέρουν διάφορα φορτία προς τις υπόλοιπες χώρες. Φτάνοντας στο σήμερα, η ναυτιλία της Ελλάδας συνεχίζει παρά τις όποιες ενδιάμεσες κρίσεις να είναι σταθερά κυρίαρχη παγκοσμίως. Στοιχεία του 2021 από την «UNCTAD» (UNCTAD) και την ετήσια ανασκόπησή της (Σχήμα 1), δείχνουν ότι για το 2021 η Ελλάδα αντιπροσωπεύει περίπου το 17% της χωρητικότητας σε τόνους (dwt). Μετά από σχεδόν έναν χρόνο και πιο συγκεκριμένα, σύμφωνα με στοιχεία της Ετήσιας έκθεσης Ελλήνων εφοπλιστών για την περίοδο 2022 (Ένωση Εφοπλιστών Ελλάδος), μέχρι στιγμής, ο ελληνικός στόλος διαθέτει 5514 πλοία ενώ οι Έλληνες πλοιοκτήτες ελέγχουν το 21% περίπου της παγκόσμιας

χωρητικότητας σε τόνους (dwt). Επίσης από τα επίσημα στοιχεία της Ένωσης Ελλήνων Εφοπλιστών προκύπτει ότι οι Έλληνες πλοιοκτήτες ελέγχουν το 25% του παγκόσμιου στόλου μεταφοράς χύδην ξηρού φορτίου και σχεδόν το 32% του παγκόσμιου στόλου πετρελαιοφόρων. Επιπλέον, οι Έλληνες πλοιοκτήτες ελέγχουν περίπου το 22% του παγκόσμιου στόλου μεταφοράς υγροποιημένου φυσικού αερίου (LNG) και το 14% του παγκόσμιου στόλου μεταφοράς υγροποιημένου αερίου πετρελαίου (LPG). Να σημειωθεί ότι το ένα τρίτο του ελληνόκτητου στόλου έχει σημαία κράτους μέλους της Ευρωπαϊκής Ένωσης. Το γεγονός ότι η Ελλάδα είναι το σταυροδρόμι τριών ηπείρων, της Ασίας, της Αφρικής και της Ευρώπης την καθιστά αναμφισβήτητα, βασικό γρανάζι της ναυτιλίας τόσο σε παγκόσμιο επίπεδο όσο και σε ευρωπαϊκό. Σε ευρωπαϊκό επίπεδο, η γεωγραφική θέση της Ελλάδας, την έχει μετατρέψει σε κύρια χώρα εισροών πηγών ενέργειας μέσω θαλάσσης για λογαριασμό της Ευρώπης από τον υπόλοιπο κόσμο. Σε παγκόσμιο επίπεδο δε, η σημασία του ελληνόκτητου στόλου είναι τεράστια, καθώς μεταφέρει πρώτης τάξης αγαθά όπως πετρέλαιο, γεωργικά και δασικά προϊόντα και άνθρακα. Φυσικά η ναυτιλία βοηθά και εντός συνόρων, αφού μαζί με τον τουρισμό αποτελεί την κύρια πηγή εσόδων και εξαγωγών, και συμβάλλει σημαντικά στο ισοζύγιο πληρωμών της χώρας. Αποτελεί μάλιστα, το 6,5% του ΑΕΠ. Επιπροσθέτως, ο κλάδος της ναυτιλίας στην Ελλάδα, δίνει την δυνατότητα σε ανθρώπους να εργαστούν αφού απασχολεί περίπου 290.000 άτομα συνολικά. Οι Έλληνες πλοιοκτήτες από την μεριά τους, ανανεώνουν συνεχώς τον ελληνικό στόλο προχωρώντας σε αγορές πλοίων, σε καιρούς αβεβαιότητας και με καινούργιες προκλήσεις. Τα νέα αυτά πλοία είναι πιο αποδοτικά και φιλικά προς το περιβάλλον μιας και παγκόσμιος στόχος που έχει τεθεί τα τελευταία χρόνια είναι όπως αναφέρθηκε στην προηγούμενη ενότητα, η μείωση εκπομπών διοξειδίου του άνθρακα.



Σχήμα 1
Πηγή: (https://unctad.org/system/files/official-document/rmt2021_en_0.pdf)

2.10 Σημαντικά στοιχεία της ελληνικής Ναυτιλίας

Η ναυτιλία της Ελλάδας σίγουρα διαδραματίζει πολύ σπουδαίο ρόλο για την οικονομία της και την εξέλιξή της. Συγκριτικά, σε σχέση με άλλες χρονιές, ο ελληνόκτητος στόλος έχει βελτιωθεί, ενισχυθεί και ανανεωθεί. Υπάρχουν μερικά άξια αναφοράς στοιχεία και χαρακτηριστικά της ελληνικής ναυτιλίας που εξηγούν κιόλας τον λόγο που η Ελλάδα έχει κτίσει μια γερή ναυτιλιακή βιομηχανία. Ένα βασικό χαρακτηριστικό στοιχείο της ελληνικής ναυτιλίας είναι ο τύπος του διακινούμενου φορτίου που δραστηριοποιείται. Πιο συγκεκριμένα, δραστηριοποιείται κυρίως στην μεταφορά χύδην ξηρών φορτίων (**bulk/tramp**) όπως άνθρακας και σιδηρομεταλλεύματα αλλά και στην μεταφορά πετρελαίου. Το γεγονός ότι η ναυτιλία στην χώρα μας ασχολείται με τον συγκεκριμένο τομέα φορτίων, την κάνει να παρουσιάζει στοιχεία τέλειου ανταγωνισμού αν και στην πράξη τέλειος ανταγωνισμός δεν υφίσταται. Η μορφή ναυλώσεων που επιλέγουν οι Έλληνες ναυλωτές είναι αυτή των ναυλοσυμφώνων κατά τον χρόνο (**Time charter party contracts**). Στατιστικά στοιχεία (Ένωση Εφοπλιστών Ελλάδος) για την περίοδο 2007 με 2019 δείχνουν πως η ναυτιλία στην Ελλάδα παρουσιάζει περαιτέρω ανάπτυξη αφού οι Έλληνες πλοιοκτήτες υπερδιπλασίασαν την δυναμικότητα του στόλου όσον αφορά τις θαλάσσιες μεταφορές. Επίσης για το 2021 σε σύγκριση με το 2014 φαίνεται πως η συνολική χωρητικότητα του ελληνόκτητου στόλου σε όρους dwt, έχει αυξηθεί κατά 45,8% ενώ σε σύγκριση με το 2019 έχει αυξηθεί κατά 7,5%. Δηλαδή ακόμα και σε περίοδο ύφεσης της αγοράς λόγω της πανδημίας του κορονοϊού, η χωρητικότητα αυξήθηκε. Ένα από τα πιο σπουδαία χαρακτηριστικά του ελληνικού στόλου, αν όχι το σπουδαιότερο και σημαντικότερο, είναι η ασφάλειά του. Ο ελληνόκτητος στόλος λοιπόν, θεωρείται ο πιο ασφαλής στόλος παγκοσμίως μιας και μόνο το 0,44% του εμπορικού ελληνικού στόλου και μόλις το 0,5% του στόλου με βάση την χωρητικότητα είχε κάποιο ατύχημα. Εκτός όμως από την ασφάλειά του, ένα επιπλέον θετικό στοιχείο του ελληνόκτητου στόλου είναι αυτό της ηλικίας του. Τα στοιχεία του Μαρτίου του 2022 της «HIS Markit» και τα οποία επεξεργάστηκε το « Greek Shipping Co-operation Committee» (All about shipping) έδειξαν ότι η μέση ηλικία του ελληνόκτητου στόλου είναι 12,33 έτη, ενώ η μέση ηλικία του παγκόσμιου στόλου είναι 14,7 έτη. Στο ίδιο συμπέρασμα σχετικά με την μέση ηλικία του ελληνόκτητου στόλου καταλήγουν και τα στοιχεία της Ένωσης Ελλήνων Εφοπλιστών για το 2022 μέχρι τώρα, καθώς η μέση ηλικία του ελληνόκτητου στόλου σύμφωνα με αυτά τα στοιχεία είναι 9,99 έτη ενώ η μέση ηλικία του παγκόσμιου στόλου είναι 10,28 έτη. Μολονότι η ναυτιλία της Ελλάδας είναι παγκοσμίως η πιο ισχυρή και έχει αρκετά θετικά χαρακτηριστικά, έχει και ένα βασικό αρνητικό χαρακτηριστικό. Αυτό είναι ο αριθμός των ελληνόκτητων πλοίων που φέρουν την ελληνική σημαία επάνω τους. Συνολικά, από τα 5.514 πλοία τα 687 μόνο είναι νηολογημένα με ελληνική σημαία. Η χωρητικότητά των πλοίων σε dwt είναι ίση με 61.8 εκατομμύρια τόνους. Με βάση την ελληνική σημαία στα πλοία η Ελλάδα σαν ναυτιλιακή δύναμη, κατατάσσεται 8^{η} σε παγκόσμιο επίπεδο ενώ βρίσκεται στην 2^{η} θέση σε Ευρωπαϊκό επίπεδο. Όπως δείχνουν και τα στατιστικά στοιχεία που εξέδωσε το «Committee» στην ετήσια έκθεσή του, τα βασικά νηολόγια των ελληνόκτητων πλοίων είναι αυτά της Λιβερίας, των Νήσων Μάρσαλ και της Μάλτας, ενώ στην τέταρτη θέση στην κατάταξη βρίσκονται της Ελλάδας.

2.11 Τα μεγαλύτερα λιμάνια του κόσμου

Ενδιαφέρον έχει να παρουσιαστούν τα μεγαλύτερα λιμάνια του κόσμου, όπου πραγματοποιούνται φυσικά πολλά δρομολόγια καθημερινά και είναι σημεία αναφοράς για την παγκόσμια ναυτιλία. Θα παρουσιαστούν τα πέντε μεγαλύτερα λιμάνια στον κόσμο στην συγκεκριμένη υπό ενότητα. Σύμφωνα λοιπόν με την μονάδα μέτρησης μεγέθους λιμανιών που είναι η λεγόμενη TEU «Twenty-foot Equivalent unit» και με τα δεδομένα του 2020 από το επίσημο site του παγκόσμιου συμβούλιου ναυτιλίας (World Shipping Council) τα πέντε μεγαλύτερα και με τον περισσότερο όγκο φορτίων και τον μεγαλύτερο αριθμό πλοίων που διακινήθηκαν με την πάροδο του χρόνου είναι τα λιμάνια της Σανγκάης, της Σιγκαπούρης, το λιμάνι Ningbo-Zhoushan, το λιμάνι Shenzhen και αυτό του Guangzhou. Το πρώτο και μεγαλύτερο λιμάνι, αυτό της Σανγκάης διαθέτει 5 σταθμούς εμπορευματοκιβωτίων και έχει βαθμολογία 43,5 εκατομμύρια TEU. Το λιμάνι αυτό έχει τεράστιο ρόλο στην οικονομία της Κίνας καθώς είναι υπεύθυνο για το 1/4 της συνολικής εισαγόμενης και εξαγόμενης κίνησης φορτίου της Κίνας. Όσον αφορά το λιμάνι της Σιγκαπούρης, με βαθμολογία TEU ίση με 36,6 εκατομμύρια, διαδραματίζει σπουδαίο ρόλο διότι διαχειρίζεται τις μισές αποστολές αργού πετρελαίου στον κόσμο. Επιπλέον, θεωρείται στρατηγικό λιμάνι λόγω της γεωγραφικής του θέσης, αφού βρίσκεται στον βασικό θαλάσσιο δρόμο μεταξύ Ασίας και μέσης Ανατολής. Εξαιτίας αυτής της θέσης, το λιμάνι της Σιγκαπούρης είναι ένας παγκόσμιος σημαντικός σταθμός για τον ανεφοδιασμό πλοίων και συνεπώς για την διεθνή ναυτιλία. Το τρίτο λιμάνι στην κατάταξη με βαθμολογία 28,72 εκατομμύρια TEU, είναι εκείνο με το όνομα Ningbo-Zhoushan. Το λιμάνι του Ningbo-Zhoushan φιλοξενεί επίσης έναν τερματικό σταθμό αργού πετρελαίου χωρητικότητας 250.000 τόνων και μια θέση φόρτωσης μεταλλευμάτων 200.000 τόνων. Άλλες εγκαταστάσεις περιλαμβάνουν έναν ειδικά κατασκευασμένο τερματικό σταθμό για πλοία έκτης γενιάς και μια ειδική κουκέτα για υγρά χημικά προϊόντα. Στην συνέχεια της κατάταξης βρίσκεται το λιμάνι Shenzhen με 26,55 εκατομμύρια TEU. Με συνολικά 140 θέσεις ελλιμενισμού, το λιμάνι του Shenzhen διαθέτει εγκαταστάσεις που απλώνονται στον κόλπο Da Chan, Shekou, Chiwan, Mawan, Yantian, Dongjiaotou, Fuyong, Xiadong, Shayuchong και Neihe. Τέλος, το πέμπτο μεγαλύτερο λιμάνι παγκοσμίως, είναι αυτό του Guangzhou με 23,19 εκατομμύρια TEU. Το λιμάνι του Guangzhou είναι ένα ιδιαίτερα σημαντικό μέρος του διεθνούς εμπορίου της Κίνας, δεδομένου ότι το εμπόριο του λιμανιού φτάνει σε πάνω από 300 λιμάνια σε περισσότερες από 80 χώρες. Αυτό που προκαλεί εντύπωση είναι ότι το ένα από αυτά τα πέντε λιμάνια είναι στην Σιγκαπούρη ενώ τα υπόλοιπα τέσσερα λιμάνια βρίσκονται στην Κίνα. Είναι συνεπώς προφανές γιατί η Κίνα είναι μια τεράστια οικονομική δύναμη με σπουδαία ναυτιλία.

2.12 Το λιμάνι του Πειραιά

Το λιμάνι του Πειραιά δεν είναι τυχαίο ότι συμπεριλήφθηκε ξεχωριστά σε μια ενότητα. Πρόκειται για το σημαντικότερο και μεγαλύτερο λιμάνι της Ελλάδας και το μεγαλύτερο της μεσογείου αντίστοιχα. Η γεωγραφική του θέση το καθιστά κομβικό μέσο για την μεταφορά εμπορευμάτων, αλλά και για τον τουρισμό αφού δεν συνδέει μόνο τα ελληνικά νησιά με την ηπειρωτική χώρα αλλά και τα Βαλκάνια και τις χώρες της μεσογείου με αυτές της Μαύρης

Θάλασσας. Ειδικά για τον τουρισμό μιλώντας, το λιμάνι του Πειραιά είναι το μεγαλύτερο επιβατικό λιμάνι της Ευρώπης και ένα από τα μεγαλύτερα παγκοσμίως. Για το 2017 μάλιστα, τα στοιχεία (2017) έδειξαν ότι το λιμάνι του Πειραιά κατέλαβε την 24 θέση στην παγκόσμια κατάταξη με τους συνολικούς ταξιδιώτες κρουαζιέρας να είναι 1.055.559. Συγκριτικά στοιχεία για το 2018 και το 2019 έδειξαν ότι οι τουρίστες το 2019 ήταν 1.098.091 επιβάτες, έναντι 961.632 το 2018, αύξηση της τάξεως 14,2%. Για το 2019, σύμφωνα με την (Lloyd's List), σε κλίμακα TEU, το λιμάνι του Πειραιά βρισκόταν στην 26 θέση. Σύμφωνα με την επίσημη ιστοσελίδα του (ΟΛΠ), το 2014 το λιμάνι είχε έσοδα 21,72 εκατομμύρια ευρώ ενώ η ετήσια χωρητικότητα ήταν 73,1 εκατομμύρια τόνοι φορτίου.

ΚΕΦΑΛΑΙΟ 3ο

3. Στατιστική Μηχανική Μάθηση (Machine Learning)

3.1 Η έννοια της στατιστικής μηχανικής μάθησης

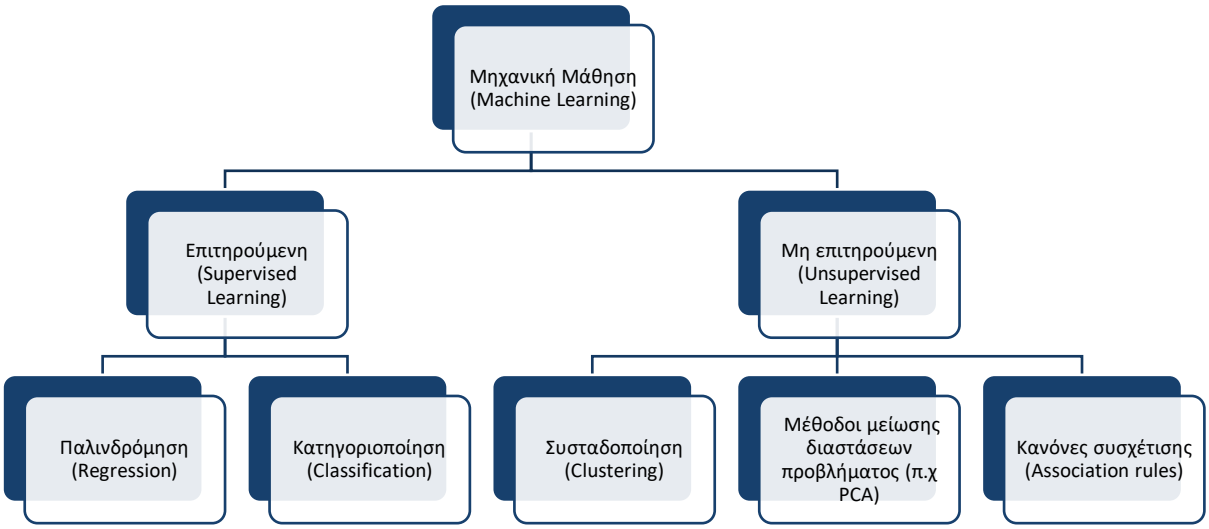
Τα δεδομένα που ρέουν στις επιχειρήσεις, στους δημόσιους και ιδιωτικούς οργανισμούς και κατακλύζουν πλέον τον κόσμο είναι πάρα πολλά εξαιτίας της ραγδαίας ανάπτυξης της επιστήμης των υπολογιστών και της τεχνολογίας. Ο όγκος των δεδομένων είναι τόσο τεράστιος που καθιστά πολύ δύσκολο να εντοπιστεί και να εξαχθεί χρήσιμη γνώση και πληροφορία μέσα από αυτά. Η ανίχνευση «κρυφής» πληροφορίας, η εύρεση σχέσεων, τάσεων και νέων προτύπων και γενικότερα η ανάλυση αυτού του τεράστιου πλήθους δεδομένων μόνο απλή δεν μπορεί να θεωρείται. Την λύση στο πρόβλημα αυτό, της επεξεργασίας, της ανάλυσης και ερμηνείας του τεράστιου πλήθους δεδομένων έρχεται να δώσει ο συνδυασμός δύο επιστημονικών πεδίων, αυτού της στατιστικής και εκείνου της πληροφορικής. Η στατιστική πρόκειται διαχρονικά για μία από τις πιο σημαντικές επιστήμες καθώς έχει εφαρμογή σχεδόν σε όλους τους υπόλοιπους επιστημονικούς και μη επιστημονικούς κλάδους όπως σημαντική την καθιστά και η καθημερινή της χρήση σε διάφορα προβλήματα των ανθρώπων. Η στατιστική ασχολείται με την συλλογή δεδομένων, την επεξεργασία τους, την ανάλυσή και την παρουσίασή τους και τέλος την ερμηνεία τους με σκοπό την εξαγωγή χρήσιμων συμπερασμάτων σχετικά με τωρινές ή μελλοντικές και αβέβαιες καταστάσεις. Δεν είναι άστοχο να θεωρηθεί ότι κατά μία έννοια, ο ανθρώπινος νους λειτουργεί με βάση τις αρχές της στατιστικής επιστήμης. Δηλαδή ο ανθρώπινος νους προσπαθεί μέσω των εμπειριών του, μέσω της παρατήρησης και της ανάλυσης αυτών που βιώνει να καταλήξει σε κάποια συμπεράσματα και να τα γενικεύσει δημιουργώντας με αυτόν τον τρόπο κάποια πρότυπα. Τις περισσότερες φορές όμως, για να αναλυθούν τα υπάρχοντα δεδομένα χρησιμοποιώντας αλγορίθμους και μοντέλα προβλέψεων έτσι ώστε να εκτιμηθούν και να προβλεφθούν μελλοντικές καταστάσεις, απαιτείται τεράστια υπολογιστική ισχύς. Αυτή η επιτακτική ανάγκη για την εξόρυξη γνώσης από ένα υπέρογκο πλήθος διαθέσιμων δεδομένων και παράλληλα η ανάγκη για μεγάλη υπολογιστική ισχύ, έχει φέρει στην επιφάνεια ένα νέο σχετικά πεδίο της επιστήμης των υπολογιστών και της πληροφορικής που ονομάζεται «Στατιστική Μηχανική Μάθηση». Η στατιστική μηχανική μάθηση χρησιμοποιεί συνδυαστικά στην ουσία τεχνικές προγραμματισμού και στατιστικής έτσι ώστε να αναπτυχθούν και να προσαρμοστούν μοντέλα τα οποία εκπαιδεύονται πάνω στα διαθέσιμα δεδομένα που έχουν οι ιδιωτικοί, δημόσιοι οργανισμοί και επιχειρήσεις. Τα δεδομένα αυτά στην συνέχεια, οι αλγόριθμοι μηχανικής μάθησης τα επεξεργάζονται, τα αναλύουν και βρίσκουν «κρυφή» πληροφορία και πρότυπα μέσα σε αυτά με τέτοιο ευφυή τρόπο έτσι ώστε, όταν εισέρχονται νέα δεδομένα στους διάφορους αυτούς οργανισμούς, να μπορούν τα μοντέλα αυτά, να κάνουν όσο το δυνατόν γίνεται πιο σωστές προβλέψεις χωρίς να είναι ρητά προγραμματισμένα. Αυτή η δυνατότητα των αλγορίθμων

στατιστικής μηχανικής μάθησης να «μαθαίνουν» από ιστορικά δεδομένα και να μπορούν μέσα από αυτά να προβλέπουν νέες τάσεις και καταστάσεις χωρίς να είναι προγραμματισμένοι απόλυτα για τον σκοπό αυτό, καθιστά την μηχανική μάθηση από πολλούς, υπό πεδίο της τεχνητής νοημοσύνης («Artificial Intelligence»). Συνεπώς οι αλγόριθμοι στατιστικής μηχανικής μάθησης συντελούν ένα σύνολο μεθόδων και τεχνικών που, όπως οι άνθρωποι μέσω της μαθησιακής τους ικανότητας και εμπειρίας, προσπαθούν να επιλύσουν προβλήματα που θεωρούνταν αδύνατο να επιλυθούν τα προηγούμενα χρόνια.

3.2 Κατηγορίες τεχνικών μηχανικής μάθησης

Οι τεχνικές μηχανικής μάθησης χωρίζονται κατά κύριο λόγο σε τρεις μεγάλες κατηγορίες που είναι η επιτηρούμενη ή αλλιώς επιβλεπόμενη μάθηση (Supervised Learning), η μη επιτηρούμενη ή αλλιώς μη επιβλεπόμενη μάθηση (Unsupervised Learning) και η ενισχυτική μάθηση (Reinforcement Learning) (The Elements of Statistical Learning, 2001) (Στατιστική μηχανική μάθηση, 2021). Επιπλέον, σύμφωνα και με την διεθνή βιβλιογραφία υπάρχει και μια τέταρτη κατηγορία μηχανικής μάθησης, η ημι-επιτηρούμενη μάθηση (Supervised and Unsupervised Learning for Data Science, 2019) (Semi - Supervised Learning, 2006). Ιδιαίτερη βάση δίνεται κυρίως στις πρώτες δύο κατηγορίες, δηλαδή την επιτηρούμενη και την μη επιτηρούμενη μάθηση. Στην επιτηρούμενη μάθηση τα διαθέσιμα δεδομένα έχουν «ετικέτες (labels)» σχετικά με το επιθυμητό τους αποτέλεσμα. Δηλαδή στην επιτηρούμενη μάθηση οι αλγόριθμοι δέχονται ως είσοδο τις τιμές ενός μέρους του συνόλου δεδομένων (πειραματικά δεδομένα ή δεδομένα εκπαίδευσης) για τα οποία δεδομένα εισόδου, γνωρίζουν που αντιστοιχίζονται, ποιο είναι με άλλα λόγια το αποτέλεσμά τους. Το αποτέλεσμα των δεδομένων εισόδου, δηλαδή η έξοδος με άλλα λόγια των δεδομένων που εισάγονται στον αλγόριθμο για την εκπαίδευσή του αναφέρονται και ως δεδομένα εξόδου. Στόχος της επιτηρούμενης μάθησης είναι να μάθει ο αλγόριθμος έναν κανόνα με βάση τον οποίο θα είναι σε θέση να αντιστοιχεί τις εισόδους με τα αποτελέσματά τους (τις εξόδους) έτσι ώστε όταν εισέρχονται νέα δεδομένα να μπορεί ο αλγόριθμος να προβλέπει όσο πιο σωστά και αξιόπιστα γίνεται το αποτέλεσμα των νέων δεδομένων εισόδου.

Αντιθέτως στην μη επιτηρούμενη μάθηση, ο αλγόριθμος χωρίς να γνωρίζει κάτι για τα αποτελέσματα των δεδομένων και τις σχέσεις μεταξύ των δεδομένων εισόδου και των αποτελεσμάτων, θα πρέπει μόνος, χωρίς δηλαδή την χρήση δεδομένων εισόδου και κάποιου επιθυμητού αποτελέσματος, να βρει την δομή των δεδομένων. Με την χρήση της μη επιτηρούμενης μάθησης μπορούν να βρεθούν κρυμμένα πρότυπα και μοτίβα μέσα στα δεδομένα. Η επιτηρούμενη μάθηση χωρίζεται σε προβλήματα παλινδρόμησης (regression) και προβλήματα κατηγοριοποίησης (classification) ενώ η μη επιτηρούμενη μηχανική μάθηση χωρίζεται σε προβλήματα συσταδοποίησης (clustering), μείωσης διαστάσεων προβλήματος (dimension reduction methods) καθώς και σε προβλήματα κανόνων συσχέτισης (association rules).



Σχήμα 2: Κατηγορίες μηχανικής μάθησης

Στην συνέχεια παρουσιάζονται με βάση την βιβλιογραφία τα διάφορα είδη των αλγορίθμων μηχανικής μάθησης που χρησιμοποιούνται αναλόγως του είδους μηχανικής μάθησης (επιτηρούμενη – μη επιτηρούμενη) και του είδους του προβλήματος που καλούμαστε να αντιμετωπίσουμε σε κάθε περίπτωση. Επιπλέον παρουσιάζονται και τα κυριότερα μέτρα αξιολόγησης αυτών των αλγορίθμων αναλόγως το πρόβλημα. Εδώ πρέπει να σημειωθεί ότι η επιλογή του κάθε αλγορίθμου που πρόκειται να χρησιμοποιηθεί για κάποιο πρόβλημα και πιο συγκεκριμένα η ικανότητα πρόβλεψής τους δηλαδή η απόδοσή του, εξαρτάται από αρκετούς παράγοντες όπως από την η δομή του συνόλου δεδομένων που έχουμε στην διάθεσή μας καθώς και από την φύση του προβλήματος που καλούμαστε να επιλύσουμε.

3.3 Η τεχνική της παλινδρόμησης – Μοντέλα παλινδρόμησης (Regression models)

Η παλινδρόμηση πρόκειται για μια στατιστική τεχνική που εξετάζει την σχέση μεταξύ δύο ή περισσότερων μεταβλητών με σκοπό την πρόβλεψη μιας μεταβλητής από αυτές με την βοήθεια των υπόλοιπων μεταβλητών. Για να μπορέσει να εφαρμοστεί ένα μοντέλο παλινδρόμησης θα πρέπει οι μεταβλητές να εμφανίζουν γραμμική συσχέτιση μεταξύ τους. Δηλαδή αύξηση της τιμής της μιας μεταβλητής να προκαλεί κάποια μείωση ή αύξηση στη τιμή της άλλης μεταβλητής. Η γραμμική συσχέτιση μεταξύ δύο μεταβλητών στην πιο απλή περίπτωση με δύο μόνο μεταβλητές, σημαίνει ότι οι τιμές της εξαρτημένης μεταβλητής (dependent variable or response variable) Y ή αλλιώς της μεταβλητής απόκρισης που θέλουμε να προβλέψουμε, μπορούν να προβλεφθούν εφόσον οι τιμές της ανεξάρτητης μεταβλητής (independent variable or

exploratory variable) X είναι γνωστές. Αντίστοιχα το ίδιο μοντέλο παλινδρόμησης για δύο μεταβλητές, μπορεί να γενικευτεί και για πάνω από δύο διαθέσιμες μεταβλητές, έστω n το πλήθος. Δηλαδή οι τιμές της εξαρτημένης μεταβλητής Υ μπορούν να προβλεφθούν όταν είναι γνωστές οι τιμές των ανεξάρτητων μεταβλητών $X_1, X_2, \dots, X_n$. Η εξαρτημένη μεταβλητή Υ πρόκειται για μια συνεχή ποσοτική μεταβλητή οπότε παίρνει μόνο συνεχείς τιμές. Σχεδόν πάντοτε είναι αδύνατον να προβλεφθεί η τιμής της εξαρτημένης μεταβλητής με απόλυτη ακρίβεια διότι υπάρχει συνήθως ένα «τυχαίο σφάλμα», δηλαδή κάποιοι παράγοντες τυχαίοι που δεν είναι δυνατόν να ελεγχθούν κατά την προσαρμογή του μοντέλου παλινδρόμησης, κάτι το οποίο δεν επιτρέπει την ακριβή πρόβλεψη της τιμής της εξαρτημένης μεταβλητής Υ για κάποια τιμή της εξαρτημένης τιμής Χ ή για κάποιες τιμές των υπόλοιπων μεταβλητών $X_1, X_2, \dots, X_n$ σε γενικότερο πλαίσιο. Συνεπώς η γενική μορφή ενός γραμμικού μοντέλου που μπορεί να περιγράψει την σχέση μεταξύ της μεταβλητής απόκρισης Υ και των ανεξάρτητων μεταβλητών $X_1, X_2, \dots, X_n$ είναι η (Ανάλυση Παλινδρόμησης, 2010):

$$\boldsymbol{Y = \alpha + \beta X + \varepsilon},$$

όπου $\mathbf{X}$ είναι το διάνυσμα των ανεξάρτητων μεταβλητών $\mathbf{X} = (X_1, X_2, \dots, X_n)$, $\boldsymbol{\beta}$ είναι το διάνυσμα των συντελεστών των ανεξάρτητων μεταβλητών δηλαδή $\boldsymbol{\beta} = (\beta_1, \beta_2, \dots, \beta_n)$ και $\boldsymbol{\varepsilon}$ δηλώνει το τυχαίο σφάλμα κατά την προσαρμογή του μοντέλου παλινδρόμησης.

3.3.1 Απλή & Πολλαπλή γραμμική παλινδρόμηση (Simple Linear & Multiple Linear Regression model)

Η απλούστερη περίπτωση ενός γραμμικού μοντέλου στην ανάλυση παλινδρόμησης είναι η απλή γραμμική παλινδρόμηση, δηλαδή όταν θέλουμε να προβλέψουμε τις τιμές της εξαρτημένης Υ μέσω των τιμών μιας ανεξάρτητης μεταβλητής Χ. Σε αυτήν την περίπτωση η σχέση που συνδέει τις δύο μεταβλητές είναι η (Εφαρμοσμένη στατιστική με έμφαση στις επιστήμες υγείας, 2016):

$$Y = \beta_0 + \beta_1 X + \varepsilon$$

Αν για την πρόβλεψη των τιμών της εξαρτημένης μεταβλητής Υ υπάρχουν διαθέσιμες k το πλήθος μεταβλητές, τότε το μοντέλο πολλαπλής γραμμικής παλινδρόμησης γράφεται ως:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta X_k + \varepsilon$$

Σκοπός είναι να βρεθεί κάποια ευθεία που να προσαρμόζεται όσο το δυνατόν καλύτερα τα δεδομένα. Δηλαδή, εστιάζοντας στην περίπτωση της απλής γραμμικής παλινδρόμησης να περιγράφει ικανοποιητικά σε μεγάλο βαθμό την σχέση μεταξύ της εξαρτημένης μεταβλητής Υ και της ανεξάρτητης μεταβλητής Χ. Για να επιτευχθεί αυτό θα πρέπει να βρεθούν οι τιμές των συντελεστών β_0 και β_1. Αυτή η διαδικασία εύρεσης των τιμών των συντελεστών β_0 και β_1 για την εύρεση της «βέλτιστης» ευθείας που περιγράφει τα δεδομένα λέγεται εκτίμηση ενώ οι τιμές που θα παίρνουν αυτοί οι συντελεστές μόλις ολοκληρωθεί η διαδικασία αυτή λέγονται εκτιμήτριες. Από όλες τις μεθόδους που έχουν δοκιμαστεί, εκείνη η ευθεία η οποία φαίνεται να προσαρμόζεται καλύτερα στα δεδομένα είναι αυτή που ελαχιστοποιεί το άθροισμα τετραγώνων

των σφαλμάτων $\boldsymbol{\varepsilon_i}$ και είναι γνωστή ευρέως με το όνομα «μέθοδος ελαχίστων τετραγώνων». Αυτό που πρέπει να ελαχιστοποιηθεί είναι το άθροισμα τετραγώνων των σφαλμάτων:

$$\sum_{i=1}^{n} \varepsilon_i^2 = \sum_{i=1}^{n} (y_i - (\beta_0 + \beta_1 x_i))^2$$

Οι εκτιμήτριες ελαχίστων τετραγώνων (συμβολίζονται με $\boldsymbol{b_0}$ και $\boldsymbol{b_1}$) δηλαδή οι τιμές των συντελεστών b_0 και b_1 που ελαχιστοποιούν την παραπάνω σχέση βρίσκονται υπολογίζοντας τις παρακάτω σχέσεις:

$$\boldsymbol{b_1} = \frac{n\sum_{i=1}^{n} x_i y_i - (\sum_{i=1}^{n} x_i) - (\sum_{i=1}^{n} y_i)}{n\sum_{i=1}^{n} x_i^2 - (\sum_{i=1}^{n} x_i)^2}, \qquad \boldsymbol{b_0} = \frac{1}{n}\sum_{i=1}^{n} y_i - \widehat{b_1}\,\frac{1}{n}\sum_{i=1}^{n} x_i$$

Η ευθεία $\hat{y} = b_1 + b_0 x$ καλείται ευθεία ελαχίστων τετραγώνων.

<u>Ερμηνεία εκτιμητριών ελαχίστων τετραγώνων</u>

Ο συντελεστής b_1 της ευθείας $\hat{y} = b_1 + b_0 x$ εκφράζει την μεταβολή που θα επέλθει στην εξαρτημένη μεταβλητή Y όταν η ανεξάρτητη μεταβλητή X μεταβληθεί κατά μία μονάδα. Ο συντελεστής b_0 της ευθείας $\hat{y} = b_1 + b_0 x$ παριστάνει την τιμή που θα πάρει η εξαρτημένη μεταβλητή Y όταν η ανεξάρτητη μεταβλητή X είναι ίση με το 0.

Αν έχουμε κ το πλήθος ανεξάρτητες μεταβλητές τότε ο συντελεστής b_1 της ευθείας $\hat{y} = b_0 + b_1 x_1 + b_2 x_2 + \cdots + b_k x_k$ δείχνει την μεταβολή που θα επέλθει στην εξαρτημένη μεταβλητή Y αν η ανεξάρτητη μεταβλητή X_1 αυξηθεί κατά μία μονάδα και οι υπόλοιπες ανεξάρτητες μεταβλητές παραμείνουν σταθερές.

<u>Εκτιμημένα Σφάλματα (Περίπτωση απλής γραμμικής παλινδρόμησης)</u>

- Η ποσότητα SSE (ή RSS) = $\sum_{i=1}^{n} \hat{\varepsilon}_i^2 = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$ ονομάζεται άθροισμα τετραγώνων των εκτιμημένων σφαλμάτων (Sum of Squares of Errors) και προκύπτει από την γνωστή σχέση $\boldsymbol{\hat{\varepsilon}_i = y_i - \hat{y}_i = y_i - (b_0 + b_1 x)}$, όπου $\hat{\varepsilon}_i$ εκφράζει την διαφορά του πραγματικού y_i από το εκτιμώμενο $\hat{y}_i$ και ονομάζεται κατάλοιπο ή αλλιώς υπόλοιπο (residual). Η ποσότητα SSE μετρά την μεταβλητότητα που δεν γίνεται να ερμηνευθεί από το μοντέλο παλινδρόμησης που έχει προσαρμοστεί.
- Η ποσότητα SSR = $\sum_{i=1}^{n} (\hat{y}_i - \bar{y})^2$ ονομάζεται άθροισμα τετραγώνων λόγω της παλινδρόμησης (Regression Sum of Squares) και μετρά την μεταβλητότητα που ερμηνεύεται από το μοντέλο παλινδρόμησης που έχει προσαρμοστεί στα δεδομένα.
- Η ποσότητα SST = $\sum_{i=1}^{n} (y_i - \bar{y})^2$ ονομάζεται συνολικό άθροισμα τετραγώνων (Total Sum of Squares) και παριστάνει την ολική μεταβλητότητα του μοντέλου παλινδρόμησης που έχει προσαρμοστεί. Ισχύει ότι SST = SSR + SSE. Η τελευταία σχέση για την ολική μεταβλητότητα του μοντέλου παλινδρόμησης ισχύει και στην περίπτωση της πολλαπλής παλινδρόμησης.

<u>Υποθέσεις προσαρμογής μοντέλου γραμμικής παλινδρόμησης</u>

Για να δώσει ένα μοντέλο γραμμικής παλινδρόμησης που προσαρμόζεται σε κάποια δεδομένα αξιόπιστα αποτελέσματα θα πρέπει να τηρούνται κάποιες βασικές υποθέσεις.

1. Η εξαρτημένη μεταβλητή Y συνδέεται με την ανεξάρτητη μεταβλητή X σύμφωνα με την γραμμική σχέση:

$$Y_i = \beta_0 + \beta_1 X_i + \varepsilon_i ,$$

όπου β_0 και β_1 είναι άγνωστες παράμετροι και η τιμή της μεταβλητής απόκρισης X_i πρόκειται για γνωστό αριθμό. Τα σφάλματα ε_i και η τιμή της εξαρτημένης μεταβλητής Y_i είναι τυχαίες μεταβλητές

2. Ομασκεδαστικότητα σφαλμάτων (έχουν σταθερή διακύμανση τα σφάλματα)

3. Κανονικότητα σφαλμάτων

4. Τα σφάλματα πρέπει να είναι ασυσχέτιστα μεταξύ τους

5. Στην περίπτωση της πολλαπλής παλινδρόμησης, να μην υπάρχει το φαινόμενο της πολυσυγγραμικότητας (multicollinearity) δηλαδή να είναι δύο ή περισσότερες ανεξάρτητες μεταβλητές ισχυρά συσχετισμένες μεταξύ τους

3.3.2 Παλινδρόμηση Ridge (Ridge Regression)

Σε μοντέλα πολλαπλής παλινδρόμησης που προσαρμόζονται, εμφανίζονται πολύ συχνά υψηλές συσχετίσεις μεταξύ δύο ή περισσότερων ανεξάρτητων μεταβλητών. Αυτό είναι γνωστό σαν πρόβλημα πολυσυγγραμικότητας που υπάρχει στα δεδομένα. Όταν υπάρχει πολυσυγγραμικότητα και επομένως ορισμένες μεταβλητές εμφανίζουν πολύ μεγάλη συσχέτιση μεταξύ τους, τότε οι εκτιμήσεις των παραμέτρων δεν είναι ακριβείς και εμφανίζουν μεγάλες αποκλίσεις από τις πραγματικές τους τιμές πράγμα που τείνει να δυσκολέψει την ερμηνεία των παραμέτρων του μοντέλου και να αυξήσει την σημαντικότητα κάποιων μεταβλητών ενώ στην πραγματικότητα δεν είναι σημαντικές. Επειδή για να αποφευχθεί η πολυσυγγραμικότητα των δεδομένων θα πρέπει να αφαιρεθούν κάποιες ανεξάρτητες μεταβλητές από το μοντέλο παλινδρόμησης, κάτι το οποίο δεν είναι πάντα εφικτό στην πράξη, εφαρμόζεται για αυτόν τον λόγο σε τέτοιες περιπτώσεις η παλινδρόμηση Ridge (Εισαγωγή στην Ridge Regression, 2021) (StatLect, 2021). Στην παλινδρόμηση Ridge αντί να χρησιμοποιούνται αμερόληπτοι εκτιμητές για τις εκτιμήσεις των συντελεστών του μοντέλου παλινδρόμησης, χρησιμοποιούνται μεροληπτικοί εκτιμητές επειδή η μεροληψία τους είναι πολύ μικρή και σε περίπτωση πολυσυγγραμικότητας δίνουν πιο ακριβείς εκτιμήσεις για τις παραμέτρους του μοντέλου. Η λογική της μεθόδου Ridge Regression είναι ίδια με αυτής της πολλαπλής παλινδρόμησης μόνο που διαφοροποιεί κάπως την μέθοδο ελαχίστων τετραγώνων για την ελαχιστοποίηση του αθροίσματος τετραγώνων των σφαλμάτων. Ουσιαστικά χρησιμοποιεί μια ποινή με την χρήση μιας επιπλέον παραμέτρου ρύθμισης που συμβολίζεται συνήθως με λ και η οποία πρόκειται για μια μη αρνητική τιμή, μικρότερη συνήθως από 0,3 η οποία προστίθεται στα διαγώνια στοιχεία

πίνακα συσχετίσεων και που επιτρέπει μεροληπτικούς εκτιμητές για τους συντελεστές του μοντέλου παλινδρόμησης. Η τεχνική παλινδρόμησης Ridge δεν μπορεί να κάνει επιλογή των πιο σημαντικών μεταβλητών για το τελικό μοντέλο παλινδρόμησης που πρόκειται να προσαρμοστεί καθώς οι τιμές των συντελεστών μπορεί μέσω αυτής της μεθόδου να συρρικνώνονται αρκετά αλλά δεν γίνονται ίσοι με το μηδέν. Παρόλα αυτά, οι πιο «ασήμαντες» μεταβλητές του μοντέλου θα έχουν συντελεστές των οποίων η τιμή θα είναι πολύ κοντά στο μηδέν. Μπορεί να μην υπάρχουν σε αυτήν την περίπτωση αμερόληπτοι εκτιμητές αλλά με αυτήν την διαδικασία μειώνεται σημαντικά η διασπορά των εκτιμήσεων. Για την εύρεση αμερόληπτων εκτιμητριών ελαχίστων τετραγώνων $\widehat{\boldsymbol{b}}$ στην περίπτωση της πολλαπλής γραμμικής παλινδρόμησης χρησιμοποιείται η σχέση:

$$\widehat{\boldsymbol{b}} = (\boldsymbol{X'X})^{-1}\,\boldsymbol{X'Y}$$

Στην περίπτωση της Ridge παλινδρόμησης όπως αναφέρθηκε προστίθεται μια μη αρνητική σταθερά λ στα διαγώνια στοιχεία του πίνακα συσχετίσεων για την εύρεση μεροληπτικών εκτιμητών οι οποίοι υπολογίζονται σύμφωνα με την σχέση:

$$\widehat{\boldsymbol{b}} = (\boldsymbol{X'X} + \boldsymbol{\lambda I})^{-1}\,\boldsymbol{X'Y}\,,$$

όπου I στην παραπάνω σχέση είναι ο ταυτοτικός (ή μοναδιαίος πίνακας) του οποίου τα στοιχεία της κύριας διαγωνίου είναι ίσα με 1 και όλα τα υπόλοιπα στοιχεία είναι ίσα με το 0. Επίσης, στην παραπάνω σχέση με $\mathbf{X}'$ συμβολίζεται ο ανάστροφος του πίνακα $\mathbf{X}$ (συμβολίζεται και με $\mathbf{X^T}$). Ο πίνακας $\mathbf{X'X}$ είναι ο «πίνακας πληροφορίας» του πολλαπλού γραμμικού μοντέλου. Η αντίστοιχη εξίσωση που πρέπει να ελαχιστοποιηθεί στην Ridge παλινδρόμηση είναι η κάτωθι:

$$\sum_{i=1}^{n} \boldsymbol{\varepsilon_i^2} = \sum_{i=1}^{n}(y_i - \sum_{j=0}^{p} x_{ij}\,\beta_j)^2 + \lambda \sum_{j=0}^{p} \beta_j^2\,,$$

και θα πρέπει να ισχύει ότι $\sum_{j=0}^{p} \beta_j^2 <$ c, όπου c > 0.

Η ποσότητα $\lambda \sum_{j=0}^{p} \beta_j^2$ είναι η ποινή που χρησιμοποιείται για την συρρίκνωση των συντελεστών των παραμέτρων του μοντέλου. Με την εφαρμογή της μεθόδου Ridge λοιπόν μειώνεται η διακύμανση των εκτιμήσεων των συντελεστών, εξαλείφεται η πολυσσυγγραμικότητα και συνεπώς αποφεύγεται η υπερπροσαρμογή (overfitting) των δεδομένων. Στην περίπτωση που η σταθερά λ τείνει στο 0 δηλαδή $\lambda \to 0$ τότε εμφανίζεται η εξίσωση της μεθόδου ελαχίστων τετραγώνων της γραμμικής παλινδρόμησης. Συνεπώς η παλινδρόμηση Ridge πρόκειται για μια τεχνική παλινδρόμησης που προσπαθεί να εξισορροπήσει τον συμβιβασμό «απώλειας αμερόληπτων εκτιμήσεων – μείωση διακύμανσης εκτιμήσεων» με σκοπό να αποφευχθεί η υπερπροσαρμογή των δεδομένων συρρικνώνοντας τους υψηλούς συντελεστές των «μη σημαντικών» παραμέτρων του μοντέλου.

3.3.3 Παλινδρόμηση Lasso (Lasso Regression)

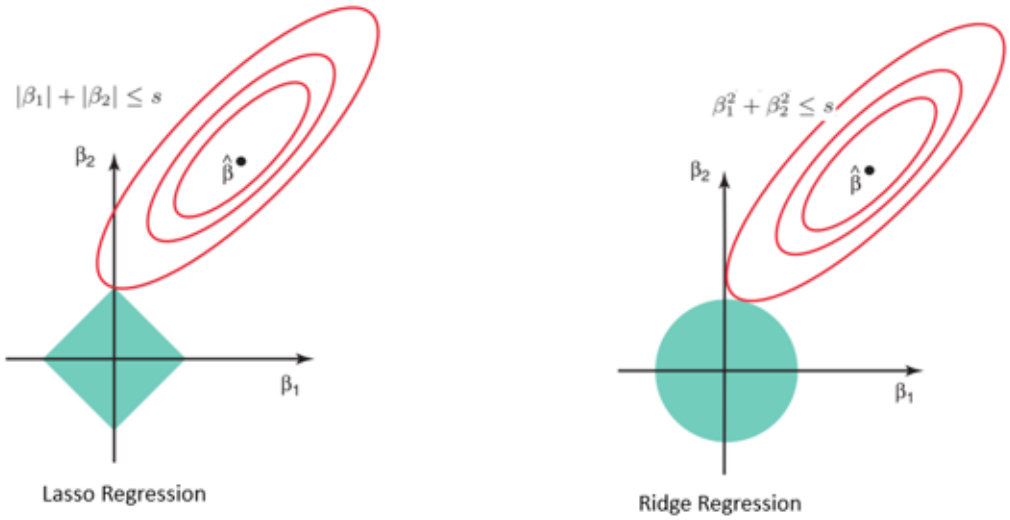
Σε κάποια σύνολα δεδομένων ο αριθμός των ανεξάρτητων μεταβλητών (έστω κ το πλήθος) είναι μεγαλύτερος από τον αριθμό των παρατηρήσεων ($\kappa > n$). Η μέθοδος παλινδρόμησης Lasso (Least Absolute Shrinkage and Selection Operator) λειτουργεί σχεδόν με την ίδια λογική της μεθόδου Ridge, μόνο που η συγκεκριμένη μέθοδος έχει ένα σημαντικό

πλεονέκτημα έναντι της μεθόδου Ridge. Αυτό είναι το ότι μπορεί να επιλέξει ένα υποσύνολο μεταβλητών από το αρχικά διαθέσιμο σύνολο δεδομένων μειώνοντας με αυτόν τον τρόπο την διάσταση του χώρου και έτσι μπορεί να δημιουργήσει πιο ερμηνεύσιμα και απλούστερα μοντέλα παλινδρόμησης. Χρησιμοποιείται και εδώ μια ρυθμιστική παράμετρος λ η οποία δημιουργεί μεροληπτικούς εκτιμητές για τους συντελεστές των παραμέτρων του μοντέλου μόνο που σε αυτή την περίπτωση όσο αυξάνεται η τιμή της παραμέτρου λ τόσο οι τιμές κάποιων συντελεστών ή μπορούν να μειωθούν τόσο ώστε να είναι ίσοι με το μηδέν. Και σε αυτήν την τεχνική παλινδρόμησης χρησιμοποιείται μια ποινή για τους υψηλούς συντελεστές των ασήμαντων ανεξάρτητων μεταβλητών. Οι συντελεστές που έχουν τιμή ίση με το 0 απαλείφονται από το μοντέλο παλινδρόμησης που πρόκειται να προσαρμοστεί. Αυτό έχει σαν αποτέλεσμα να μειώνεται αρκετά η διακύμανση των εκτιμήσεων και να εξάγονται πιο αξιόπιστες εκτιμήσεις για τους συντελεστές του μοντέλου. Στην περίπτωση της Lasso παλινδρόμησης η ποσότητα που πρέπει να ελαχιστοποιηθεί για την εύρεση των καταλληλότερων εκτιμητών φαίνεται παρακάτω (Discussion: Subset Selection, Ridge Regression and the Lasso, 2001):

$$\sum_{i=1}^{n} \varepsilon_i^2 = \sum_{i=1}^{n} (y_i - \sum_{j=0}^{p} x_{ij} \beta_j)^2 + \lambda \sum_{j=0}^{p} |\beta_j|$$

με τον περιορισμό $\sum_{j=0}^{p} |\beta_j| < g$, όπου $g > 0$

Η ποσότητα $\lambda \sum_{j=0}^{p} |\beta_j|$ είναι η ποινή που χρησιμοποιεί η μέθοδος Lasso για την συρρίκνωση των τιμών των συντελεστών του μοντέλου παλινδρόμησης. Όπως στην περίπτωση της Ridge παλινδρόμησης έτσι και εδώ αν η σταθερά λ τείνει στο 0 δηλαδή $\lambda \to 0$ τότε εμφανίζεται η εξίσωση της μεθόδου ελαχίστων τετραγώνων της γραμμικής παλινδρόμησης. Συνεπώς η μέθοδος Lasso, σε αντίθεση με την μέθοδο Ridge παρότι προσπαθεί να εξισορροπήσει τον συμβιβασμό «απώλειας αμερόληπτων εκτιμήσεων – μείωση διακύμανσης εκτιμήσεων», έχει σκοπό να γίνει επιλογή των πιο σημαντικών μεταβλητών από το αρχικό σύνολο δεδομένων για την δημιουργία και προσαρμογή ενός πιο απλού και ερμηνεύσιμου προβλεπτικού μοντέλου συρρικνώνοντας με την βοήθεια μιας παραμέτρου ποινής τις τιμές κάποιων συντελεστών σε τέτοιο βαθμό ώστε να είναι ίσες με μηδέν. Αξίζει να σημειωθεί ότι τόσο η μέθοδος Ridge όσο και η μέθοδος Lasso επιλύουν το πρόβλημα της υπερπροσαρμογής των δεδομένων (overfitting).
Το σχήμα 3 που φαίνεται παρακάτω, περιγράφει γεωμετρικά την λειτουργία των μεθόδων παλινδρόμησης Lasso και Ridge για δύο μόνο παραμέτρους. Οι ελλείψεις, δηλαδή οι κόκκινοι κύκλοι είναι στην ουσία οι συναρτήσεις κόστους για την κάθε μέθοδο. Οι περιοχές με πράσινο ανοικτό χρώμα που έχουν σχήμα ρόμβου και κύκλου πρόκειται για τους περιορισμούς που θέτουν οι δύο μέθοδοι με βάση τις παραμέτρους ποινής τους. Οι συντελεστές προσδιορίζονται βρίσκοντας το πρώτο σημείο που όπου τα ελλειπτικά περιγράμματα αγγίζουν την περιοχή των περιορισμών. Η διαφορά των δύο μεθόδων (DataCamp, 2022) εξαρτάται από το σχήμα των περιορισμών που έχουν. Η περιοχή περιορισμού της μεθόδου Lasso έχει σχήμα ρόμβου οπότε κάθε φορά που οι ελλειπτικές περιοχές τέμνονται με τις γωνίες του ρόμβου ένας τουλάχιστον από τους συντελεστές γίνεται ίσος με το μηδέν. Αντιθέτως, επειδή το σχήμα περιορισμού της μεθόδου Ridge είναι κύκλος οι τιμές των συντελεστών ναι μεν συρρικνώνονται και βρίσκονται κοντά στο μηδέν, από την άλλη δε, δεν γίνονται ποτέ ίσοι με το μηδέν.



Σχήμα 3: Γραφική απεικόνιση λειτουργίας Lasso και Ridge παλινδρόμησης
Πηγή σχημάτων: (https://www.datacamp.com/tutorial/tutorial-lasso-ridge-regression)

3.3.4 Παλινδρόμηση ελαστικού δικτύου (Elastic net regression)

Η μέθοδος παλινδρόμησης Elastic net συνδυάζει ταυτόχρονα τις μεθόδους Ridge και Lasso και συγκεκριμένα τις ποινές που χρησιμοποιούν αυτές οι δύο μέθοδοι για τους εκτιμητές των συντελεστών του μοντέλου παλινδρόμησης προκειμένου να ξεπεραστούν δύο περιορισμοί που έχει η μέθοδος Lasso (The Elements of Statistical Learning, 2001). Στην μέθοδο παλινδρόμησης Lasso αν υπάρχουν υψηλά συσχετισμένες μεταβλητές τότε η παράμετρος ποινής που χρησιμοποιείται, επιλέγει για την προσαρμογή του μοντέλου μόνο μια από αυτές τις μεταβλητές, και τις τιμές των συντελεστών των μεταβλητών που απομακρύνει από το μοντέλο τις συρρικνώνει και τις κάνει ίσες με το μηδέν. Αυτό έχει σαν αποτέλεσμα σε σύνολα δεδομένων όπου υπάρχουν αρκετές μεταβλητές με υψηλή συσχέτιση να απομακρύνεται σημαντικός αριθμός από αυτές και έτσι να χάνεται πληροφορία και συνεπώς να εξάγονται μεροληπτικές εκτιμήτριες. Ακόμα ένα μειονέκτημα είναι όταν υπάρχει τεράστια διαφορά μεταξύ πλήθους μεταβλητών και πλήθους δεδομένων και πιο συγκεκριμένα όταν το πλήθος μεταβλητών είναι πολύ μεγαλύτερο από το συνολικό δείγμα (έστω n το σύνολο του δείγματος). Σε μια τέτοια περίπτωση ο αριθμός των συντελεστών του μοντέλου παλινδρόμησης μπορεί να έχει μέχρι n το πλήθος μη μηδενικούς συντελεστές, με άλλα λόγια μέχρι n ανεξάρτητες μεταβλητές για το μοντέλο πρόβλεψης. Για παράδειγμα αν υπήρχαν 1000 παράμετροι σε ένα σύνολο δεδομένων μεγέθους ίσου με 20 τότε ως ανεξάρτητες μεταβλητές θα μπορούσαν να χρησιμοποιηθούν από τις διαθέσιμες 1000 παραμέτρους, το πολύ οι 20. Αυτό φυσικά θα οδηγήσει σε αναξιόπιστα αποτελέσματα. Για να ξεπεραστούν αυτά τα δύο προβλήματα λοιπόν οι Zou και Hastie (2005) πρότειναν αυτήν την νέα μέθοδο παλινδρόμησης Elastic net και την δική της σταθμισμένη ποινή για τους συντελεστές των παραμέτρων του μοντέλου παλινδρόμησης. Αυτή η ποινή όπως προαναφέρθηκε είναι μια μίξη

των ποινών που χρησιμοποιούν οι μέθοδοι Ridge και Lasso και καταφέρνει να συρρικνώσει τις τιμές των συντελεστών των «ασήμαντων» μεταβλητών και ταυτόχρονα να επιλέξει τις πιο σημαντικές μεταβλητές. Ακόμα και αν υπάρχουν υψηλά συσχετισμένες μεταβλητές αν θεωρηθούν σημαντικές τότε θα ληφθούν κανονικά υπόψιν όλες τους και όχι μία εξ 'αυτών. Επίσης δεν τίθεται πλέον ο περιορισμός του μέγιστου αριθμού μεταβλητών που μπορούν να χρησιμοποιηθούν στο μοντέλο παλινδρόμησης ανάλογα με τον συνολικό αριθμό δείγματος. Η συνάρτηση κόστους της παλινδρόμησης Elastic net εισάγει μια νέα παράμετρο που συμβολίζεται με «α» και παίρνει τιμές μεταξύ 0 και 1. Αν πάρει την τιμή 0 τότε εφαρμόζεται η παλινδρόμηση Ridge ενώ αν η παράμετρος α πάρει την τιμή 1 τότε εφαρμόζεται στην ουσία η παλινδρόμηση Lasso. Η συνάρτηση κόστους που πρέπει να ελαχιστοποιηθεί φαίνεται παρακάτω στην εξίσωση:

$$\sum_{i=1}^{n} \varepsilon_i^2 = \sum_{i=1}^{n}(y_i - \beta_0 - \sum_{j=1}^{p} x_{\iota j}\,\beta_j)^2 + \lambda \sum_{j=1}^{p} \frac{(1-a)}{2}\beta_j^2 + a|\beta_j|$$

3.3.5 Παλινδρόμηση διανυσμάτων υποστήριξης (Support Vector Regression)

Η μέθοδος παλινδρόμησης Support Vector στηρίζεται στην ουσία και έχει την ίδια λογική με την τεχνική κατηγοριοποίησης «Support Vector Machines» (The Elements of Statistical Learning, 2001). Σκοπός της μεθόδου Support Vector παλινδρόμησης είναι να βρεθεί μια «βέλτιστη» ευθεία η οποία θα περιέχει όσο το δυνατόν περισσότερα δεδομένα για την πρόβλεψη των διακριτών τιμών της εξαρτημένης μεταβλητής Y του μοντέλου. Αυτή η «βέλτιστη» ευθεία είναι γνωστή ως υπερεπίπεδο (hyperplane) (Support Vector Machines, 2014). Μέσα σε έναν χώρο με n διαστάσεις θα πρέπει ο αλγόριθμος αυτός να βρει το βέλτιστο υπερεπίπεδο που εφαρμόζει καλύτερα στα δεδομένα συμπεριλαμβάνοντας φυσικά όσο πιο πολλά δεδομένα γίνεται. Σημαντικό χαρακτηριστικό της παλινδρόμησης των διανυσμάτων υποστήριξης είναι οι γραμμές ορίων (boundary lines). Αυτές οι γραμμές τοποθετούνται σε απόσταση **ε** (epsilon) από το υπερεπίπεδο και χρησιμοποιούνται για να δημιουργηθεί ένα περιθώριο (margin) μεταξύ των σημείων δεδομένων. Αυτή η απόσταση **ε** αναφέρεται συχνά και ως «μέγιστο περιθώριο σφάλματος». Τα σημεία δεδομένων που βρίσκονται εκτός της απόστασης **ε**, δέχονται μια ποινή για το σφάλμα τους και συνήθως συμβολίζονται με ξ και υποδηλώνουν την απόκλιση από το περιθώριο. Σημαντική παράμετρος των διανυσμάτων υποστήριξης πέρα από το υπερεπίπεδο και τις γραμμές ορίων είναι και ο πυρήνας ο οποίος δεν είναι τίποτε άλλο από μαθηματικές συναρτήσεις που στόχο έχουν να βρουν το καλύτερο υπερεπίπεδο σε χώρους υψηλών διαστάσεων όπου τα σημεία δεδομένων δεν είναι γραμμικά διαχωρίσιμα. Οι πιο συνήθεις χρησιμοποιούμενοι πυρήνες είναι οι:

1. Linear
2. Non – Linear
3. Radial Basis function
4. Sigmoid
5. Polynomial

Σε αντίθεση με άλλα μοντέλα παλινδρόμησης, ο συγκεκριμένος αλγόριθμος προβλέπει διακριτές τιμές της μεταβλητής απόκρισης Y. Δηλαδή έχει επιτυχία και ουσία να εφαρμοστεί όταν η μεταβλητή απόκρισης Y λαμβάνει διακριτές τιμές. Επιπλέον, δεν ελαχιστοποιεί το

άθροισμα τετραγώνων των εκτιμημένων σφαλμάτων αλλά προσπαθεί να βρει την καλύτερη ευθεία εντός του χώρου μιας τιμής που ορίζεται σαν κατώφλι (threshold) για τον σκοπό αυτό. Η τιμή του κατωφλίου αυτού είναι στην πραγματικότητα η απόσταση μεταξύ του υπερεπιπέδου και μιας γραμμής ορίου. Τα πιο σημαντικά πλεονεκτήματα του αλγορίθμου Support Vector για την προσαρμογή ενός μοντέλου παλινδρόμησης είναι ότι είναι ανθεκτικός σε ακραίες τιμές (outliers), ερμηνεύεται εύκολα το μοντέλο που έχει προσαρμοστεί και έχει υψηλή συνήθως προβλεπτική ικανότητα για πολύπλοκα προβλήματα. Ένα μειονέκτημά του είναι ότι δεν έχει καλή προβλεπτική ικανότητα αν τα δεδομένα έχουν «θόρυβο». Η εξίσωση του υπερεπιπέδου μπορεί να γραφτεί στην παρακάτω μορφή:

$$\boldsymbol{Y = wx + b,}$$

όπου το **w** είναι ένα διάνυσμα βαρών και το **b** ονομάζεται μεροληψία.
Οι αντίστοιχες εξισώσεις των γραμμών ορίων μπορούν να γραφτούν ως εξής:

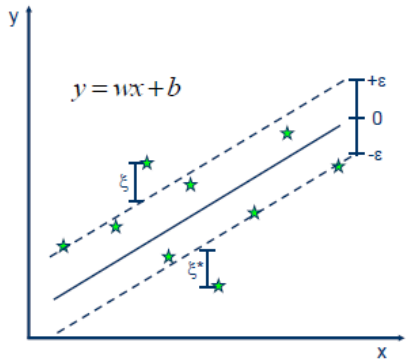
$$\boldsymbol{wx + b = \varepsilon} \text{ και } \boldsymbol{wx + b = - \varepsilon}$$

Η ποσότητα που πρέπει να ελαχιστοποιηθεί για την εύρεση του βέλτιστου υπερεπιπέδου δίνεται παρακάτω:

$$\boldsymbol{MIN}\ \frac{1}{2}\left|\left|\boldsymbol{w}\right|\right|^2 + \boldsymbol{C}\sum_{i=1}^{n}|\xi_i|$$

Η **C** πρόκειται για μια παράμετρο η οποία μετρά την ανοχή που υπάρχει για τα σημεία δεδομένων εκτός της απόστασης **ε.**

FIGURE 1



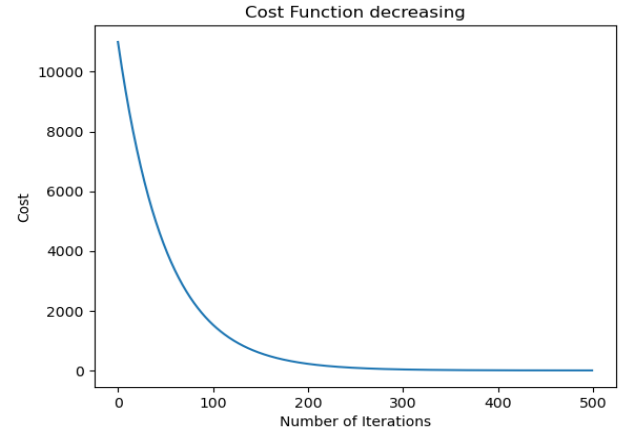
Σχήμα 4: Γραφική απεικόνιση της λειτουργίας της μεθόδου Support Vector Regression
Πηγή σχήματος: (https://www.saedsayad.com/support_vector_machine_reg.htm)

Η μεσαία ευθεία γραμμή είναι το υπερεπίπεδο ενώ οι δύο παράλληλες διακεκομμένες γραμμές είναι οι γραμμές ορίων. Τα σημεία που είναι συμβολισμένα με αστεράκια και τα οποία βρίσκονται επάνω στις γραμμές ορίων είναι τα διανύσματα υποστήριξης. Με ξ είναι συμβολισμένη η απόσταση των σημείων δεδομένων εκτός των γραμμών ορίων από αυτές.

3.3.6 Παλινδρόμηση Gradient Descent (Gradient Descent Regression)

Η τεχνική παλινδρόμησης Gradient Descentπρόκειται για μια επαναληπτική μέθοδο η οποία μέσα από μια επαναληπτική διαδικασία βελτιστοποίησης προσπαθεί να εκτιμήσει τις κατάλληλες τιμές των συντελεστών των παραμέτρων του μοντέλου παλινδρόμησης έτσι ώστε να

βελτιστοποιηθεί η τιμή της συνάρτησης κόστους που σημαίνει να ελαττωθεί όσο περισσότερο γίνεται η τιμή της (The Elements of Statistical Learning, 2001) (TowardsDatascience, 2018). Στην περίπτωση που υπάρχει μόνο μία ανεξάρτητη μεταβλητή στο μοντέλο, ο αλγόριθμος Gradient Descent βρίσκει την βέλτιστη ευθεία που εφαρμόζεται καλύτερα πάνω στα δεδομένα εκπαίδευσης του αλγορίθμου, ξεκινώντας κάθε φορά με μια τιμή των συντελεστών b_0 και b_1. Συνήθως, η αρχική τιμή από την οποία ξεκινούν οι συντελεστές αυτοί είναι η τιμή 0. Σε κάθε επανάληψη ο αλγόριθμος διαφοροποιεί τις διάφορες τιμές των συντελεστών με βάση ενός «ρυθμού εκμάθησης (Learning Rate) ώστε να προσδιοριστεί για ποια τιμή των συντελεστών του μοντέλου παλινδρόμησης ελαχιστοποιείται η τιμή της συνάρτησης κόστους. Ο ρυθμός εκμάθησης αντιπροσωπεύει την ταχύτητα με την οποία ο αλγόριθμος εκπαιδεύεται. Δεν πρέπει ο ρυθμός εκμάθησης να έχει πολύ υψηλή τιμή διότι τότε ο αλγόριθμος Gradient Descent μπορεί είτε να μην συγκλίνει ποτέ στην ελάχιστη τιμή της συνάρτησης κόστους είτε να αποκλίνει αρκετά από αυτήν. Επιπλέον, δεν πρέπει να έχει ούτε πολύ χαμηλή τιμή διότι τότε ο αλγόριθμος θα είναι αρκετά αργός για να υλοποιήσει την όλη επαναληπτική διαδικασία. Το όνομα της μεθόδου έγκειται στο γεγονός ότι η τιμή της συνάρτησης κόστους έχει μια πτωτική πορεία (κατηφορική κλίση) αναλόγως την επανάληψη που βρίσκεται ο αλγόριθμος και των τιμών που έχουν σε κάθε επανάληψη οι συντελεστές. Παρακάτω φαίνεται σε ένα διάγραμμα η λειτουργία της τεχνικής Gradient Descent δηλαδή το πως μειώνεται η τιμή της συνάρτησης κόστους όσο αυξάνονται οι επαναλήψεις που εκτελεί ο αλγόριθμος για την εύρεση της βέλτιστης εκτίμησης των συντελεστών του μοντέλου παλινδρόμησης.

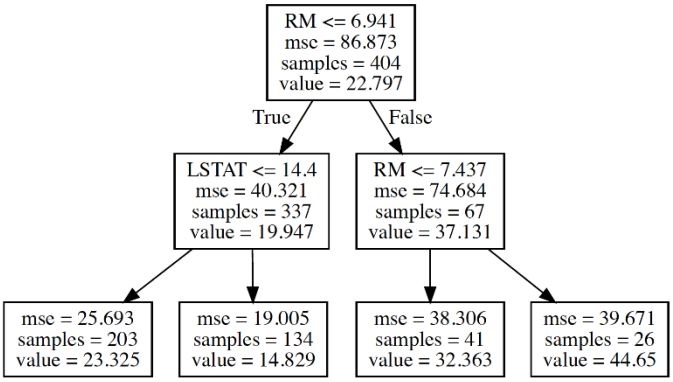


Σχήμα 5: Διαγραμματική απεικόνιση της μεθόδου παλινδρόμησης Gradient Descent για την ελαχιστοποίηση της συνάρτησης κόστους

3.3.7 Παλινδρόμηση με την χρήση Decision Trees (Decision Trees Regression)

Ο αλγόριθμος με δέντρα απόφασης (Decision Trees) πρόκειται για έναν αλγόριθμο επιτηρούμενης μάθησης ο οποίος μπορεί να εφαρμοστεί και για επίλυση προβλημάτων κατηγοριοποίησης των δεδομένων αλλά και σε προβλήματα παλινδρόμησης (The Elements of Statistical Learning, 2001) (Εφαρμοσμένη στατιστική και στατιστική μηχανική μάθηση, 2021) (Analytics Vidhya). Τα δέντρα αποφάσεων είναι παρόμοια με τα διαγράμματα ροής και σκοπός τους είναι φυσικά η πρόβλεψη των τιμών της μεταβλητής απόκρισης με βάση των τιμών των ανεξάρτητων μεταβλητών. Αυτή η πρόβλεψη επιτυγχάνεται με την χρήση ενός δυαδικού δέντρου

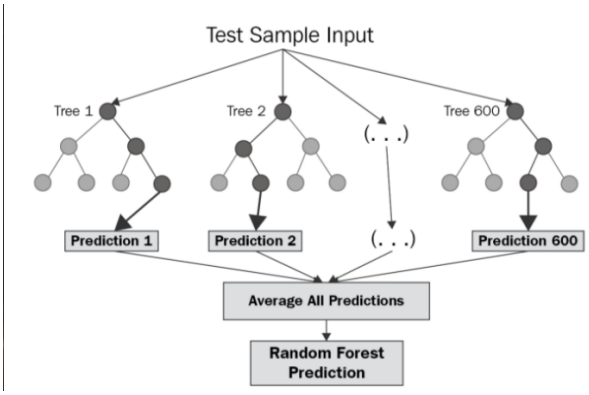
το οποίο περιέχει τον ριζικό κόμβο (root node), τους εσωτερικούς κόμβους ή κόμβους απόφασης (nodes ή decision nodes), τις διακλαδώσεις (branches) και τους τερματικούς κόμβους (end nodes) ή εναλλακτικά τους κόμβους φύλλου (leaf nodes) οι οποίοι είναι τα τερματικά σημεία του αλγορίθμου τα οποία δεν έχουν άλλες διακλαδώσεις. Με άλλα λόγια οι κόμβοι φύλλου περιέχουν τα αποτελέσματα της εξαρτημένης μεταβλητής. Στον ριζικό κόμβο περιέχεται το αρχικό σύνολο δεδομένων από το οποίο ο αλγόριθμος επιλέγει μια ανεξάρτητη μεταβλητή, την πιο σημαντική για την ακρίβεια, για να ξεκινήσει την όλη διαδικασία ελέγχων και στην συνέχεια αυτός ο ριζικός κόμβος διασπάται σε δύο ή σε περισσότερες διακλαδώσεις οι οποίες διακλαδώσεις περιέχουν τις τιμές εκείνης της μεταβλητής που εξετάζεται. Σε κάθε εσωτερικό κόμβο πραγματοποιείται έλεγχος μιας ανεξάρτητης μεταβλητής μόνο, και αναλόγως το αποτέλεσμα αυτού του ελέγχου, αν είναι δηλαδή «Αληθές» ή «Ψευδές», όπως λειτουργούν δηλαδή οι συνθήκες στα διαγράμματα ροής, ο αλγόριθμος συνεχίζει είτε στην αριστερή είτε στην δεξιά διακλάδωση αντίστοιχα του δέντρου. Όσον αφορά τον έλεγχο στους εσωτερικούς κόμβους, γίνεται σύγκριση και λαμβάνεται η απόφαση, για το ποιες μεταβλητές είναι σημαντικές για την πρόβλεψη των τιμών της μεταβλητής απόκρισης. Υπολογίζεται και συγκρίνεται λοιπόν, σε κάθε κόμβο το άθροισμα τετραγώνων των εκτιμημένων σφαλμάτων (SSE) ή το μέσο τετραγωνικό σφάλμα (MSE). Τελικά η μεταβλητή που έχει το ελάχιστο άθροισμα τετραγώνων των σφαλμάτων ή αντίστοιχα εκείνη που έχει το ελάχιστο μέσο τετραγωνικό σφάλμα είναι εκείνη που αποτελεί τον κόμβο απόφασης (decision node) ο οποίος με την σειρά του θα διασπαστεί σε άλλες διακλαδώσεις και η διαδικασία αυτή θα συνεχιστεί μέχρις ότου ολοκληρωθούν όλοι οι απαραίτητοι έλεγχοι.. Αφού ολοκληρωθούν οι έλεγχοι για όλες τις μεταβλητές που είναι σημαντικές, ο αλγόριθμος καταλήγει στον τελικό κόμβο φύλλου όπου γίνεται η πρόβλεψη των τιμών της μεταβλητής απόκρισης. Το μεγάλο πλεονέκτημα αυτού του αλγορίθμου είναι ότι απεικονίζει την όλη διαδικασία μέχρι τα τελικά αποτελέσματα της μεταβλητής απόκρισης οπότε μέσω αυτής της μεθόδου διακρίνονται οι μεταβλητές που χρησιμοποιήθηκαν και μπορούν να απεικονιστούν όλες οι εναλλακτικές λύσεις για το εκάστοτε πρόβλημα. Οι περιορισμοί τις μεθόδου είναι ότι δεν πρέπει να δημιουργούνται πολύ μικρά ή πολύ μεγάλα δέντρα γιατί αυτό υποδηλώνει είτε ότι δεν χρησιμοποιήθηκαν αρκετές από τις σημαντικότερες μεταβλητές του συνόλου δεδομένων είτε αντίστοιχα ότι όταν εμφανίζεται ένα αρκετά μεγάλο δέντρο, να υπάρχει πρόβλημα υπερπροσαρμογής των δεδομένων (overfitting). Αυτό το πρόβλημα των πολύ μεγάλων δέντρων όπου μπορεί να υπάρχει υπερπροσαρμογή των δεδομένων μπορεί να λυθεί με την χρήση ενός παρόμοιου αλγορίθμου με αυτόν των δέντρων αποφάσεων ο οποίος θα παρουσιαστεί στην επόμενη υπό ενότητα. Παρακάτω στο σχήμα 6 παρουσιάζεται η μορφή ενός δέντρου αποφάσεων για την πρόβλεψη μια συνεχούς μεταβλητής Y.



Σχήμα 6: Αναπαράσταση διαγραμματική του δέντρου αποφάσεων
Πηγή σχήματος: (https://stackoverflow.com/questions/59375220/how-to-get-the-mse-of-the-node-in-the-decisiontreeregressor-of-scikit-learn)

3.3.8 Παλινδρόμηση με την χρήση της τεχνικής Random Forests (Random Forrest Regression)

Η τεχνική παλινδρόμησης με τυχαία δάση (Random Forests) πρόκειται για μια συνδυαστική μέθοδο η οποία χρησιμοποιώντας πολλαπλά τυχαία δάση παράλληλα με μια άλλη τεχνική που ονομάζεται «bagging» προσπαθεί να επιλύσει το πρόβλημα υπεπροσαρμογής των δεδομένων που αντιμετωπίζει ο αλγόριθμος των δέντρων αποφάσεων, με στόχο την καλύτερη, πιο ακριβή και πιο αξιόπιστη πρόβλεψη των τιμών της μεταβλητής απόκρισης (Analytics Vidhya) (The Elements of Statistical Learning, 2001). Στην ουσία, η διαδικασία που ακολουθεί ο αλγόριθμος είναι η εξής. Δημιουργεί πολλαπλά δέντρα αποφάσεων το καθένα από τα οποία λειτουργεί ξεχωριστά από τα υπόλοιπα δέντρα και κάνει πρόβλεψη των τιμών της εξαρτημένης μεταβλητής με βάση τις μεταβλητές και τα κριτήρια που επέλεξε να κάνει αυτήν την πρόβλεψη. Ξεκινώντας από το αρχικό σύνολο δεδομένων, επιλέγει τυχαία υποσύνολα όλων των μεταβλητών τα οποία χρησιμοποιούνται στους κόμβους των διαφόρων δέντρων αποφάσεων που δημιουργούνται μέχρι το εκάστοτε δέντρο απόφασης να φτάσει στον τερματικό του κόμβο. Η επιλογή διαφορετικού υποσυνόλου μεταβλητών λύνει το πρόβλημα που αντιμετωπίζει η τεχνική decision trees όπου επειδή επιλέγονται οι σημαντικότερες μεταβλητές για τους εσωτερικούς κόμβους του δέντρου, τα δέντρα που δημιουργούνται είναι πολύ συσχετισμένα και συνεπώς θα δίνουν όμοια αποτελέσματα κάθε φορά. Έπειτα χρησιμοποιεί την μέθοδο «bagging» για να μειώσει την υψηλή διακύμανση που ενδέχεται να εμφανίζει ο αλγόριθμος decision trees, με στόχο να βρει το μέσο αποτέλεσμα όλων των προβλέψεων που καταλήγουν τα πολλαπλά δημιουργημένα δέντρα αποφάσεων για να το χρησιμοποιήσει ως τελικό αποτέλεσμα πρόβλεψης των τιμών της μεταβλητής απόκρισης. Παρότι ο αλγόριθμος των τυχαίων δασών λειτουργεί καλά για πολύ μεγάλα σύνολα δεδομένων και αποφεύγει τις περισσότερες φορές το πρόβλημα υπερπροσαρμογής, η χρήση του αποφεύγεται όταν υπάρχουν στον σύνολο δεδομένων κατηγορικές μεταβλητές με πολλά επίπεδα, διότι ο αλγόριθμος μεροληπτεί υπέρ των κατηγορικών μεταβλητών που έχουν τα περισσότερα επίπεδα.



Σχήμα 7: Διαγραμματική αναπαράσταση του αλγορίθμου παλινδρόμησης «Random Forest»
Πηγή σχήματος: (https://levelup.gitconnected.com/random-forest-regression-209c0f354c84)

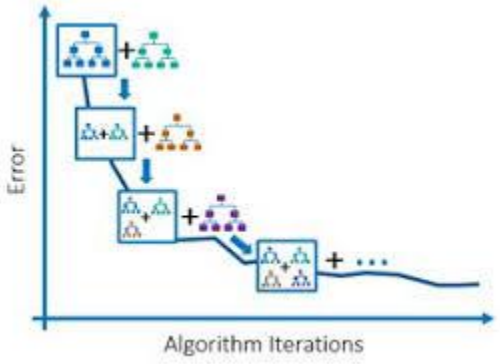
3.3.9 Παλινδρόμηση με την χρήση της τεχνικής Extra Tree Regression (Extra Tree Regression)

Η τεχνική παλινδρόμησης Extra Tree (Extremely Randomized Trees), πρόκειται για μια πολύ παρόμοια τεχνική με αυτή των τυχαίων δασών με κάποιες διαφορές φυσικά (Baeldung, 2023) (QuantDare, 2020) (Extremely randomized trees, 2006). Όπως ο αλγόριθμος Random Forests, έτσι και ο αλγόριθμος Extra Trees χρησιμοποιεί πολλαπλά δέντρα αποφάσεων για την πρόβλεψη των τιμών της εξαρτημένης μεταβλητής του συνόλου δεδομένων. Η διαφοροποίηση της συγκεκριμένης μεθόδου παλινδρόμησης σε σχέση με την μέθοδο τυχαίων δασών έγκειται στο γεγονός ότι εδώ ο αλγόριθμος Extra Trees χρησιμοποιεί ολόκληρο το σύνολο δεδομένων σε αντίθεση με τον αλγόριθμο Random Forests ο οποίος χρησιμοποιεί ένα υποσύνολο του συνόλου δεδομένων. Κάθε δέντρο που σχηματίζεται περιέχει ένα μοναδικό δείγμα το οποίο δεν γίνεται να εμφανιστεί στα άλλα δέντρα που δημιουργούνται γεγονός που κάνει ακόμα πιο ασυσχέτιστα τα δέντρα. Επίσης και οι μεταβλητές που επιλέγονται για κάθε δέντρο, επιλέγονται με τυχαίο τρόπο. Η δεύτερη κύρια διαφορά των δύο μεθόδων είναι ότι η μέθοδος Extra Trees επιλέγει την τιμή διαχωρισμού των μεταβλητών με τυχαίο τρόπο και όχι με βάση κάποιο κριτήριο που βρίσκει μια βέλτιστη τοπική τιμή διαχωρισμού όπως κάνει η μέθοδος Random Forests για την διάσπαση των εσωτερικών κόμβων σε διακλαδώσεις. Αυτή η διαδικασία των Extra Trees τείνει να κάνει τα δέντρα που δημιουργούνται πιο ασυσχέτιστα και να μειώνεται ακόμη περισσότερο η διακύμανση κάνοντας πιο πιθανή την αποφυγή του φαινομένου της υπερπροσαρμογής των δεδομένων. Σε πρακτικά προβλήματα, οι δύο μέθοδοι δίνουν σχεδόν τα ίδια αποτελέσματα αλλά ο αλγόριθμος Extra Tree φαίνεται να εκτελείται πιο γρήγορα σε σχέση με τον αλγόριθμο Random Forests.

3.3.10 Παλινδρόμηση με την χρήση της τεχνικής Gradient Boosting (Gradient Boosting Regression)

Η τεχνική παλινδρόμησης «Gradient Boosting» πρόκειται για ένα σύνολο από αδύναμα μοντέλα πρόβλεψης μηχανικής μάθησης τα οποία χρησιμοποιούνται συνδυαστικά για να επιτευχθεί η μέγιστη και καλύτερη δυνατή απόδοση συνολικά (The Elements of Statistical Learning, 2001) (TowardsDatascience, 2022). Ένας σταθερός αριθμός πολλαπλών δέντρων αποφάσεων χρησιμοποιείται στον αλγόριθμο Gradient Boosting όπου τα δέντρα αυτά θεωρούνται

αδύναμοι «μαθητές» ή αδύναμα μοντέλα. Η μέθοδος αυτή λειτουργεί επαναληπτικά όπου σε κάθε επανάληψη δημιουργείται ένα δέντρο αποφάσεων το οποίο στοχεύει στην πρόβλεψη της τιμής της εξαρτημένης μεταβλητής. Σε κάθε επανάληψη υπολογίζεται η διαφορά μεταξύ της εκτιμημένης τιμής και της πραγματικής τιμής της μεταβλητής απόκρισης δηλαδή το υπόλοιπο (residual). Κάθε νέο δέντρο που δημιουργείται σε κάθε επανάληψη του αλγορίθμου Gradient Boosting λαμβάνει την πληροφορία από το προηγούμενο δέντρο απόφασης και την χρησιμοποιεί για να ελαχιστοποιήσει το μέσο τετραγωνικό σφάλμα των προηγούμενων αδύναμων μοντέλων (δέντρων αποφάσεων). Η μέθοδος αυτή σταματά όταν το μέσο τετραγωνικό σφάλμα δεν μπορεί να ελαττωθεί άλλο δηλαδή όταν έχει βελτιστοποιηθεί η συνάρτηση κόστους ή όταν έχει πραγματοποιηθεί ο μέγιστος αριθμός επαναλήψεων του μοντέλου Gradient Boosting.



Σχήμα 8: Διαγραμματική αναπαράσταση του αλγορίθμου παλινδρόμησης «Random Forest»
Πηγή σχήματος: (https://www.cse.chalmers.se/~richajo/dit866/files/gb_explainer.pdf)

Ειδικές κατηγορίες της μεθόδου «Gradient Boosting» αποτελούν οι αλγόριθμοι:

1. XGboost
2. LightGBM
3. CatBoost
4. AdaBoost

Οι παραπάνω αλγόριθμοι μπορούν να εφαρμοστούν και σε προβλήματα κατηγοριοποίησης των δεδομένων όμως δεν θα αναλυθούν περεταίρω στα πλαίσια της παρούσας διπλωματικής εργασίας.

3.3.11 Μέτρα αξιολόγησης μοντέλων παλινδρόμησης

Για την αξιολόγηση των μοντέλων παλινδρόμησης, χρησιμοποιούνται κάποια στατιστικά μέτρα τα οποία αναλόγως την τιμή που έχουν, δείχνουν αν ένα μοντέλο πρόβλεψης έχει καλή ακρίβεια όσον αφορά τις προβλέψεις του. Η ακρίβεια των μοντέλων μελετάται μέσω των υπολοίπων τους δηλαδή της διαφοράς μεταξύ πραγματικών τιμών της μεταβλητής απόκρισης και των προβλεπόμενων τιμών. Τα πιο γνωστά μέτρα αξιολόγησης των μοντέλων παλινδρόμησης είναι τα:

- Συντελεστής προσδιορισμού R^2 «Coefficient of determination»
- Μέσο τετραγωνικό σφάλμα «Mean Squared Error (MSE)»
- Ρίζα μέσου τετραγωνικού σφάλματος «Root Mean Squared Error (RMSE)»
- Μέσο απόλυτο σφάλμα «Mean Absolute Error (MAE)»

- Μέσο απόλυτο ποσοστό σφάλματος «Mean Absolute Percentage Error (MAPE)»

Συντελεστής προσδιορισμού R^2

Ο συντελεστής προσδιορισμού R^2 δίνεται από την σχέση:

$$R^2 = \frac{SSR}{SST}$$

Το «SSR» είναι το άθροισμα τετραγώνων λόγω της παλινδρόμησης (Regression Sum of Squares) και εκφράζει την μεταβλητότητα που ερμηνεύεται από το μοντέλο παλινδρόμησης που έχουμε προσαρμόσει. Το «SST» είναι ολική μεταβλητότητα του μοντέλου παλινδρόμησης. Γενικά για να είναι ικανοποιητικό ένα μοντέλο παλινδρόμησης θα πρέπει η τιμή του συντελεστή προσδιορισμού R^2 να είναι όσο το δυνατόν πιο κοντά στο 1 γίνεται. Ο συντελεστής αυτός παίρνει τιμές μεταξύ 0 και 1.

Μέσο τετραγωνικό σφάλμα (MSE)

Το μέσο τετραγωνικό σφάλμα ορίζεται ως:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 = \frac{SSE}{n},$$

και είναι ο μέσος όρος της τετραγωνικής διαφοράς της πραγματικής τιμής με την προβλεπόμενη τιμή του μοντέλου παλινδρόμησης που έχει εφαρμοστεί. Γενικά μικρότερες τιμές του μέσου τετραγωνικού σφάλματος υποδηλώνουν καλύτερη προσαρμογή το μοντέλου παλινδρόμησης που έχει προσαρμοστεί στα δεδομένα. Το μειονέκτημα του MSE είναι ότι επηρεάζεται η τιμή του από την ύπαρξη παράτυπων σημείων (outliers) στα δεδομένα, δηλαδή αν εντοπιστούν παράτυπα σημεία τότε το ΜΣΕ επιβάλλει μια «ποινή» και μεγαλώνει η τιμή του .

Ρίζα μέσου τετραγωνικού σφάλματος (RMSE)

Η ρίζα του μέσου τετραγωνικού σφάλματος ορίζεται ως:

$$RMSE = \sqrt{MSE}$$

Το συγκεκριμένο μέτρο αξιολόγησης χρησιμοποιείται αρκετά σε μοντέλα «deep learning» όπως τα νευρωνικά δίκτυα. Η τιμή του RMSE που υπολογίζεται είναι στην ίδια μονάδα μέτρησης με την τιμή της μεταβλητής απόκρισης σε αντίθεση με το MSE όπου η τιμή του είναι η μετρημένη με την μονάδα μέτρησης της μεταβλητής απόκρισης υψωμένη στο τετράγωνο.

Μέσο απόλυτο σφάλμα (MAE)

Το μέσο απόλυτο σφάλμα υπολογίζει τις διαφορές μεταξύ προβλεπόμενων και πραγματικών τιμών της μεταβλητής απόκρισης και ορίζεται ως:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

Η τιμή του MAE που υπολογίζεται είναι στην ίδια μονάδα μέτρησης με την τιμή της μεταβλητής απόκρισης και επίσης πρόκειται για ένα μέτρο αξιολόγησης που είναι ανθεκτικό στην ύπαρξη

παράτυπων σημείων. Το μειονέκτημα του συγκεκριμένου μέτρου αξιολόγησης είναι ότι δεν μπορεί να συγκρίνει την αποδοτικότητα των μοντέλων παλινδρόμησης για διαφορετικές κατηγορίες δεδομένων. Όσο μεγαλύτερη τιμή έχει το μέσο απόλυτο σφάλμα τόσο χειρότερη είναι η προσαρμογή του μοντέλου στα δεδομένα και συνεπώς η απόδοσή του.

Μέσο απόλυτο ποσοστό σφάλματος (MAPE)

Το μέσο απόλυτο ποσοστό σφάλματος εκφράζει την ακρίβεια του μοντέλου που προσαρμόστηκε σαν ποσοστό και ορίζεται ως:

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right|$$

Επίσης μπορεί να συγκρίνει την αποδοτικότητα των μοντέλων για διαφορετικές κατηγορίες δεδομένων. Όσο μικρότερο είναι το ποσοστό του συγκεκριμένου μέτρου αξιολόγησης τόσο καλύτερη είναι η ακρίβεια του μοντέλου που προσαρμόστηκε.

3.4 Η μέθοδος της Κατηγοριοποίησης (Classification method)

Η τεχνική της κατηγοριοποίησης πρόκειται για μια επιτηρούμενη τεχνική μηχανικής μάθησης που έχει σκοπό να κατατάξει τα δεδομένα σε κάποιες προκαθορισμένες κατηγορίες οι οποίες μπορεί να είναι δύο ή και παραπάνω. Ο λόγος που η κατηγοριοποίηση ανήκει στην κατηγορία «Supervised Learning» είναι ότι μπορεί να εκπαιδευτεί ο αλγόριθμος πάνω στα υπάρχοντα δεδομένα που εμπεριέχουν την κλάση – κατηγορία στην οποία ανήκουν οι παρατηρήσεις και στην συνέχεια να «μαθαίνει» ανάλογα με τις τιμές συγκεκριμένων χαρακτηριστικών την κλάση που κατατάσσονται αυτές οι παρατηρήσεις. Με αυτόν τον τρόπο, αφού θα έχει εκπαιδευτεί ο αλγόριθμος στην πρόβλεψη κατηγορίας για το σύνολο των τιμών των χαρακτηριστικών των διαθέσιμων δεδομένων, όταν θα εισάγονται καινούργια δεδομένα ο αλγόριθμος κατηγοριοποίησης θα είναι σε θέση ανάλογα με τις τιμές των χαρακτηριστικών να προβλέπει όσο πιο ορθά και αξιόπιστα γίνεται την κατηγορία στην οποία ανήκουν οι καινούργιες παρατηρήσεις. Στα πλαίσια της παρούσας διπλωματικής εργασίας θα παρουσιαστούν μερικές από τις πιο ευρέως γνωστές μέθοδοι κατηγοριοποίησης.

3.4.1 Γραμμική διαχωριστική ανάλυση (Linear Discriminant Analysis «LDA»)

Σκοπός της μεθόδου Γραμμικής διαχωριστικής ανάλυσης είναι δημιουργώντας κάποιους κανόνες να κατατάξει τις παρατηρήσεις σε μία κατηγορία από τις Κ διαθέσιμες με βάση τις τιμές διαφόρων χαρακτηριστικών που υπάρχουν στο σύνολο δεδομένων (Data Mining and Analysis, 2014) (The Elements of Statistical Learning, 2001). Βασικές προϋποθέσεις για την εφαρμογή της είναι ότι για τις Κ κατηγορίες είναι γνωστές οι κατανομές τους και ότι οι παρατηρήσεις λαμβάνουν συνεχείς τιμές. Ο κύριος στόχος της μεθόδου είναι να προβάλει τα χαρακτηριστικά (μεταβλητές) ενός υψηλότερου χώρου διαστάσεων, σε έναν χώρο με μικρότερες διαστάσεις χρησιμοποιώντας έναν κανόνα. Για να το επιτύχει αυτό θα πρέπει ο κανόνας που θα βρεθεί να μπορεί να μεγιστοποιήσει την απόσταση μεταξύ των Κ κατηγοριών και να ελαχιστοποιήσει την

διακύμανση εντός των κατηγοριών. Η διασπορά μεταξύ των κατηγοριών η οποία πρόκειται για την απόσταση των μέσων τιμών των κατηγοριών μπορεί να υπολογιστεί ως εξής:

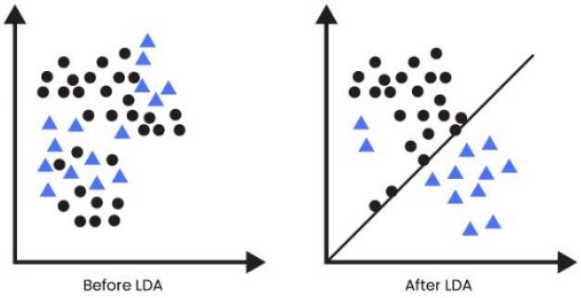
$$S_b = \sum_{i=1}^{k} N_i \ (\bar{x}_i - \ \bar{x})(\bar{x}_i - \ \bar{x})^T$$

Ύστερα υπολογίζεται η διασπορά εντός των κλάσεων δηλαδή η απόσταση μεταξύ του μέσου και της παρατήρησης της κάθε κατηγορίας ως ακολούθως:

$$S_w = \sum_{i=1}^{k} \sum_{j=1}^{N} (x_{\mathrm{ij}} - \bar{x}_i) \ (x_{\mathrm{ij}} - \ \bar{x}_i)^T$$

Στο τελικό στάδιο πραγματοποιείται η προβολή του χώρου σε χαμηλότερη διάσταση που μεγιστοποιεί την διακύμανση μεταξύ των κατηγοριών και ελαχιστοποιεί την διακύμανση εντός των κατηγοριών. Αυτή η προβολή της χαμηλότερης διάστασης του χώρου που συμβολίζεται με U, ονομάζεται διαχωριστικός κανόνας ή κριτήριο του Fisher και φαίνεται παρακάτω:

$$U_{LDA} = \frac{U^T S_b U}{U^T S_w U}$$



Σχήμα 9: : Γραφική αναπαράσταση της λειτουργίας της μεθόδου Linear Discriminant Analysis (Αριστερά: πριν την LDA, Δεξιά: Μετά την LDA)
Πηγή:(https://www.analyticssteps.com/blogs/introduction-linear-discriminant-analysis-supervised-learning)

3.4.2 Μέθοδος Λογιστικής παλινδρόμησης (Logistic Regression)

Η λογιστική παλινδρόμηση πρόκειται για μια τεχνική παλινδρόμησης αλλά μπορεί να χρησιμοποιηθεί και ως τεχνική κατηγοριοποίησης των δεδομένων σε δύο κατηγορίες. Σε αντίθεση με τα μοντέλα απλή γραμμικής παλινδρόμησης και πολλαπλής παλινδρόμησης, η λογιστική παλινδρόμηση χρησιμοποιείται όταν η εξαρτημένη μεταβλητή είναι δίτιμη, δηλαδή όταν λαμβάνει μόνο δύο τιμές. Στην ναυτιλία για παράδειγμα, η εξαρτημένη μεταβλητή θα μπορούσε να αντιπροσωπεύει την κατηγορία ενός πλοίου όσον αφορά τις εκπομπές διοξειδίου του άνθρακα στην ατμόσφαιρα, δηλαδή το αν ένα πλοίο είναι ρυπογόνο (τιμή 1) ή μη ρυπογόνο (τιμή 0) ανάλογα με την ποσότητα διοξειδίου του άνθρακα που απελευθερώνει στην ατμόσφαιρα κατά την διάρκεια των ταξιδιών του. Σκοπός λοιπόν της λογιστικής παλινδρόμησης είναι να περιγράψει την σχέση που συνδέει την πιθανότητα να συμβεί ή να μην συμβεί ένα γεγονός για μια μεταβλητή (δίτιμη εξαρτημένη μεταβλητή), με βάση διάφορα άλλα χαρακτηριστικά (ανεξάρτητες μεταβλητές). Η μέση τιμή μιας δίτιμης μεταβλητής αντιστοιχεί στην «επιτυχία» εμφάνισης ενός γεγονότος για το χαρακτηριστικό που μελετάται (π.χ. ένα πλοίο να είναι

ρυπογόνο, αφού η εξαρτημένη μεταβλητή λαμβάνει την τιμή 1 σε μια τέτοια περίπτωση). Δηλαδή ισχύει ότι:

$$p = P(Y = 1)$$

Για να προσαρμοστεί ένα μοντέλο λογιστικής παλινδρόμησης που να περιγράφει την σχέση πιθανότητας επιτυχίας μιας μεταβλητής με βάση διάφορες άλλες παραμέτρους θα μπορούσε να θεωρηθεί η κάτωθι εξίσωση:

$$p_i = \beta_0 + \sum_{i=1}^{p} \beta_i X_i,$$

όπου η πιθανότητα επιτυχίας p έχει στην ουσία αντικαταστήσει την μεταβλητή απόκρισης Y. Παρόλα αυτά το παραπάνω μοντέλο μπορεί να πάρει οποιαδήποτε τιμή στο σύνολο των πραγματικών αριθμών όμως μια «πιθανότητα» όπως είναι γνωστό από την θεωρία πιθανοτήτων, μπορεί να πάρει τιμή μόνο στο διάστημα [0,1]. Ένα βελτιωμένο μοντέλο που θα μπορούσε να προσαρμοστεί είναι το ακόλουθο:

$$p_i = e^{\beta_0 + \sum_{i=1}^{p} \beta_i X_i}$$

Και πάλι όμως η πιθανότητα επιτυχίας σε αυτήν την περίπτωση μπορεί να πάρει τιμή μεγαλύτερη της μονάδας, οπότε τελικά το μοντέλο λογιστικής παλινδρόμησης που πρέπει να προσαρμοστεί περιγράφεται με την ακόλουθη σχέση:

$$p_i = \frac{e^{\beta_0 + \sum_{i=1}^{p} \beta_i X_i}}{1 + e^{\beta_0 + \sum_{i=1}^{p} \beta_i X_i}}$$

Επιπλέον ισχύει ότι:

$$\frac{p_i}{1 - p_i} = e^{\beta_0 + \sum_{i=1}^{p} \beta_i X_i}$$

Τέλος, λογαριθμίζοντας την παραπάνω σχέση, προκύπτει ότι:

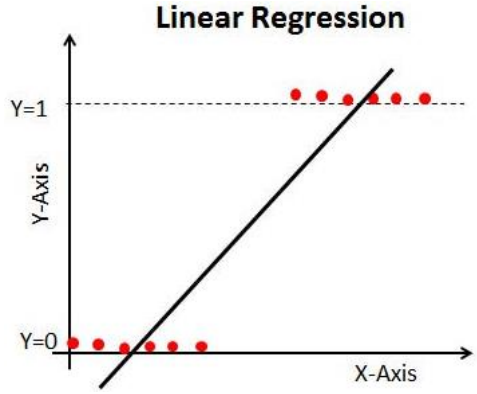
$$log\left(\frac{p_i}{1 - p_i}\right) = \beta_0 + \sum_{i=1}^{p} \beta_i X_i$$

όπου η ποσότητα $\frac{p_i}{1 - p_i}$ είναι ο λόγος συμπληρωματικών πιθανοτήτων (odds ratio) και δείχνει το πόσο πιο πιθανό είναι να συμβεί ένα γεγονός από το να μην συμβεί.

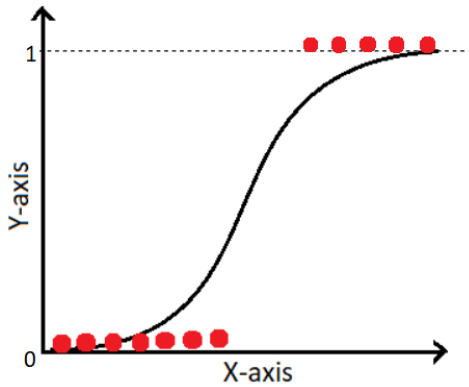
Όσον αφορά τις εκτιμήσεις των συντελεστών παλινδρόμησης των παραμέτρων του μοντέλου, χρησιμοποιείται η μέθοδος μέγιστης πιθανοφάνειας (maximum likelihood method) με λογιστική μορφή.

$$l = logL = log\{ \prod_{i=1}^{n} p_i^{Y_i} (1 - p_i^{Y_i})\},$$

όπου $l = ln(\beta_0, \beta_1)$. Οι εκτιμήσεις των συντελεστών β_0, β_1 καλούνται εκτιμητές μέγιστης πιθανοφάνειας.



Σχήμα 10: Σύγκριση γραμμικής παλινδρόμησης και λογιστικής παλινδρόμησης (διάγραμμα γραμμικής παλινδρόμησης)
Πηγή σχήματος: (https://medium.datadriveninvestor.com/logistic-regression-18afd48779ce)



Σχήμα 11: Σύγκριση γραμμικής παλινδρόμησης και λογιστικής παλινδρόμησης (διάγραμμα Λογιστικής παλινδρόμησης)
Πηγή: (https://medium.com/analytics-vidhya/the-math-behind-logistic-regression-c2f04ca27bca)

3.4.3 Αφελής Κατηγοριοποιητής Bayes (Naive Bayes Classification)

Η τεχνική κατηγοριοποίησης Naive Bayes, χρησιμοποιεί το θεώρημα του Bayes έτσι ώστε να κατατάξει τα δεδομένα σε γνωστές κατηγορίες και για αυτό πρόκειται για πιθανοτική μέθοδο κατηγοριοποίησης (Data Mining and Analysis, 2014) (The Elements of Statistical Learning, 2001). Ο κατηγοριοποιητής Bayes με την χρήση του κανόνα του Bayes, επιδιώκει να προβλέψει ότι η κατηγορία που ανήκει μια παρατήρηση είναι εκείνη η κατηγορία που μεγιστοποιεί την εκ των υστέρων πιθανότητα (posterior probability). Οι βασικές υποθέσεις που κάνει ο αλγόριθμος κατηγοριοποίησης Naive Bayes είναι ότι οι μεταβλητές είναι ανεξάρτητες μεταξύ τους δηλαδή δεν υπάρχουν ζευγάρια μεταβλητών που να εμφανίζουν συσχέτιση και κάθε μεταβλητή συμβάλει το ίδιο σημαντικά στην πρόβλεψη της τιμής της εξαρτημένης μεταβλητής και κατ' επέκταση της κατηγορίας που ανήκουν τα δεδομένα. Παρότι οι υποθέσεις του απλοϊκού κατηγοριοποιητή Bayes φαίνονται μη ρεαλιστικές σε προβλήματα του πραγματικού κόσμου, λειτουργεί αρκετά αποτελεσματικά και δίνει αξιόπιστες εκτιμήσεις. Το θεώρημα του Bayes, με απλά λόγια, υπολογίζει την πιθανότητα να συμβεί ένα γεγονός A, δοθέντος ότι έχει ήδη πραγματοποιηθεί ένα άλλο γεγονός B. Το θεώρημα του Bayes διατυπώνεται ως ακολούθως:

$$P(A \mid B) = \frac{P(B \mid A)\, P(A)}{P(B)}$$

Η παραπάνω σχέση, στο πρόβλημα κατηγοριοποίησης μπορεί να επαναδιατυπωθεί ως εξής:

$$P(\kappa_i \mid x) = \frac{P(x \mid \kappa_i)\, P(\kappa_i)}{P(x)},$$

όπου κ_i είναι η i κατηγορία που ανήκει μια παρατήρηση x από το σύνολο δεδομένων που περιέχει n παρατηρήσεις x_i. Πιο συγκεκριμένα, από τον παραπάνω τύπο ισχύει ότι:

$P(\kappa_i \mid x)$: είναι η εκ των υστέρων πιθανότητα

$P(\mathbf{x} \mid \kappa_i)$: είναι η πιθανότητα να παρατηρηθεί η x παρατήρηση με γνώμονα ότι η πραγματική κατηγορία είναι η κ_i δηλαδή πρόκειται για την πιθανοφάνεια.

$P(\kappa_i)$: δείχνει την εκ των προτέρων πιθανότητα της κατηγορίας κ_i.

$P(\mathbf{x})$: αντιπροσωπεύει την πιθανότητα να παρατηρηθεί η x παρατήρηση από οποιαδήποτε κατηγορία από τις κ_i (probability of evidence). Δηλαδή:

$$P(x) = \sum_{j=1}^{l} P(x \mid \kappa_j)\, P(\kappa_j)$$

Τέλος, η προβλεπόμενη κατηγορία για την x παρατήρηση φαίνεται στην παρακάτω σχέση:

$$\hat{y} = \arg max_{\kappa_i} \{P(\kappa_i \mid x)\}$$

Αξίζει να σημειωθεί ότι υπάρχουν τρεις ακόμη γνωστοί τύποι μοντέλου κατηγοριοποίησης Naive Bayes οι οποίοι διακρίνονται παρακάτω:

- Gaussian: Υποθέτει ότι οι ανεξάρτητες μεταβλητές ακολουθούν κανονική κατανομή
- Multinomial: Χρησιμοποιείται κυρίως για προβλήματα κατηγοριοποίησης εγγράφων και υποθέτει ότι τα δεδομένα ακολουθούν πολυωνυμική κατανομή.
- Bernoulli: Λειτουργεί όπως ο κατηγοριοποιητής Multinomial Naive Bayes μόνο που σε αυτήν την περίπτωση οι ανεξάρτητες μεταβλητές είναι τύπου Boolean δηλαδή λαμβάνουν τιμές «ΨΕΥΔΗΣ» ή «ΑΛΗΘΗΣ».

3.4.4 Μέθοδος K- Nearest Neighbors (K-Nearest Neighbors Classification «KNN»)

Ο αλγόριθμος των κ πλησιέστερων γειτόνων πρόκειται για ένα μη παραμετρικό μοντέλο κατηγοριοποίησης και παλινδρόμησης, επιτηρούμενης μηχανικής μάθησης το οποίο δεν κάνει καμία υπόθεση για το σύνολο δεδομένων (Εισαγωγή στην μέθοδο Nearest Neighbor, 2021). Συλλέγοντας δεδομένα από ένα σύνολο δεδομένων εκπαίδευσης, προσπαθεί ο αλγόριθμος KNN να κάνει ο προβλέψεις για την κατηγορία που ανήκουν τα καινούργια δεδομένα. Αυτό το επιτυγχάνει με βάση την ομοιότητα μεταξύ των παρατηρήσεων δηλαδή θεωρεί ότι τα σημεία δεδομένων (data points) που βρίσκονται σε κοντινή απόσταση είναι και πιο όμοια και συνεπώς θα ανήκουν στην ίδια κατηγορία. Πιο συγκεκριμένα στην συγκεκριμένη μέθοδο αποδίδεται μια ετικέτα κατηγορίας για ένα νέο σημείο δεδομένων με βάση την κατηγορία που ανήκουν τα κ πλησιέστερα γειτονικά του σημεία δεδομένων του συνόλου δεδομένων εκπαίδευσης. Για να βρει ο αλγόριθμος της «κ πλησιέστερους γείτονες» για ένα νέο σημείο δεδομένων θα πρέπει να

υπολογίσει την απόσταση μεταξύ αυτού του νέου σημείου δεδομένων και των σημείων δεδομένων που ήδη υπάρχουν σε έναν χώρο n διαστάσεων με p χαρακτηριστικά. Υπάρχουν πολλά μέτρα αποστάσεων μεταξύ των σημείων που υπολογίζονται (Applied Multivariate Statistical Analysis, 2007). Έστω x_i = ($x_{i1}, x_{i2}, x_{i3}, \dots, x_{ip}$) το διάνυσμα των παρατηρήσεων για τα p χαρακτηριστικά που αφορά την i εγγραφή (i = 1,2,…n). Τα πιο ευρέως χρησιμοποιούμενα μέτρα αποστάσεων μεταξύ δύο παρατηρήσεων (εναλλακτικά σημείων δεδομένων) x_i = (x_{i1}, $x_{i2}, x_{i3}, \dots, x_{ip}$) και x_j = ($x_{j1}, x_{j2}, x_{j3}, \dots, x_{jp}$) είναι τα ακόλουθα:

- Ευκλείδεια απόσταση: $d(x_i, x_j) = \sqrt{\sum_{r=1}^{p}(x_{ir} - y_{jr})^2}$

- Απόσταση Mahalanobis: $d(x_i, x_j) = \sqrt{(x_i - x_j)^T S^{-1}(x_i - x_j)}$, όπου S είναι ο θετικά ορισμένος πίνακας διακυμάνσεων – συνδιακυμάνσεων. Χρησιμοποιείται αντί της Ευκλείδειας απόστασης διότι λαμβάνει υπόψη και της συνδιακυμάνσεις ανάμεσα της μεταβλητές.

- Απόσταση Manhattan: $d(x_i, x_j) = \sum_{r=1}^{p} |x_{ir} - x_{jr}|$, η οποία χρησιμοποιείται όταν υπάρχουν παράτυπα σημεία (outliers) στα δεδομένα, διότι είναι πιο ανθεκτική σε αυτά.

- Απόσταση Minkowski: $d(x_i, x_j) = (\sum_{r=1}^{p} |x_{ir} - x_{jr}|^{\varphi})^{1/\varphi}$, όπου φ ≥ 1. Η απόσταση Minkowski είναι μια γενίκευση της απόστασης «Manhattan (φ = 1)» και της «Ευκλείδειας απόστασης (φ = 2)».

- Απόσταση του Chebyshev: $d(x_i, x_j) = max_{r=1,2,..,p} |x_{ir} - x_{jr}|$. Η απόσταση του Chebyshev ισχυρίζεται ότι δύο παρατηρήσεις θεωρούνται διαφορετικές, αν έχουν μεγάλες διαφορές σε τουλάχιστον μια μεταβλητή.

- Απόσταση Bhattacharyya: $d(x_i, x_j) = (\sum_{r=1}^{p}(\sqrt{x_{ir}} - \sqrt{x_{jr}})^2)^{1/2}$. Η απόσταση Bhattacharyya χρησιμοποιείται συνήθως όταν τα δεδομένα αποτελούνται από ποσοστά.

Σχήμα 12: Γραφική αναπαράσταση της λειτουργίας της μεθόδου K-Nearest Neighbors για κ=3
Πηγή σχήματος: (https://vatsalparsaniya.github.io/ML_Knowledge/Nearest_neighbours/Readme.html)

3.4.5 Μέθοδος δέντρων αποφάσεων (Decision Trees Classification)

Η μέθοδος Decision Trees λειτουργεί της έχει αναφερθεί και προηγουμένως και σε προβλήματα κατηγοριοποίησης των δεδομένων σε κλάσσεις. Η δομή του είναι ακριβώς η ίδια με αυτή της δέντρου απόφασης στην περίπτωση της παλινδρόμησης. Δηλαδή το δέντρο αποτελείται από τον ριζικό κόμβο (root node) που αντιπροσωπεύει το πιο σημαντικό χαρακτηριστικό του συνόλου δεδομένων, της εσωτερικούς κόμβους ή κόμβους απόφασης (nodes ή decision nodes), της διακλαδώσεις (branches) και της τερματικούς κόμβους (end nodes) ή εναλλακτικά της κόμβους φύλλου (leaf nodes) οι οποίοι είναι τα τερματικά σημεία του αλγορίθμου τα οποία δεν έχουν της διακλαδώσεις. Οι εσωτερικοί κόμβοι αντιπροσωπεύουν τα χαρακτηριστικά του συνόλου δεδομένων, οι διακλαδώσεις αντιπροσωπεύουν της κανόνες απόφασης και κάθε κόμβος φύλου αντιπροσωπεύει το αποτέλεσμα. Οι εσωτερικοί κόμβοι «απόφασης» χρησιμοποιούνται για την λήψη οποιασδήποτε απόφαση και μπορούν να έχουν πολλαπλούς κλάδους ανάλογα με της τιμές των χαρακτηριστικών, ενώ οι κόμβοι φύλου είναι τα αποτελέσματα, της αναφέρθηκε και προηγουμένως, αυτών των αποφάσεων και δεν έχουν της διακλαδώσεις. Υπάρχουν αρκετοί αλγόριθμοι οι οποίοι κατασκευάζουν ένα δέντρο απόφασης και οι πιο γνωστοί είναι οι:

- ID3
- C4.5
- CART

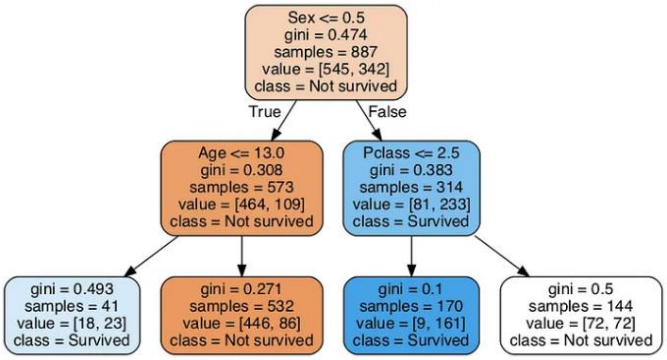
Για να βρεθούν τα πιο σημαντικά χαρακτηριστικά που θα αποτελούν το ριζικό κόμβο και της εσωτερικούς κόμβους αποφάσεων μέχρι την λήψη της τελικής απόφασης για την κατηγορία των δεδομένων, στην περίπτωση της τεχνικής κατηγοριοποίησης, χρησιμοποιούνται διάφορα κριτήρια (Attribute Selection Measures) (tutorials point, n.d.) της τα:

- Information Gain: Στηρίζεται στην έννοια της «εντροπίας» όπου στον κλάδο των μαθηματικών και της φυσικής μετρά την ακαθαρσία (impurity) ή ενναλακτικά την τυχαιότητα σε ένα σύστημα. Στην επιστήμη δεδομένων αναφέρεται στην ακαθαρσία της

συνόλου δεδομένων. Το κριτήριο «Information Gain» έχει σκοπό την μείωση της εντροπίας ξεκινώντας από τον ριζικό κόμβο μέχρι τον τερματικό κόμβο και αυτό το κριτήριο το χρησιμοποιεί ο αλγόριθμος ID3.

- Gain Ratio: Αυτό το κριτήριο χειρίζεται το πρόβλημα μεροληψίας που εμφανίζει το «Inforamtion Gain» υπέρ των μεταβλητών που έχουν μεγάλο αριθμό διακριτών τιμών. Το συγκεκριμένο κριτήριο χρησιμοποιεί ο αλγόριθμος C4.5 για την κατασκευή του δέντρου απόφασης.

- Gini Index: Η μεταβλητή με την ελάχιστη τιμή «Gini» επιλέγεται ως η μεταβλητή διαχωρισμού. Το κριτήριο αυτό χρησιμοποιείται από τον αλγόριθμο CART.



Σχήμα 13: Γραφική απεικόνιση της τεχνικής κατηγοριοποίησης με δέντρο αποφάσεων
Πηγή:(https://towardsdatascience.com/an-introduction-to-decision-trees-with-python-and-scikit-learn-1a5ba6fc204f)

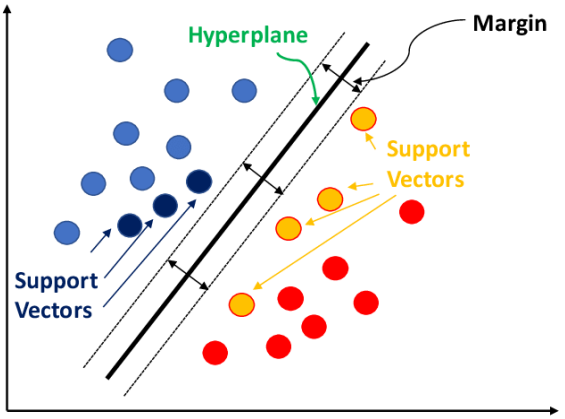
3.4.6 Μέθοδος μηχανών διανυσμάτων υποστήριξης (Support Vector Machines Classifier «SVM»)

Η μέθοδος «Support Vector Machines» πρόκειται για μια μέθοδο κατηγοριοποίησης των δεδομένων και είναι η ίδια σχεδόν μέθοδος με εκείνη που χρησιμοποιείται στην περίπτωση της παλινδρόμησης μόνο που εδώ χρησιμοποιείται για να χωρίσει ή καλύτερα να ταξινομήσει τα δεδομένα σε δύο κατηγορίες. Για το πρόβλημα κατηγοριοποίησης των δεδομένων σε δύο κατηγορίες, ο αλγόριθμος μηχανών διανυσμάτων υποστήριξης έχει σαν κύριο στόχο την εύρεση ενός βέλτιστου διαχωριστικού ορίου δηλαδή ενός υπερεπιπέδου (hyperplane) που να μεγιστοποιεί το περιθώριο μεταξύ των κατηγοριών και να χωρίζει ουσιαστικά έναν χώρο με n διαστάσεις σε κατηγορίες έτσι ώστε όταν εισέλθει ένα καινούργιο σημείο δεδομένων να μπορεί να ταξινομηθεί ορθά και με όσο το δυνατόν πιο υψηλή ακρίβεια στην σωστή κατηγορία. Για την δημιουργία αυτού του υπερεπιπέδου ο αλγόριθμος επιλέγει κάποια σημεία τα οποία βρίσκονται πιο κοντά σε αυτό το υπερεπίπεδο. Αυτά τα ακραία σημεία είναι τα διανύσματα υποστήριξης, από τα οποία προκύπτει και η ονομασία του συγκεκριμένου μοντέλου. Υπάρχουν δύο κύρια είδη περιπτώσεων του μοντέλου SVM. Αυτές οι δύο περιπτώσεις είναι:

- Γραμμική περίπτωση SVM: Είναι η περίπτωση όπου τα δεδομένα είναι γραμμικά διαχωρίσιμα. Δηλαδή το διαχωριστικό επίπεδο που επιλέγεται, ταξινομεί τέλεια σε κάθε κατηγορία το κάθε σημείο δεδομένων.
- Μη γραμμική περίπτωση SVM: Πρόκειται για την περίπτωση όπου τα δεδομένα δεν είναι γραμμικά διαχωρίσιμα, δηλαδή δεν υπάρχει μια ευθεία γραμμή στον χώρο που να μπορεί να ταξινομήσει τα σημεία δεδομένων σε δύο κατηγορίες.

Σε περίπτωση που το σύνολο δεδομένων δεν είναι γραμμικά διαχωρίσιμο τότε ο αλγόριθμος των μηχανών διανυσμάτων υποστήριξης χρησιμοποιεί το λεγόμενο «τέχνασμα του πυρήνα» όπου στην ουσία με την βοήθεια των συναρτήσεων πυρήνα (Kernel functions), προσθέτει ακόμα μία διάσταση στον χώρο των δύο διαστάσεων που βρίσκονται τα δεδομένα με σκοπό να βρεθεί η καλύτερη από όλες τις μη γραμμικές υπερεπιφάνειες που να μεγιστοποιεί πάλι, το περιθώριο μεταξύ των δύο κλάσεων. Οι συναρτήσεις πυρήνα που χρησιμοποιούνται και είναι γνωστές ευρέως είναι οι κάτωθι (scikit - kearn):

- Linear Kernel
- Polynomial Kernel
- Sigmoid Kernel
- Radial Basis Function
- Gaussian Kernel
- Anova radial basis Kernel



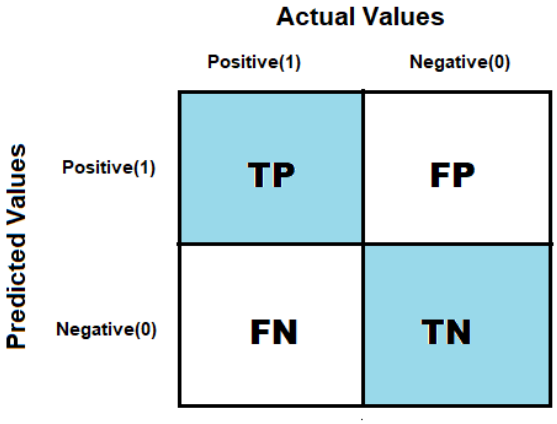
Σχήμα 14: Γραφική αναπαράσταση λειτουργίας αλγορίθμου «Support Vector Machines»
Πηγή σχήματος: (https://datatron.com/what-is-a-support-vector-machine/)

3.4.7 Μέτρα αξιολόγησης μοντέλων κατηγοριοποίησης

Στην διεθνή βιβλιογραφία υπάρχουν αρκετά μέτρα αξιολόγησης που χρησιμοποιούνται για να εξετάσουν την προβλεπτική ικανότητα των διαφόρων μεθόδων κατηγοριοποίησης που εφαρμόζονται (Data Mining and Analysis, 2014). Στην συνέχεια, θα παρουσιαστούν μερικά από τα πιο γνωστά μέτρα εξ ’αυτών.

● **Πίνακας ή μήτρα σύγχυσης (Confusion Matrix):** Ο πίνακας σύγχυσης χρησιμοποιείται για να περιγράψει την απόδοση του μοντέλου κατηγοριοποίησης σε ένα σύνολο δεδομένων δοκιμής

όπου οι πραγματικές τιμές είναι γνωστές. Φυσικά μπορεί να χρησιμοποιηθεί και όταν υπάρχουν παραπάνω από δύο κατηγορίες.



Σχήμα 15: Η μορφή της πίνακα σύγχυσης

Στο σχήμα 15 παρουσιάζεται η πιο απλή μορφή του 2X2 πίνακα σύγχυσης. Στην περίπτωση αυτήν υπάρχουν μόνο δύο κατηγορίες, η κατηγορία κ_1 όπου αναπαριστά την θετική κατηγορία, και υπάρχει και η κατηγορία κ_0 όπου αντιστοιχεί στην αρνητική κατηγορία. Στον οριζόντιο άξονα του πίνακα σύγχυσης είναι οι προβλεπόμενες τιμές, δηλαδή οι προβλεπόμενες κατηγορίες, ενώ στον κατακόρυφο άξονα είναι οι πραγματικές κατηγορίες που ανήκουν οι παρατηρήσεις (τα δεδομένα). Τα εσωτερικά στοιχεία της πίνακα σύγχυσης έχουν της ονομασίες «Αληθώς θετικά (TP)», «Ψευδώς θετικά (FP)», «Ψευδώς αρνητικά (FN)» και «Αληθώς Αρνητικά (TN)».

- ***Αληθώς θετικά (True Positives)*:** Είναι το σύνολο των παρατηρήσεων όπου ο κατηγοριοποιητής προβλέπει πως ανήκουν στην κατηγορία κ_1 και όντως στην πραγματικότητα ανήκουν στην κατηγορία κ_1.

- ***Ψευδώς θετικά (False Positives)*:** Πρόκειται για το σύνολο των παρατηρήσεων που ο αλγόριθμος κατηγοριοποίησης προβλέπει πως ανήκουν στην κατηγορία κ_1 ενώ στην πραγματικότητα ανήκουν στην κατηγορία κ_0.

- ***Ψευδώς αρνητικά (False Negatives)*:** Αντιπροσωπεύει το πλήθος των παρατηρήσεων της οποίες ο κατηγοριοποιητής προβλέπει πως ανήκουν στην κατηγορία κ_0 ενώ στην πραγματικότητα ανήκουν στην κατηγορία κ_1.

- ***Αληθώς αρνητικά (True Negatives)*:** Είναι το πλήθος των παρατηρήσεων της οποίες ο αλγόριθμος κατηγοριοποίησης προβλέπει πως ανήκουν στην κατηγορία κ_0 ενώ όντως στην πραγματικότητα ανήκουν στην κατηγορία κ_0.

● **Ποσοστό σφαλμάτων (Error Rate):** Είναι το ποσοστό των ψευδών προβλέψεων στο σύνολο των δοκιμών και βρίσκεται μέσω του τύπου:

$$\text{Error Rate} = \frac{FP+FN}{n}$$

● **Ακρίβεια (Accuracy):** Είναι το ποσοστό των αληθών προβλέψεων στο σύνολο των δοκιμών και υπολογίζεται μέσω του τύπου:

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN}$$

Όσο υψηλότερη τιμή έχει η ακρίβεια τόσο καλύτερος είναι ο κατηγοριοποιητής.

● **Ανάκληση (Recall):** Είναι το ποσοστό των πραγματικά θετικών περιπτώσεων οι οποίες μπόρεσαν να προβλεφθούν από το μοντέλο που προσαρμόστηκε και ορίζεται ως:

$$\text{Recall} = \frac{TP}{TP+FN}$$

Όσο μεγαλύτερη τιμή έχει η ανάκληση στο διάστημα [0,1] όπου είναι το πεδίο τιμών της, τόσο καλύτερος είναι ο κατηγοριοποιητής.

● **Εξειδικευμένη ακρίβεια (Precision):** Η εξειδικευμένη ακρίβεια για της δύο κατηγορίες κ_1 και κ_2 μπορεί να υπολογιστεί μέσω των σχέσεων:

$$\text{Precision}_{\theta\text{ετικών}} = \frac{TP}{TP+FP}$$

$$\text{Precision}_{Αρνητικών} = \frac{TN}{TN+FN}$$

Η εξειδικευμένη ακρίβεια για την θετική κατηγορία είναι το ποσοστό των αληθώς θετικών προβλέψεων στο σύνολο των θετικών προβλέψεων είτε ψευδώς θετικών είτε αληθώς θετικών.
Αντίστοιχα, η εξειδικευμένη ακρίβεια για την αρνητική κατηγορία είναι το ποσοστό των αληθώς αρνητικών προβλέψεων στο σύνολο των αρνητικών προβλέψεων είτε ψευδώς αρνητικών είτε αληθώς αρνητικών.
Η συνολική εξειδικευμένη ακρίβεια του κατηγοριοποιητή είναι ο σταθμισμένος μέσος όρος των τιμών της εξειδικευμένης ακρίβειας για κάθε κατηγορία. Το πεδίο τιμών της εξειδικευμένης ακρίβειας είναι το διάστημα [0,1]. Όσο μεγαλύτερη τιμή έχει η εξειδικευμένη ακρίβεια, τόσο καλύτερος είναι ο κατηγοριοποιητής.

● **Ποσοστό αληθώς θετικών ή Ευαισθησία:** Το ποσοστό αληθώς θετικών προβλέψεων ή αλλιώς η ευαισθησία είναι στην ουσία η ανάκληση για την θετική κατηγορία κ_1. Υπολογίζεται μέσω του τύπου:

$$\text{TPR} = \text{Recall}_{\theta\text{ετικών}} = \frac{TP}{TP+FN}$$

● **Ποσοστό αληθώς αρνητικών ή Ειδικότητα:** Το ποσοστό αληθώς αρνητικών προβλέψεων ή αλλιώς η ειδικότητα, είναι η ανάκληση για την αρνητική κατηγορία και υπολογίζεται μέσω του τύπου:

$$\text{TNR} = \text{Recall}_{Αρνητικών} = \frac{TN}{TN+FP}$$

● **Ποσοστό ψευδώς θετικών προβλέψεων:** Ορίζεται ως ακολούθως:

$$\text{FPR} = \frac{FP}{FP+TN} = 1 - \text{ειδικότητα}$$

● **Ποσοστό ψευδώς αρνητικών προβλέψεων:** Ορίζεται ως ακολούθως:

$$\text{FNR} = \frac{FN}{TP+FN} = 1 - \text{ευαισθησία}$$

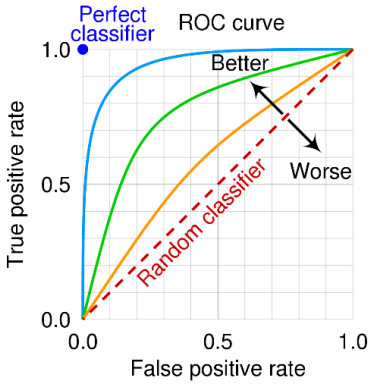
● **F-μέτρο (F-Score):** Το μέτρο F, ουσιαστικά προσπαθεί να φέρει σε ισορροπία της τιμές εξειδικευμένης ακρίβειας (Precision) και ανάκλησης (Recall) υπολογίζοντας τον αρμονικό της μέσο για κάθε κατηγορία κ_i. Υπολογίζεται μέσω της σχέσης:

$$F_i = \frac{2 \cdot precision_i \cdot recall_i}{precision_i + recall_i}$$

Το συνολικό F-μέτρο είναι ο μέσος όρος των τιμών του F-μέτρου για κάθε κατηγορία κ_i. Το μέτρο F παίρνει τιμές στο διάστημα [0,1] και όσο μεγαλύτερη τιμή έχει, τόσο καλύτερος είναι ο κατηγοριοποιητής.

● **Καμπύλή ROC:** Η καμπύλη ROC «Receiver Operating Characteristic» πρόκειται για την γραφική παράσταση της ευαισθησίας (TPR) έναντι της ποσότητας (1-ειδικότητα) των οποίων οι τιμές μεταβάλλονται για κάθε τιμή της διαχωριστικής τιμής c. Η διαχωριστική τιμή c είναι το σημείο αποκοπής ή με άλλα λόγια μια τιμή που πάνω ή κάτω από αυτήν εμφανίζεται ένα γεγονός που μελετάται. Για να θεωρηθεί ένα μοντέλο κατηγοριοποίησης ότι έχει καλή απόδοση θα πρέπει, η καμπύλη που σχηματίζεται να είναι όσο πιο κοντά γίνεται στην πάνω αριστερή γωνία του γραφήματος. Η περιοχή που σχηματίζεται κάτω από την καμπύλη «Area Under Curve (AUC)» είναι ένα μέτρο ακρίβειας που δείχνει το γενικό ποσοστό των σωστά κατηγοριοποιημένων παρατηρήσεων. Τιμές κοντά στην μονάδα δείχνουν πολύ καλή ακρίβεια του μοντέλου κατηγοριοποίησης που προσαρμόστηκε. Για να βρεθεί το βέλτιστο σημείο χρησιμοποιείται το κριτήριο του Youden ευρέως γνωστό και ως «Δείκτης Youden»:

$$Youden\ J = sensitivity + specificity - 1$$



Σχήμα 16: Γραφική απεικόνιση κάποιων καμπυλών ROC
Πηγή σχήματος: (Wikipedia, «Receiver operating characteristic»)

3.5 Η τεχνική της συσταδοποίησης «Clustering»

Η μέθοδος της συσταδοποίησης πρόκειται για ένα είδος μη επιτηρούμενης μηχανικής μάθησης η οποία εξετάζει ένα σύνολο δεδομένων για την ομοιότητα που παρουσιάζουν σε σχέση με ένα πλήθος χαρακτηριστικών και σκοπός της μεθόδου είναι να δημιουργήσει τις λεγόμενες «συστάδες» ή εναλλακτικά ομάδες, από τις παρατηρήσεις που είναι περισσότερο όμοιες μεταξύ τους. Για να θεωρηθεί επιτυχημένη η εφαρμογή της μεθόδου της συσταδοποίησης, θα πρέπει οι παρατηρήσεις μέσα σε κάθε συστάδα να είναι όσο το δυνατόν πιο ομοιογενείς γίνεται και οι παρατηρήσεις που ανήκουν σε διαφορετικές συστάδες να διαφέρουν όσο γίνεται πιο πολύ (Εφαρμοσμένη στατιστική και στατιστική μηχανική μάθηση, 2021). Οι κύριες κατηγορίες των μεθόδων συσταδοποίησης είναι δύο:

1. Ιεραρχικές μέθοδοι:
    - Διαιρετικές μέθοδοι (Divisive)
    - Συσσωρευτικές μέθοδοι (Agglomerative)

2. Μη ιεραρχικές μέθοδοι

3.5.1 Μη ιεραρχικές μέθοδοι – Αλγόριθμος K-Means

Ξεκινώντας με την μη ιεραρχική μέθοδο, σκοπός της είναι να κατατάξει τις n παρατηρήσεις του συνόλου δεδομένων σε k συστάδες (K means algorithm, 2014). Ο περιορισμός της μη ιεραρχικής μεθόδου είναι ότι ο αριθμός k των συστάδων που πρόκειται να δημιουργηθούν πρέπει να είναι καθορισμένος από την αρχή της υλοποίησής της. Ο τρόπος λειτουργίας της μη ιεραρχικής μεθόδου είναι σχετικά απλός, λειτουργεί επαναληπτικά και χρησιμοποιεί την έννοια του κέντρου βάρους των συστάδων. Αρχικά, δημιουργούνται στο σύνολο δεδομένων, k τυχαία σημεία και γύρω από αυτά κατατάσσονται οι παρατηρήσεις ανάλογα με την απόστασή τους από τα κέντρα των ήδη δημιουργημένων συστάδων, μέχρι να δημιουργηθεί ο επιθυμητός και εκ των προτέρων καθορισμένος αριθμός των συστάδων. Για τον υπολογισμό των αποστάσεων των παρατηρήσεων από τα κέντρα των συστάδων χρησιμοποιείται κυρίως η «Ευκλείδεια απόσταση».

Ωστόσο και άλλες αποστάσεις όπως για παράδειγμα οι αποστάσεις «Mahalanobis» και «Manhattan» χρησιμοποιούνται συχνά. Ο πιο γνωστός αλγόριθμος αυτής της μεθόδου είναι ο αλγόριθμος «K-Means».

Ο αλγόριθμος K-Means πρόκειται για έναν μη ιεραρχικό αλγόριθμο διαμέρισης (partitioning algorithm) συσταδοποίησης και έχει σαν περιορισμό ότι το πλήθος των k επιθυμητών συστάδων πρέπει να γνωστό από την αρχή. Λειτουργεί πολύ καλά για τεράστια σύνολα δεδομένων καθώς ο χρόνος εκτέλεσής του είναι μικρότερος συγκριτικά με την ιεραρχική συσταδοποίηση. Ο αλγόριθμος K-Means σαν πρώτο βήμα, υπολογίζει τα αρχικά κέντρα των συστάδων, δηλαδή με άλλα λόγια καθορίζει ένα σύνολο από k μητρικά σημεία στον χώρο δεδομένων. Στην συνέχεια, υπολογίζει τις αποστάσεις της κάθε μιας παρατήρησης από τα κέντρα βαρών των δημιουργημένων συστάδων έστω μ_i, και ταξινομεί κάθε παρατήρηση στην συστάδα όπου το κέντρο της έχει την μικρότερη απόσταση από την παρατήρηση.

$$\mu_i = \frac{1}{n_i} \sum_{x_j \, \epsilon \, C_i} x_j,$$

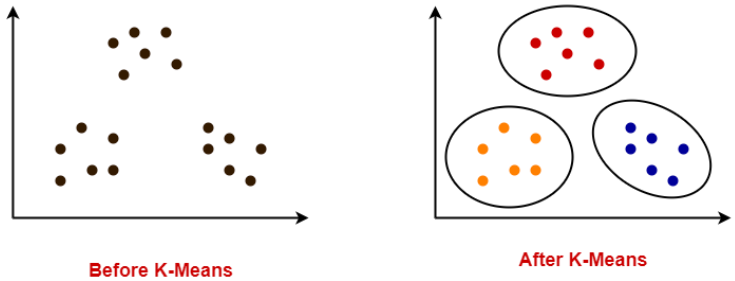
όπου n_i είναι το πλήθος των παρατηρήσεων x_j (j = 1,2,…n) που ανήκουν στην συστάδα C_i (i = 1,2,..,k).

Ύστερα, αφού έχουν ταξινομηθεί οι παρατηρήσεις σε συστάδες, ο αλγόριθμος υπολογίζει τα νέα κέντρα των ομάδων με βάση τις παρατηρήσεις που ταξινομήθηκαν σε κάθε ομάδα. Εφόσον τα νέα κέντρα που δημιουργήθηκαν δεν έχουν διαφορές με τα παλιά, τότε ο αλγόριθμος σταματά αλλιώς υπολογίζει ξανά την αποστάσεις των παρατηρήσεων από τα κέντρα των ομάδων μέχρι να μην υπάρχουν διαφορές στα κέντρα των νέων συστάδων από τα παλιά κέντρα. Επειδή η όλη διαδικασία εξαρτάται από τα αρχικά κέντρα των ομάδων, για να βρεθεί η βέλτιστη συσταδοποίηση ο αλγόριθμος K-Means χρησιμοποιεί μια επαναληπτική τεχνική με στόχο να βρει εκείνη την συσταδοποίηση, έστω C, η οποία ελαχιστοποιεί το άθροισμα τετραγώνων των σφαλμάτων (SSE). Δηλαδή από την σχέση:

$$\text{SSE(C)} = \sum_{i=1}^{k} \sum_{x_j \, \epsilon \, C_i} || \; x_j \; - \; \mu_i \; ||^{\,2},$$

ο αλγόριθμος K-Means ψάχνει να βρει την συσταδοποίηση που ελαχιστοποιεί το SSE, δηλαδή:

$$C^* = \arg min_C \; \{\text{SSE(C)}\}$$



Σχήμα 17: Γραφική απεικόνιση λειτουργίας αλγορίθμου K-Means
Πηγή σχημάτων: (https://www.gatevidyalay.com/k-means-clustering-algorithm-example/)

3.5.2 Ιεραρχική συσταδοποίηση

Η ιεραρχική συσταδοποίηση μπορεί όπως αναφέρθηκε και προηγουμένως να διακριθεί σε δύο κύριες μεθόδους. Στις συσσωρευτικές μεθόδους και στις διαιρετικές μεθόδους. Οι συσσωρευτικές μέθοδοι αρχίζουν με n το πλήθος συστάδες και κάνοντας συνεχώς συγχωνεύσεις τελικά δημιουργούν μια συστάδα η οποία περιέχει όλες τις παρατηρήσεις του συνόλου δεδομένων. Από την άλλη πλευρά, οι διαιρετικοί αλγόριθμοι εκτελούν την αντίθετη διαδικασία από αυτή των συσσωρευτικών αλγορίθμων. Δηλαδή αρχίζουν με μια συστάδα η οποία περιέχει n το πλήθος παρατηρήσεις και στην συνέχεια, διαιρούν τα δεδομένα σε συστάδες μικρότερου μεγέθους, μέχρι να δημιουργηθούν συστάδες που να περιέχουν μία παρατήρηση η κάθε συστάδα. Το πρόβλημα των ιεραρχικών μεθόδων συσταδοποίησης είναι ότι είναι ασύμφορες για πολύ μεγάλα σύνολα δεδομένων γιατί χρησιμοποιούν σε κάθε επανάληψή τους έναν «πίνακα αποστάσεων» ο οποίος στην πραγματικότητα αντιπροσωπεύει τις αποστάσεις όλων των παρατηρήσεων από τις άλλες και αυτό έχει σαν αποτέλεσμα να απαιτείται αρκετός χώρος στην μνήμη του υπολογιστή και φυσικά χρόνος ώστε να υλοποιηθούν όλες οι επαναλήψεις και ο υπολογισμός των αποστάσεων σε κάθε επανάληψη. Υπάρχουν αρκετές μέθοδοι για τον υπολογισμό των αποστάσεων των συστάδων που δημιουργήθηκαν έτσι ώστε να γίνουν οι μελλοντικές συγχωνεύσεις. Μερικές από τις πιο γνωστές μεθόδους είναι οι ακόλουθες:

- Complete Linkage Method ή Furthest neighbor method: Η μέθοδος αυτή υπολογίζει την απόσταση μεταξύ δύο συστάδων ως την μεγαλύτερη απόσταση από μια παρατήρηση η οποία βρίσκεται στην μία συστάδα με μια άλλη παρατήρηση που βρίσκεται σε μια άλλη συστάδα. Δημιουργούνται μεγάλες και συμπαγείς συστάδες όμως κάποιες φορές δεν καταφέρνει η μέθοδος να φτιάξει μικρές, συμπαγείς συστάδες.

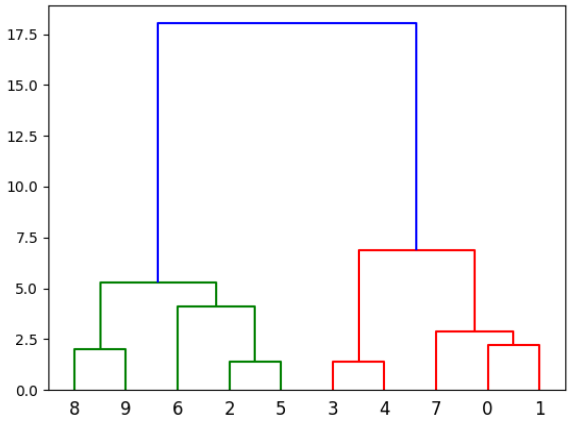
- Single linkage method ή Nearest neighbor method: Σε αντίθεση με την μέθοδο «Furthest neighbor», η μέθοδος «Nearest neighbor» υπολογίζει την απόσταση μεταξύ δύο συστάδων ως την μικρότερη απόσταση από μια παρατήρηση που βρίσκεται στην μια συστάδα με μια άλλη παρατήρηση που βρίσκεται σε μια άλλη συστάδα. Το μειονέκτημα της μεθόδου είναι ότι δημιουργεί μη συμπαγείς συστάδες, άλλες πολύ μεγάλες και άλλες πολύ μικρές.

- Weighted average linkage method: Εδώ η απόσταση μεταξύ των συστάδων ορίζεται ως ο μέσος των αποστάσεων των παρατηρήσεων της μια συστάδας με τις παρατηρήσεις της άλλης συστάδας.

- Centroid method: Υπολογίζεται ως η απόσταση των κέντρων των συστάδων. Παρότι δημιουργεί συμπαγείς συστάδες, το βασικό μειονέκτημα της μεθόδου αυτής είναι ότι μπορεί να χρησιμοποιηθεί μόνο για ποσοτικά δεδομένα διότι κάνει χρήση της ευκλείδειας απόστασης.

- Ward method: Χρησιμοποιείται αρκετά συχνά στην πράξη διότι ελαχιστοποιεί την διακύμανση μέσα στις συστάδες και μπορεί να δημιουργεί συστάδες με παρόμοιο αριθμό παρατηρήσεων.

Τα αποτελέσματα που παράγει η ιεραρχική συσταδοποίηση σχετικά με τον αριθμό των συστάδων που δημιουργούνται και τον τρόπο που συγχωνεύονται οι παρατηρήσεις, μπορούν να απεικονιστούν μέσω ενός γραφήματος που ονομάζεται «δενδρόγραμμα».



Σχήμα 18: Δενδρόγραμμα που προκύπτει από εφαρμογή ιεραρχικής συσταδοποίησης

3.5.3 Μέτρα αξιολόγησης συσταδοποίησης

Στην διεθνή βιβλιογραφία υπάρχουν αρκετά μέτρα αξιολόγησης της τεχνικής συσταδοποίησης. Στα πλαίσια της παρούσας διπλωματικής εργασίας θα αναφερθούν τα πιο γνωστά από αυτά τα μέτρα.

- **Silhouette Coefficient:** Ο συντελεστής Silhouette παίρνει τιμές από -1 έως 1. Τιμές των παρατηρήσεων κοντά στο 1 δείχνουν ότι η παρατήρηση βρίσκεται κοντά σε σημεία της δικής της συστάδας και μακριά από υπόλοιπες συστάδες. Τιμές κοντά στο 0 δείχνουν ότι η παρατήρηση x_i βρίσκεται κοντά στο όριο μεταξύ δύο συστάδων. Μια τιμή κοντά στο -1 δείχνει ότι η παρατήρηση βρίσκεται σε κάποια άλλη συστάδα από ότι στην δική του συστάδα, άρα πολύ πιθανόν να έχει ταξινομηθεί σε λάθος συστάδα. Σύμφωνα και με το βιβλίο (Data Mining and Analysis: Fundamentals Concepts and Algorithms, Mohammed J. Zaki et al (2014), p. 473) (2014), για κάθε σημείο x_i, ο τύπος υπολογισμού του συντελεστή Silhouette είναι ο ακόλουθος:

$$s_i = \frac{\mu_{out}^{min}(x_i) - \mu_{in}(x_i)}{max\{\mu_{out}^{min}(x_i),\ \mu_{in}(x_i)\}},$$

όπου με $\mu_{in}\ (x_i)$ συμβολίζεται η μέση απόσταση του x_i από σημεία της συστάδας που ανήκει, ενώ με $\mu_{out}^{min}\ (x_i)$ συμβολίζεται ο μέσος των αποστάσεων του x_i από τα σημεία της πλησιέστερης συστάδας.

Ο τελικός συντελεστής Silhouette ορίζεται ως η μέση τιμή s_i για όλα τα σημεία:

$$SC = \frac{\sum_{i=1}^{n} s_i}{n}$$

- **Δείκτης Davies – Bouldin:** Η μικρότερη τιμή που μπορεί να πάρει αυτός ο δείκτης είναι 0. Όσο μικρότερη είναι η τιμή του δείκτη Davies – Bouldin, τόσο καλύτερη είναι η συσταδοποίηση επειδή αυτό σημαίνει ότι η απόσταση μεταξύ των μέσων των συστάδων είναι μεγάλη και κάθε συστάδα έχει μικρή διασπορά.

Για ένα ζεύγος συστάδων C_i και C_j ο δείκτης Davies-Bouldin δίνεται από την σχέση:

$$DB = \frac{1}{K} \sum_{i=1}^{k} max_{i \neq j}\{DB_{ij}\},$$

όπου $DB_{ij} = (\sigma_{\mu_i} + \sigma_{\mu_j}) / \mathrm{d}(\mu_i, \mu_j)$

Με σ_{μ_i} συμβολίζεται η συνολική διακύμανση της συστάδας C_i και με $\mathrm{d}(\mu_i, \mu_j)$ συμβολίζεται η ευκλείδεια απόσταση μεταξύ των μέσων των συστάδων C_i και C_j.

- **Δείκτης Dunn:** Ορίζεται ως ο λόγος της ελάχιστης απόστασης μεταξύ παρατηρήσεων που ανήκουν σε διαφορετικές συστάδες προς την μέγιστη απόσταση μεταξύ παρατηρήσεων όπου ανήκουν στην ίδια συστάδα. Σύμφωνα και με το βιβλίο (Data Mining and Analysis: Fundamentals Concepts and Algorithms, Mohammed J. Zaki et al (2014), p. 472-473) (2014), «όσο μεγαλύτερη τιμή λαμβάνει ο δείκτης Dunn, τόσο πιο καλή είναι η συσταδοποίηση γιατί αυτό σημαίνει ότι ακόμα και η πλησιέστερη απόσταση μεταξύ σημείων που ανήκουν σε διαφορετικές συστάδες είναι πολύ μεγαλύτερη από την απώτερη απόσταση μεταξύ των σημείων που ανήκουν στην ίδια συστάδα». Η σχέση από την οποία δίνεται είναι η κάτωθι:

$$Dunn = W_{out}^{min} / W_{in}^{max},$$

όπου με W_{out}^{min} συμβολίζεται η ελάχιστη διασυσταδική απόσταση ενώ με W_{in}^{max} συμβολίζεται η μέγιστη ενδοσυσταδική απόσταση

- **Δείκτης Calinski - Harabasz:** Αλλιώς ονομάζεται και κριτήριο του λόγου διακύμανσης. Όσο μεγαλύτερες τιμές παίρνει ο συγκεκριμένος δείκτης τόσο καλύτερα έχει πραγματοποιηθεί η συσταδοποίηση αφού υψηλές τιμές σημαίνουν ότι οι συστάδες είναι σωστά διαχωρισμένες. Δείχνει ότι η ενδοσυσταδική διασπορά είναι μικρότερη από την διασυσταδική. Ισούται με το πηλίκο του αθροίσματος της διασποράς μεταξύ των συστάδων διά του αθροίσματος της διασποράς εντός των συστάδων. Για ένα σύνολο δεδομένων D = $\{d_1, d_2, d_3, \dots, d_n\}$ ο δείκτης Calinski-Harabasz δίνεται από την σχέση:

$$CH = \left[\frac{n-K}{K-1}\right] [\ \sum_{k=1}^{K} n_k ||c_k - \mu||^2 / \sum_{k=1}^{K} \sum_{i=1}^{n_k} ||d_i - c_k||^2\ ],$$

όπου με n_k και με c_k συμβολίζονται ο αριθμός των σημείων και το κέντρο της συστάδας k αντίστοιχα. Με μ συμβολίζεται ο μέσος όρος του συνόλου δεδομένων, δηλαδή

$$\mu = \frac{\sum_{j=1}^{n} d_j}{n}$$

- **Δείκτης Fowlkes - Mallows:** Σκοπός του συγκεκριμένου μέτρου αξιολόγησης είναι να ποσοτικοποιήσει την ομοιότητα μεταξύ συσταδοποιήσεων που προκύπτουν από διαφορετικούς αλγορίθμους συσταδοποίησης. Χρησιμοποιείται συνήθως για την αξιολόγηση της απόδοσης συσταδοποίησης ενός συγκεκριμένου αλγορίθμου υποθέτοντας ότι η συστάδα με την οποία συγκρίνεται είναι η βασική αλήθεια, δηλαδή με άλλα λόγια η τέλεια συστάδα. Λαμβάνει τιμές στο διάστημα [0,1] όπου τιμές κοντά στο 1 υποδηλώνουν εξαιρετική συσταδοποίηση.

- **Άθροισμα τετραγώνων των σφαλμάτων (SSE):** Δείχνει το σφάλμα της διαφοράς μεταξύ παρατηρούμενων και προβλεπόμενων τιμών. Όσο μικρότερη τιμή έχει το SSE, τόσο πιο αξιόπιστη είναι η συσταδοποίηση.

Φυσικά όπως αναφέρθηκε και στην αρχή της συγκεκριμένης υπό ενότητας, στην διεθνή βιβλιογραφία υπάρχουν και άλλα αρκετά μέτρα για την αξιολόγηση της συσταδοποίησης όπως το στατιστικό Rand, η καθαρότητα (Purity), η εντροπία υπό συνθήκη, το μέτρο των αμοιβαίων πληροφοριών (Mutual Information), το στατιστικό Hubert, ο συντελεστής Jaccard και πολλά άλλα.

3.6 Η μέθοδος ανάλυσης κύριων συνιστωσών (Principal Component Analysis «PCA»)

Μια ακόμη μέθοδος η οποία πρόκειται για μέθοδο μη επιτηρούμενης μάθησης είναι αυτή της ανάλυσης κυρίων συνιστωσών (Data Mining and Analysis, 2014) (Applied Multivariate Statistical Analysis, 2007) (built in, 2023). Η τεχνική της ανάλυσης σε κύριες συνιστώσες πρόκειται για έναν ορθογώνιο μετασχηματισμό του χώρου του προβλήματος και έχει σαν σκοπό

την μείωση των διαστάσεων του προβλήματος που πρέπει να επιλυθεί διατηρώντας όσο περισσότερη πληροφορία γίνεται. Δηλαδή αποσκοπεί στο να αντικαταστήσει ένα σύνολο μεταβλητών X_1 , X_2 , X_3 ,…, X_p με ένα μικρότερο σύνολο ασυσχέτιστων μεταβλητών τις συνιστώσες που προκύπτουν από γραμμικούς συνδυασμούς των αρχικών μεταβλητών. Αυτοί οι γραμμικοί συνδυασμοί των αρχικών μεταβλητών που προκύπτουν, θα είναι ασυσχέτιστοι μεταξύ τους και θα περιέχουν ένα μεγάλο ποσοστό της πληροφορίας εξηγείται από τις αρχικές μεταβλητές. Με αυτόν τον τρόπο η τεχνική «PCA» επιτυγχάνει να δημιουργήσει ένα σύνολο λιγότερων και ασυσχέτιστων μεταβλητών οι οποίες εξηγούν ένα μεγάλο ποσοστό της συνολικής μεταβλητότητας των αρχικών δεδομένων. Για να βρεθούν οι κύριες συνιστώσες απαιτούνται να γίνουν φυσικά κάποια βήματα. Πρώτο βήμα είναι να πραγματοποιηθεί τυποποίηση των μεταβλητών X_1, X_2,…, X_pέτσι ώστε η ανάλυση να μην επηρεάζεται από τις μονάδες μέτρησης. Αυτό γίνεται ως εξής:

$$Z_i = \frac{x_i - \bar{x}}{s},$$

όπου $\bar{x}$ είναι ο δειγματικός μέσος κάθε μεταβλητής X_i και με s συμβολίζεται η δειγματική τυπική απόκλιση της κάθε μεταβλητής.
Με αυτόν τον τρόπο οι p μεταβλητές θα έχουν μέση τιμή ίση με το 0 και διακύμανση ίση με την μονάδα. Στην συνέχεια δημιουργούνται οι i το πλήθος συνιστώσες που είναι ίσες με τον αριθμό των αρχικών μεταβλητών. μέσω των γραμμικών συνδυασμών των αρχικών μεταβλητών, έστω Z_i όπου:

$$Z_i = \sum_{i=1}^{p} \sum_{j=1}^{p} \alpha_{ij} X_i,$$

όπου α_{ij} είναι οι συντελεστές στάθμισης ή εναλλακτικά οι συντελεστές με τα βάρη των αρχικών μεταβλητών. Επειδή δίνοντας μεγαλύτερη τιμή στα βάρη α_{ij} γίνεται μεγαλύτερη και η διακύμανση της συνιστώσας Z_i, για να αποφευχθεί αυτό το πρόβλημα τα βάρη υπολογίζονται με τον περιορισμό ότι το άθροισμα τετραγώνων τους είναι ίσο με 1, δηλαδή

$$\sum_{j=1}^{p} {\alpha_{ij}}^2 = 1$$

Επόμενο ερώτημα που τίθεται είναι ποιος πίνακας θα χρησιμοποιηθεί για την εύρεση των βαρών α_{ij} και φυσικά για την εύρεση των κύριων συνιστωσών. Για τον υπολογισμό των βαρών α_{ij} υπολογίζεται ο πίνακας διακυμάνσεων συνδιακυμάνσεων των αρχικών μεταβλητών, έστω C, ο οποίος περιέχει τις διακυμάνσεις των μεταβλητών και τις συνδιακυμάνσεις μεταξύ των μεταβλητών. Η μορφή του είναι η ακόλουθη:

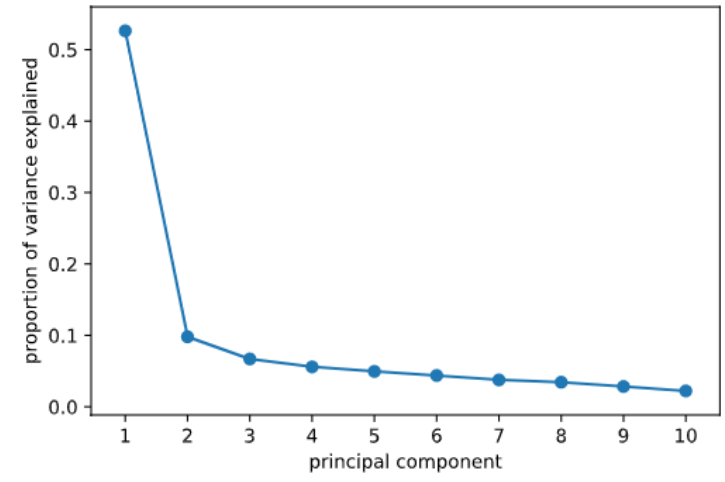
$$C = \begin{pmatrix} Cov(X_1, X_1) & \cdots & Cov(X_1, X_p) \\ \vdots & \ddots & \vdots \\ Cov(X_p, X_1) & \cdots & Cov(X_p, X_p) \end{pmatrix},$$

όπου τα διαγώνια στοιχεία του πίνακα διακυμάνσεων συνδιακυμάνσεων είναι στην πραγματικότητα οι διακυμάνσεις των μεταβλητών αφού ισχύει ότι $\mathrm{Cov}(X_i, X_i) = \mathrm{Var}(X_i)$. Εφόσον έχει προηγηθεί βέβαια τυποποίηση των μεταβλητών, τότε προκύπτει ο δειγματικός πίνακας συσχετίσεων για τα διαθέσιμα δεδομένα ο οποίος έχει μορφή:

$$\text{Corr} = \begin{pmatrix} 1 & \cdots & Corr(X_1, X_p) \\ \vdots & \ddots & \vdots \\ Corr(X_p, X_1) & \cdots & 1 \end{pmatrix},$$

δηλαδή, τα διαγώνια στοιχεία του πίνακα συσχετίσεων είναι ίσα με την μονάδα και τα υπόλοιπα στοιχεία του πίνακα αντιπροσωπεύουν την συσχέτιση μεταξύ των μεταβλητών X_i και X_j.

Στην συνέχεια, υπολογίζονται οι ιδιοτιμές, δηλαδή τα διαγώνια στοιχεία του πίνακα διακυμάνσεων συνδιακυμάνσεων και τα ιδιοδιανύσματα του πίνακα διακυμάνσεων συνδιακυμάνσεων που είναι στην ουσία αυτά τα βάρη α_{ij} («loadings»). Οι ιδιοτιμές αντιστοιχούν στις διακυμάνσεις των κύριων συνιστωσών και αντιπροσωπεύουν την μεταβλητότητα που εξηγεί η κάθε κύρια συνιστώσα. Υπάρχουν κάποιοι τρόποι με τους οποίους επιλέγονται οι σημαντικότερες κύριες συνιστώσες. Ένα κριτήριο είναι να διατηρούνται οι πρώτες κύριες συνιστώσες οι οποίες μαζί να εξηγούν τουλάχιστον ένα κατώτατο όριο ποσοστού της ολικής μεταβλητότητας των αρχικών δεδομένων, για παράδειγμα, τουλάχιστον το 75%. Ένα άλλο κριτήριο επιλογής κυρίων συνιστωσών είναι το κριτήριο του «Kaiser» σύμφωνα με το οποίο επιλέγονται όλες οι κύριες συνιστώσες που έχουν ιδιοτιμή μεγαλύτερη της μονάδας. Ένας ακόμα τρόπος επιλογής των κυρίων συνιστωσών είναι με την βοήθεια του γραφήματος «Scree plot» ο οποίος συχνά αναφέρεται ως «κανόνας του αγκώνα (Rule of Elbow)» σύμφωνα με τον οποίο, στο σημείο που η καμπύλη του γραφήματος ισοπεδωθεί, επιλέγονται ο αριθμός των κυρίων συνιστωσών πριν το σημείο που αρχίζει να εμφανίζεται αυτή ισοπέδωση της καμπύλης. Ο κάθετος άξονας του γραφήματος δείχνει είτε την τιμή της ιδιοτιμής, είτε το ποσοστό της διακύμανσης που εξηγεί η κάθε κύρια συνιστώσα, ενώ ο οριζόντιος άξονας του γραφήματος δείχνει τον αριθμό των κυρίων συνιστωσών. Παρατηρώντας το παρακάτω σχήμα 19, φαίνεται πως οι τρεις πρώτες κύριες συνιστώσες εξηγούν μαζί, συνολικά το 75% περίπου της συνολικής μεταβλητότητας των δεδομένων. Δεδομένου ότι το ελάχιστο κατώτατο όριο που έχει τεθεί για να εξηγούν οι κύριες συνιστώσες είναι το 75%, τότε θα επιλέγονταν στην προκειμένη περίπτωση, οι τρεις πρώτες κύριες συνιστώσες.



Σχήμα 19: Scree plot της μεθόδου PCA

ΚΕΦΑΛΑΙΟ 4^{Ο}

4. Εφαρμογές μηχανικής μάθησης στην ναυτιλία μέσα από την διεθνή βιβλιογραφία

4. Η ανάγκη χρήσης τεχνικών μηχανικής μάθησης στην ναυτιλία

Οι νέες προκλήσεις της ναυτιλιακής αγοράς εξαιτίας του αυξημένου κόστους κατανάλωσης καυσίμων, της ανάγκης για ακόμη πιο ασφαλή και γρήγορη μεταφορά των φορτίων σε συνδυασμό με την εκπληκτική αύξηση του μεγέθους των διαθέσιμων δεδομένων που ρέουν προς τις ναυτιλιακές εταιρίες έχουν διαμορφώσει μια επιτακτική ανάγκη για εύρεση νέων, καινοτόμων, οικονομικών τρόπων ταξιδιού μέσω θαλάσσης και γενικά την αλλαγή της στρατηγικής πολλών ναυτιλιακών επιχειρήσεων όσον αφορά τον τρόπο με τον οποίο υλοποιούν τις μεταφορές των φορτίων στα διάφορα μέρη του πλανήτη. Τα τελευταία χρόνια, όπως σε σχεδόν όλες τις βιομηχανίες, έτσι και η ναυτιλία έχει επηρεαστεί από τον τεράστιο όγκο δεδομένων και πληροφοριών που είναι διαθέσιμα και τα οποία απαιτούν ειδική μεταχείριση και ανάλυση ώστε να αποκαλυφθεί η «κρυφή» γνώση μέσα τους. Για να αντιμετωπίσουν αυτό το φαινόμενο των «Big data» και για να επεξεργαστούν προς όφελός τους οι ναυτιλιακές επιχειρήσεις την πληροφορία των δεδομένων που διαθέτουν, γίνεται εφαρμογή της επιστήμης της πληροφορικής, των μαθηματικών και της στατιστικής ανάλυσης, και πιο ειδικά ενός συγκεκριμένου τομέα που αναφέρθηκε στο προηγούμενο κεφάλαιο, αυτού της μηχανικής μάθησης. Σε αυτό το κεφάλαιο και στις επόμενες υπό ενότητες πιο συγκεκριμένα, θα παρουσιαστούν μερικά προβλήματα της διεθνής βιβλιογραφίας όπου γίνεται χρήση των αλγορίθμων μηχανικής μάθησης στην ναυτιλία. Γενικά, κοιτώντας την διεθνή βιβλιογραφία, διαπιστώνεται πως υπάρχουν αρκετές μελέτες που χρησιμοποιούν αρκετές και διαφορετικές τεχνικές μηχανικής μάθησης για να ερευνήσουν κάποιο πρόβλημα του συγκεκριμένου κλάδου. Μελέτες όπως αυτές των *Zhihui Hu et al* (2019) με τίτλο «Prediction of Fuel Consumption for Enroute Ship Based on Machine Learning», των *Mohammad Hossein Moradi et al* (2022) με τίτλο «Marine route optimization using reinforcement learning approach to reduce fuel consumption and consequently minimize CO_2 emissions» και των *Tayfun Uyanik et al* (2020) με τίτλο «Machine learning approach to ship fuel consumption: A case of container vessel» χρησιμοποιούν τεχνικές μηχανικής μάθησης για την πρόβλεψη της κατανάλωσης καυσίμων των πλοίων. Άλλες μελέτες όπως των *Andrew Rawson et al* (2021) με τίτλο «A machine learning approach for monitoring ship safety in extreme weather events» αποσκοπούν στην πρόβλεψη πιθανών ατυχημάτων λόγω έντονων καιρικών φαινομένων χρησιμοποιώντας τεχνικές μηχανικής

μάθησης. Φυσικά έχουν γίνει μελέτες που σαν στόχο είχαν την πρόβλεψη των καθυστερήσεων των πλοίων στον προορισμό τους όπως η μελέτη των *Adrian Viellechner et al* (2020) με τίτλο «Novel Data Analytics Meets Conventional Container Shipping: Predicting Delays by Comparing Various Machine Learning Algorithms». Όπως γίνεται αντιληπτό, τα θέματα τα οποία ερευνώνται στον κλάδο της ναυτιλίας με την βοήθεια της μηχανικής μάθησης ποικίλουν και διαφέρουν. Οι περισσότερες μελέτες φαίνεται ωστόσο να ασχολούνται με το μείζον θέμα της κατανάλωσης καυσίμων των πλοίων με απώτερο σκοπό την ελαχιστοποίηση του κόστους των ταξιδιών λόγω της αυξημένης τιμής των καυσίμων, αλλά και την ελαχιστοποίηση της εκπομπής ρύπων προς το περιβάλλον από τα πλοία. Ενδιαφέρον παρουσιάζει ο τρόπος με τον οποίο συλλέγονται τα δεδομένα για την μελέτη προβλημάτων στην βιομηχανία της ναυτιλίας. Σε πολλές έρευνες όπως αναφέρεται και στην συνέχεια, τα δεδομένα συλλέγονται μέσω του «AIS (Automatic Identification System) δηλαδή του αυτόματου συστήματος αναγνώρισης της θέση των πλοίων σε πραγματικό χρόνο. Συνήθως τα δεδομένα πέρα από το «AIS» προέρχονται από αναφορές καιρού και από ιστορικά αρχεία. Επιπλέον, άλλες πηγές δεδομένων στην ναυτιλία είναι τα δεδομένα φορτίων και υλικών, τα δεδομένα μηχανισμού και του σχεδιασμού του πλοίου καθώς είναι διαθέσιμα και δεδομένα παρακολούθησης περιβαλλοντικών συνθηκών και συνθηκών που επικρατούν στο πλοίο. Όλες αυτές οι πηγές δεδομένων είναι χρήσιμες διότι αναλύοντας μέσω της στατιστικής μηχανικής μάθησης αυτά τα δεδομένα οι ναυτιλιακές, θα είναι σε θέση να διαχειρίζονται κατάλληλα τον εξοπλισμό του σκάφους και να γνωρίζουν άμεσα αν υπάρχει κίνδυνος ζημιάς έτσι ώστε να μην υπάρχουν τυχόν ατυχήματα. Επίσης, οι πηγές αυτές συνεισφέρουν στο να διασφαλιστεί, πέρα από την ασφάλεια του πλοίου, και η καλή του ενεργειακή απόδοση καταναλώνοντας λιγότερα καύσιμα βρίσκοντας πιο ασφαλείς και σύντομες διαδρομές, μειώνοντας έτσι συνεπώς και την εκπομπή ρύπων. Τα δεδομένα των πλοίων όπως η ταχύτητα με την οποία κινούνται καθώς και άλλα φυσικά μεγέθη και χαρακτηριστικά ή και ακόμα η απόδοση διαφόρων μηχανημάτων επάνω σε αυτά όπως οι μηχανές τους, οι αντλίες και οι λέβητες, καταγράφονται μέσω αισθητήρων που έχουν τοποθετηθεί πάνω στα πλοία. Πλέον υπάρχουν αρκετά είδη αισθητήρων πάνω στα πλοία κάθε ένας από τους οποίους έχει και έναν διαφορετικό σκοπό, όπως την μέτρηση ταχύτητας, την ασφαλή πλοήγηση του πλοίου, την απόδοση του μηχανισμού του ή και την μέτρηση του ανέμου (Marine Digital). Βέβαια, ίσως οι σημαντικότεροι αισθητήρες στα πλοία είναι τα σόναρ (SNAR). Ουσιαστικά τα σόναρ πρόκειται για συσκευές ηλεκτροακουστικές που χρησιμοποιούν και διαδίδουν τα κύματα ηχητικής ενέργειας μέσα στην θάλασσα (2019). Σκοπός τους γενικά είναι, η ακουστική χαρτογράφηση του βυθού, η πλοήγηση πλοίων, οι υποθαλάσσιες επικοινωνίες και τηλεμετρία καθώς και η αναγνώριση και παρακολούθηση διαφόρων άλλων πλοίων ή υποθαλάσσιων αντικειμένων όπως υποβρύχια.

4.1 1η Μελέτη: Τεχνικές μηχανικής μάθησης για την πρόβλεψη ταχύτητας ενεργειακά αποδοτικών πλοίων

Ένα από τα σημαντικότερα προβλήματα που αντιμετωπίζουν οι ναυτιλιακές εταιρίες είναι αυτό της κατανάλωσης καυσίμων των πλοίων. Το κόστος για ένα ταξίδι και τα καύσιμα που απαιτούνται έχουν αυξηθεί οπότε οι ναυτιλιακές επιχειρήσεις στρέφουν το ενδιαφέρον τους στην εύρεση διαδρομών που θα τους εξοικονομούν χρήματα, καταναλώνοντας προφανώς λιγότερα

καύσιμα. Αυτή η εύρεση βέλτιστων διαδρομών έτσι ώστε να εξοικονομούνται καύσιμα μόνο εύκολη δεν θεωρείται λόγω πολλών παραγόντων που ωθούν το πλοίο στο να διανύσει περισσότερα ναυτικά μίλια και να αλλάζει συνεχώς την ταχύτητα με την οποία κινείται. Τέτοιοι παράγοντες που επηρεάζουν την ταχύτητα του πλοίου είναι οι καιρικές συνθήκες, ο κυματισμός, τα ρεύματα που υπάρχουν σε κάθε περιοχή καθώς και περιεκτικότητα αλατιού και η πυκνότητα των υδάτων που διαφέρουν από χρόνο σε χρόνο και από περιοχή σε περιοχή. Γενικά η αντοχή των πλοίων εξαρτάται από τους συγκεκριμένους και κάποιους άλλους παράγοντες οι οποίοι πρέπει να ληφθούν υπόψη στην έρευνα και να εξεταστούν για την σημαντικότητά τους.

Στην έρευνα που διεξήγαγαν οι *Misganaw Abebe et al* (2020) έγινε χρήση τεχνικών στατιστικής μηχανικής μάθησης και συγκεκριμένα τεχνικών παλινδρόμησης έτσι ώστε να προβλεφθεί η ταχύτητα των πλοίων κατά την διάρκεια του ταξιδιού. Αυτή η πρόβλεψη μπορεί να βοηθήσει στις ναυτιλιακές εταιρίες έτσι ώστε να βρεθούν καλύτερες και βέλτιστες διαδρομές για την υλοποίηση των ταξιδιών, διαδρομές δηλαδή που θα απαιτούν χαμηλή κατανάλωση καυσίμων και επομένως μειωμένο κόστος. Τα δεδομένα της έρευνας προήλθαν από το αυτόματο σύστημα αναγνώρισης της θέσης των πλοίων σε πραγματικό χρόνο («AIS») και από μεσημεριανές αναφορές του καιρού για περίοδο ενός έτους. Τα δεδομένα από το σύστημα «AIS» αποτελούνται από στατικές πληροφορίες, δυναμικές πληροφορίες καθώς και πληροφορίες πλοήγησης των καραβιών. Οι στατικές πληροφορίες περιλαμβάνουν τους αριθμούς αναγνώρισης του πλοίου όπως την ταυτότητα ναυτιλιακής κινητής υπηρεσίας (MMSI) και τον αριθμό του διεθνούς ναυτιλιακού οργανισμού (IMO), το διακριτικό κλήσης και το όνομα, ενώ περιλαμβάνει ακόμα και τους τύπους και τις διαστάσεις του πλοίου. Η ενημέρωση των στατικών δεδομένων γίνεται χειροκίνητα μιας και αλλάζουν σπανίως. Τα δυναμικά δεδομένα περιλαμβάνουν επιχειρησιακές πληροφορίες που σχετίζονται με τη πλοήγηση ενός πλοίου. Τα δεδομένα αυτά συλλέγονται μέσα σε κάποιο χρονικό διάστημα και ενημερώνονται αυτόματα σύμφωνα με την κατάσταση πλοήγησης του πλοίου. Το δείγμα αποτελούταν από 76 σκάφη εκ των οποίων τα 14 ήταν δεξαμενόπλοια (Tankers) και τα 62 ήταν φορτηγά πλοία (Cargo ships). Για την εκτίμηση της ταχύτητας των πλοίων, χρησιμοποιήθηκαν συνολικά 26 μεταβλητές. Σκοπός της έρευνας ήταν να εκτιμηθεί και να αξιολογηθεί η ταχύτητα ανά έδαφος («SOG») με την οποία υλοποιούν τα ταξίδια τους τα πλοία με βάση κάποιες παραμέτρους, όπως αυτές που αναφέρθηκαν προηγουμένως, έτσι ώστε στην συνέχεια τα πορίσματα της έρευνας να συμβάλλουν στην τελική αξιολόγηση απόδοσης του πλοίων και εν τέλει να χρησιμοποιηθούν για την εύρεση βέλτιστης διαδρομής για εξοικονόμηση καυσίμων. Αφού έγινε η κατάλληλη επεξεργασία των δεδομένων οι αλγόριθμοι παλινδρόμησης που χρησιμοποιήθηκαν ήταν οι:

I. Linear Regression
II. Polynomial Regression
III. Decision Tree regressors
IV. Gradient Boosting Regressors (GBRs)
V. Extreme Gradient Boosting (XGBRs)
VI. Random Forest Regressors (RFRs)
VII. Extra Trees Regressors (ETRs)

Για να συγκριθούν και να αξιολογηθούν τα αποτελέσματα των παραπάνω τεχνικών μηχανικής μάθησης χρησιμοποιήθηκε το στατιστικό μέτρο αξιολόγησης ενός μοντέλου παλινδρόμησης, ο συντελεστής προσδιορισμού R^2.

Οι τιμές του συντελεστή προσδιορισμού για την κάθε τεχνική που εφαρμόστηκε φαίνεται παρακάτω στον παρακάτω πίνακα 1.

Μοντέλο	R^2
Linear Regression	0,237
3rd order Polynomial Regression	0,40
Decision Tree Regressors	0,964
GBRs	0,964
XGBRs	0,969
Random Forest Regressors	0,983
Extra Trees Regressors	0,984

Πίνακα 1: Αξιολόγηση των μοντέλων παλινδρόμησης που χρησιμοποιήθηκαν

Τα αποτελέσματα της μελέτης έδειξαν ότι την καλύτερη προσαρμογή στα δεδομένα φαίνεται να την πετυχαίνει το μοντέλο «Extra Trees Regressor» αφού η τιμή του συντελεστή προσδιορισμού ισούται με 0,984. Δηλαδή αυτή η τιμή σημαίνει ότι το 98,4% της συνολικής μεταβλητότητας της εξαρτημένης μεταβλητής «SOG» (ταχύτητα πλοίου ανά έδαφος) ερμηνεύεται από τις ανεξάρτητες μεταβλητές, δηλαδή τις παραμέτρους που εξετάζονται. Πολύ καλή προσαρμογή έχουν και τα μοντέλα «DTRs», «GBRs», «XGBRs» και «RFRs». Την χειρότερη επίδοση είχαν οι τεχνικές παλινδρόμησης «Linear Regression» και «Polynomial Regression» εξαιτίας της υψηλής μη γραμμικής τάσης μεταξύ της ταχύτητας του πλοίου και του χρόνου. Η χρήση της τεχνικής «Decision Trees Forest regressors» δεν προτείνεται σύμφωνα με την έρευνα, διότι παρουσιάζει μεγαλύτερη μεταβλητότητα σε σχέση με τις υπόλοιπες τεχνικές που πέτυχαν μεγάλη ακρίβεια. Συνεπώς τα πέντε από τα επτά μοντέλα παλινδρόμησης που εφαρμόστηκαν πέτυχαν μεγάλη ακρίβεια στην πρόβλεψη της ταχύτητας των πλοίων κατά την πλεύση τους, όταν αυτά ταξιδεύουν υπό διαφορετικές καιρικές συνθήκες, όταν υπάρχουν ρεύματα και κυματισμοί και κάποιοι άλλοι παράγοντες που επηρεάζουν την απόσταση που διανύουν τα πλοία.

4.2 2η Μελέτη: Τεχνικές μηχανικής μάθησης για την πρόβλεψη ταχύτητας των πλοίων με βάση ορισμένους παράγοντες που επηρεάζουν την λειτουργική απόδοση των πλοίων

Σε αυτή την μελέτη των *Ameen M. Bassam et al* (2022) χρησιμοποιήθηκαν τεχνικές μηχανικής μάθησης και πιο συγκεκριμένα, χρησιμοποιήθηκαν μοντέλα παλινδρόμησης που αποσκοπούσαν στην πρόβλεψη ταχύτητας των πλοίων με βάση κάποιους παράγοντες που επηρέαζαν την αποδοτική λειτουργία του σχετικά με την κατανάλωση καυσίμων και την διεκπεραίωση των δρομολογίων του. Σκοπός και αυτής της μελέτης ήταν να προβλεφθεί η ταχύτητα του πλοίου έτσι ώστε να ελαττωθεί όσο περισσότερο γίνεται η κατανάλωση καυσίμων και συνεπώς η εκπομπή ρύπων στην ατμόσφαιρα καθώς και το κόστος εκτέλεσης των δρομολογίων. Για τις ανάγκες της έρευνας συλλέχθηκαν δημοσίως διαθέσιμα δεδομένα που υπήρχαν ήδη από το εγχώριο ferry boat «M/S Smyril» το οποίο εκτελούσε δρομολόγια γύρω από τις νήσους Φερόε. Τα δεδομένα προήλθαν από δρομολόγια που εκτελούσε το εν λόγω πλοίο δύο με τρεις φορές την μέρα για το διάστημα μεταξύ 16 Φεβρουαρίου και 21 Απριλίου για το έτος

2010. Το πλοίο υλοποίησε συνολικά περίπου 250 δρομολόγια. Μετά από την κατάλληλη προ επεξεργασία των δεδομένων, το συνολικό μέγεθος του δείγματος ήταν 2654 παρατηρήσεις και 10 παράμετροι οι οποίοι φαίνεται να επηρέαζαν περισσότερο την ταχύτητα του πλοίου καθώς και την αποδοτική εκτέλεση των δρομολόγιών του. Οι αλγόριθμοι που χρησιμοποιήθηκαν για την πρόβλεψη της ταχύτητας του πλοίου ήταν οι:

1. Multiple Linear Regression
2. Regression Trees
3. Regression Trees Ensembles
4. Gaussian Process Regression models
5. Support Vector Machines

Για την αξιολόγηση των παραπάνω μοντέλων παλινδρόμησης χρησιμοποιήθηκαν τα στατιστικά μέτρα R^2, «Mean Squared Error (MSE)», «Mean Absolute Error (MAE)», καθώς και το «Root Mean Square Error (RMSE)».
Το μοντέλο με την καλύτερη επίδοση για την πρόβλεψη ταχύτητας του πλοίου φαίνεται να την έχει το μοντέλο «Gaussian Process Regression» με συντελεστή προσδιορισμού R^2 ίσο με 0,91 και η τιμή του μέτρου RMSE ήταν ίση επίσης με 0,91. Την χειρότερη επίδοση φαίνεται να την είχε το μοντέλο «Support Vector Machines» αφού η τιμή του συντελεστή προσδιορισμού R^2 ήταν ίση με 0,51. Τον λιγότερο χρόνο υπολογισμού για την πρόβλεψη της ταχύτητας του πλοίου είχαν οι αλγόριθμοι «Regression Trees» και «Regression Trees Ensembles».

4.3 3η Μελέτη: Πρόβλεψη καθυστερήσεων των φορτηγών πλοίων στην παράδοση των εμπορευμάτων τους χρησιμοποιώντας τεχνικές παλινδρόμησης και κατηγοριοποίησης

Πολύ συχνά η μεταφορά των εμπορευμάτων μέσω των φορτηγών πλοίων καθυστερεί αρκετά μεγάλο χρονικό διάστημα συγκριτικά πάντα με τον αναμενόμενο χρόνο παράδοσης των φορτίων. Αυτό είναι ένα ακόμη πρόβλημα που μαστίζει τον ναυτιλιακό κλάδο. Μάλιστα η καθυστέρηση των φορτηγών πλοίων αναμένεται να αυξηθεί με γρήγορους ρυθμούς εξαιτίας της αυξημένης κυκλοφοριακής συμφόρησης που υπάρχει στα λιμάνια και στα κυριότερα σημεία ελέγχου των πλοίων όπως στην διώρυγα του Σουέζ, του Παναμά, κλπ. Επιπλέον το φαινόμενο αυτό φαίνεται ότι επηρεάζει και διογκώνει η ολοένα και περισσότερο αυξανόμενη συχνότητα εμφάνισης ακραίων καιρικών συνθηκών. Συνεπώς πολλές ναυτιλιακές επιχειρήσεις εστιάζουν στο να προβλέψουν αυτές τις καθυστερήσεις των φορτηγών πλοίων για την μεταφορά των εμπορευμάτων. Η πρόβλεψη αυτή σχετικά με την συμφόρηση των πλοίων είναι κρίσιμης σημασίας διότι γνωρίζοντας τις καθυστερήσεις των πλοίων, οι αποστολείς των προϊόντων θα μπορούν να διαλέξουν άλλους μεταφορικούς τρόπους, άλλα λιμάνια και διαφορετικές διαδρομές ώστε να εξοικονομήσουν και χρήματα και φυσικά χρόνο. Επιπροσθέτως η μεταφορά εμπορευμάτων μέσω των φορτηγών πλοίων παίζει καθοριστικό ρόλο στην παγκόσμια μεταφορά προϊόντων αφού συνδέει ολόκληρη την εφοδιαστική αλυσίδα από την διαδικασία παραγωγής μέχρι τους τελικούς καταναλωτές.

Αυτή η μελέτη των *Adrian Viellechner και Stefan Spinler* (2020) αποσκοπεί στην πρόβλεψη καθυστερήσεων των φορτηγών πλοίων για την παράδοση των φορτίων τους μεταξύ Ευρώπης και Ασίας λόγω κάποιων παραγόντων όπως οι καιρικές συνθήκες που επικρατούν. Για την πρόβλεψη αυτή θεωρήθηκαν κατάλληλοι κάποιοι αλγόριθμοι κατηγοριοποίησης και μοντέλα παλινδρόμησης. Για την ανάγκη της μελέτης συλλέχθηκαν δορυφορικά δεδομένα από το σύστημα «AIS» σχετικά με την θέση των πλοίων σε πραγματικό χρόνο. Σε γενικά πλαίσια συλλέχθηκαν δεδομένα από 75.814 αποστολές εμπορευματοκιβωτίων που αποχωρούσαν ή έφταναν σε λιμάνια της Ευρώπης και της Ασίας μεταξύ του Φεβρουαρίου του 2016 και του Αυγούστου του 2018. Προκειμένου τα αποτελέσματα της έρευνας να είναι πιο αξιόπιστα και για την διαφάνειά της θεωρήθηκε προτιμότερο να χρησιμοποιηθούν τα δεδομένα αποστολών φορτιών που έγιναν απευθείας χωρίς άλλες μεταφορτώσεις και τα οποία έλαβαν χώρα στα εκατό πιο πολυσύχναστα λιμάνια παγκοσμίως. Οπότε τελικά για την έρευνα από τα 75.814 δεδομένα των αποστολών εμπορευματοκιβωτίων χρησιμοποιήθηκαν μόνο τα 1.851 για την πρόβλεψη της καθυστέρησης των φορτηγών πλοίων. Επίσης χρησιμοποιήθηκαν συνολικά 315 επεξηγηματικές μεταβλητές εκ των οποίων τελικά μετά από τον καθαρισμό των δεδομένων έμειναν στην ανάλυση μόνο οι 166. Τα μοντέλα μηχανικής μάθησης που χρησιμοποιήθηκαν χωρίστηκαν σε δύο κατηγορίες. Σε μοντέλα κατηγοριοποίησης και παλινδρόμησης και φαίνονται στον πίνακα 2.

Αλγόριθμοι Κατηγοριοποίησης (Classification models)	Μοντέλα παλινδρόμησης (Regression models)
Neural Networks (NN)	SVM polynomial
Log Regression (LogR)	SVM radial
Random Forrest (RF)	Random Forrest (RF)
SVM polynomial	Lasso Regression (LasR)
SVM radial	Elastic Net Regression (ENR)
SVM linear	SVM radial
SVM sigmoid	SVM sigmoid
	Ridge Regression (RR)
	Linear Regression (LR)
	Neural Networks (NN)

Πίνακας 2: Μοντέλα κατηγοριοποίησης και παλινδρόμησης που εφαρμόστηκαν στην μελέτη

Για την αξιολόγηση των μοντέλων κατηγοριοποίησης και παλινδρόμησης χρησιμοποιήθηκαν κάποια μέτρα αξιολόγησης. Πιο συγκεκριμένα για τους αλγορίθμους κατηγοριοποίησης χρησιμοποιήθηκαν τα μέτρα «Root Mean Squared Error (RMSE)», «Accuracy» και «Sensitivity». Αντίστοιχα για τα μοντέλα παλινδρόμησης χρησιμοποιήθηκαν οι δείκτες αξιολόγησης «Mean Absolute Error» και «Root Mean Squared Error». Οι δείκτες RMSE και MAE βοηθούν στην σύγκριση των εκτιμώμενων τιμών που προβλέφθηκαν, με τις πραγματικές τιμές.

Στην ναυτιλία ο όρος «Sensitivity» δηλαδή «Ευαισθησία», εκφράζει το πόσο συχνά προβλέπεται καθυστέρηση των πλοίων στην περίπτωση που όντως ένα πλοίο φτάσει με καθυστέρηση στον προορισμό του. Γενικά για να είναι καλά τα μοντέλα θα πρέπει οι τιμές των δεικτών Accuracy και Sensitivity να είναι υψηλές ενώ οι τιμές των RMSE και MAE θα πρέπει να είναι όσο το

δυνατόν μικρότερες. Τα αποτελέσματα για την κάθε τεχνική μηχανικής μάθησης και για το κάθε μοντέλο φαίνονται στους πίνακες 3 και 4 στην συνέχεια.

Μοντέλο	***RMSE***	***Accuracy***	***Sensitivity***
Logistic Regression	0,41	0,75	0,75
Random Forrest	0,43	0,81	0,65
Neural Networks	0,41	0,77	0,78
SVM sigmoid	0,61	0,63	0,96
SVM linear	0,50	0,74	0,48
SVM polynomial	0,46	0,79	0,68
SVM radial	0,48	0,77	0,64

Πίνακας 3: Μέτρα αξιολόγησης των μοντέλων κατηγοριοποίησης που εφαρμόστηκαν

Μοντέλο	***RMSE***	***MAE***
Neural Networks	0,52	0,34
Random Forrest	0,63	0,40
Lasso Regression	0,79	0,56
Elastic Net Regression	0,79	0,56
Ridge Regression	0,80	0,56
Linear Regression	0,67	0,49
SVM polynomial	0,43	0,26
SVM radial	0,53	0,34
SVM linear	0,70	0,47
SVM sigmoid	0,93	0,61

Πίνακας 4: Μέτρα αξιολόγησης των μοντέλων παλινδρόμησης που εφαρμόστηκαν

Τα συμπεράσματα της μελέτης ήταν ότι την καλύτερη απόδοση από τα μοντέλα κατηγοριοποίησης φαίνεται να την είχε ο αλγόριθμος τεχνητών νευρωνικών δικτύων «Neural Networks» με τιμή RMSE ίση με 0,41 και σκορ πρόβλεψης της καθυστέρησης των φορτηγών πλοίων «Accuracy» ίσο με 0,77. Από τα μοντέλα παλινδρόμησης, την καλύτερη απόδοση είχε το μοντέλο «SVM polynomial» με τιμή RMSE ίση με 0,43 και τιμή MAE ίση με 0,26.

4.4 4η Μελέτη: Πρόβλεψη της κατανάλωσης καυσίμων πλοίων κατά την διάρκεια των ταξιδιών

Ένα αρνητικό φαινόμενο το οποίο είναι σε έξαρση τις τελευταίες δεκαετίες λόγω της εκπομπής ρύπων στην ατμόσφαιρα, αυτό του θερμοκηπίου, ταλανίζει ολόκληρη την επιστημονική κοινότητα. Εδώ και αρκετά χρόνια γίνονται προσπάθειες για να μειωθεί η εκπομπή ρύπων στο περιβάλλον. Στην προσπάθειά της να μειώσει την κατανάλωση καυσίμων των πλοίων, η ναυτιλιακή βιομηχανία προσπαθεί παράλληλα να βρει τρόπους ώστε να εκπέμπονται λιγότερα

αέρια του θερμοκηπίου από τα πλοία. Αυτό μπορεί να επιτευχθεί με διάφορους τρόπους όπως με την κατασκευή καινούργιων ενεργειακά αποδοτικών πλοίων τα οποία θα καταναλώνουν λιγότερη ενέργεια και συνεπώς καύσιμα. Οπότε με την μειωμένη κατανάλωση καυσίμων θα μειώνονται και οι ρύποι που εκπέμπονται. Όμως αυτή η δημιουργία νέων αποδοτικών πλοίων, παρότι ήδη έχουν φτιαχτεί αρκετά, θα πάρει πολύ χρόνο μέχρι να είναι εξοπλισμένη η πλειοψηφία του παγκόσμιου στόλου με τέτοιου είδους πλοία.

Την λύση λοιπόν σε αυτό το θέμα της εκπομπής ρύπων λόγω αυξημένης κατανάλωσης καυσίμων, προσπαθεί να δώσει η μηχανική μάθηση και δύο προσεγγίσεις της με τι οποίες γίνεται πρόβλεψη της κατανάλωσης καυσίμων. Σκοπός αυτής της έρευνας των *Zhihui Hu et al* (2019) ήταν να εκτιμηθεί η κατανάλωση καυσίμων των πλοίων όταν ταξίδευαν. Τα δύο μοντέλα που χρησιμοποιήθηκαν ήταν το «Back Propagation Neural Network (BPNN)» και το «Gaussian Process Regression (GPR)». Το μοντέλο BPNN πρόκειται για ένα νευρωνικό δίκτυο το οποίο αυτήν την στιγμή χρησιμοποιείται ευρέως. Τα μοντέλα παλινδρόμησης διεργασιών Gauss (GPR) έχουν χρησιμοποιηθεί ευρέως σε εφαρμογές μηχανικής μάθησης λόγω της ευελιξίας αναπαράστασής τους και των εγγενών μέτρων αβεβαιότητας σε σχέση με τις προβλέψεις. Για την ανάγκη της διεξαγωγής της μελέτης, συλλέχθηκαν 24.386 δεδομένα κατανάλωσης καυσίμων από φορτηγά πλοία. Τελικά μετά από κατάλληλη επεξεργασία, χρησιμοποιήθηκαν μόνο τα 9.317 από τα 24.386 διαθέσιμα δεδομένα. Για την σύγκριση των μοντέλων έγινε χρήση της τιμής του συντελεστή προσδιορισμού R^2 και του απαιτούμενου χρόνου που χρειάζεται για την πρόβλεψη της κατανάλωσης καυσίμων από τα δύο μοντέλα. Τα αποτελέσματα της έρευνας για την ακρίβεια των προβλέψεων των μοντέλων έδειξαν ότι η τιμή του συντελεστή προσδιορισμού R^2 για το μοντέλο BPNN ήταν ίση με 0,9817 ενώ για το μοντέλο GPR ήταν ίση με 0,9887. Παρ ’ότι φαίνεται το μοντέλο GPR να έχει ελαφρώς καλύτερη προβλεπτική ικανότητα σε σχέση το μοντέλο BPNN δεν είναι κατάλληλο για να εφαρμοστεί λόγω του απαιτούμενου χρόνου που θέλει για να υλοποιήσει την πρόβλεψη. Πιο συγκεκριμένα, για να κάνει την πρόβλεψη κατανάλωσης καυσίμων, ο αλγόριθμος GPR χρειάζεται χρόνο 2236,4 δευτερόλεπτα ενώ το μοντέλο BPNN χρειάζεται μόλις 14,7 δευτερόλεπτα. Άρα για μια «online» πρόβλεψη της κατανάλωσης καυσίμων των πλοίων κατά την διάρκεια του ταξιδιού τους είναι προτιμότερο το μοντέλο «Back Propagation Neural Network».

4.5 5η Μελέτη: Πρόβλεψη της κατανάλωσης καυσίμων χρησιμοποιώντας τεχνικές μηχανικής μάθησης : Πρόβλεψη για φορτηγό πλοίο - Κοντέινερ (Container)

Εκτός από την εκπομπή ρύπων στο περιβάλλον που φυσικά είναι πολύ σημαντικό ζήτημα για τις ναυτιλιακές εταιρίες παγκοσμίως, ακόμα ένα θέμα που αναφέρθηκε και σε προηγούμενη μελέτη, λόγω της αυξημένης κατανάλωσης καυσίμων είναι και το κόστος των καυσίμων. Προφανώς τα καύσιμα έχουν κάποιο σημαντικό, μεγάλο κόστος και όπως είναι λογικό ο στόχος των εταιριών είναι να μειωθεί η κατανάλωση καυσίμων και συνεπώς το κόστος των ταξιδιών. Υπάρχουν πολλοί παράγοντες σε κάθε ταξίδι που επηρεάζουν τον ρυθμό με τον οποίο καταναλώνονται τα καύσιμα όπως το βάρος του φορτίου, το βύθισμα του πλοίου, οι καιρικές συνθήκες κλπ, οι οποίοι δυσκολεύουν το έργο της πρόβλεψης της «οικονομικότερης» κατανάλωσης καυσίμων για ένα επικερδές, αποδοτικό και με λιγότερα έξοδα ταξίδι.

Σε αυτήν την μελέτη των *Tayfun Uyanik et al* (2020) σκοπός ήταν να προβλεφθεί η κατανάλωση καυσίμων που απαιτούταν, έτσι ώστε αφενός να μειωθεί ένα μεγάλο κομμάτι του λειτουργικού κόστους του πλοίου, και αφετέρου με την μείωση της κατανάλωσης καυσίμων να γίνει ένα πλοίο πιο ενεργειακά αποδοτικό και με αυτόν τον τρόπο να μειωθεί και η διοχέτευση ρύπων στο περιβάλλον. Η συγκεκριμένη μελέτη εφαρμόστηκε για ένα φορτηγό πλοίο και την κατανάλωση καυσίμων του κύριου κινητήρα του. Τα δεδομένα της μελέτης προήλθαν από μεσημεριανές αναφορές καιρού, από το ημερολόγιο κινητήρα καθώς και από πληροφορίες των αισθητήρων του πλοίου. Συνολικά εξετάστηκαν 75 παράμετροι και 724 παρατηρήσεις. Μετά την κατάλληλη επεξεργασία των δεδομένων εφαρμόστηκαν για την πρόβλεψη κατανάλωσης καυσίμων του πλοίου, κάποιοι αλγόριθμοι μηχανικής μάθησης οι οποίοι ήταν:

- Multiple Linear Regression (MLR)
- Ridge Regression
- Bayesian Ridge Regression
- Kernel Ridge Regression
- Lasso Regression
- Support Vector Regression (SVR)
- K Nearest Neighbors (KNN)
- Multilayer Perceptron Regression (MPR)
- Decision Tree Regressor (DTR)
- Random Forrest Regression (RF)
- Ada Boost Regression
- Hist Gradient Boost Regression (HGBR)
- Gradient Boost Regression (GBR)
- Elastic Net Regression

Για την αξιολόγηση της προβλεπτικής ικανότητας των μοντέλων χρησιμοποιήθηκαν τα στατιστικά μέτρα «Root Mean Square Error (RMSE)», «Mean Absolute Error (MAE)», και ο συντελεστής προσδιορισμού R^2. Τα αποτελέσματα του κάθε αλγορίθμου που χρησιμοποιήθηκε φαίνονται παρακάτω στον πίνακα 5.

Μοντέλο	***MAE***	***RMSE***	R^2
MLR	0,002	0,0001	0,99999
Ridge Regression	0,002	0,0001	0,99999
Bayesian Ridge Regression	0,003	0,0001	0,99999
Kernel Ridge Regression	0,003	0,0001	0,99999
Lasso Regression	0,330	0,3235	0,99983
SVR	0,051	0,0031	0,99999
KNN	1,008	3,9576	0,99803

MPR	1,812	9,7473	0,99516
DTR	0,428	1,0552	0,99947
RF	0,534	1,8286	0,99909
Ada Boost Regression	1,522	3,5264	0,99824
HGBR	0,827	5,5588	0,99724
GBR	0,311	1,0459	0,99948
Elastic Net	0,366	0,4235	0,99978

Πίνακας 5: Μέτρα αξιολόγησης μοντέλων μηχανικής μάθησης που εφαρμόστηκαν

Όλα τα μοντέλα που χρησιμοποιήθηκαν φαίνεται να έχουν πολύ ικανοποιητική προβλεπτική ικανότητα. Τα καλύτερα όμως μεταξύ αυτών, σύμφωνα με την έρευνα, είναι τα μοντέλα τους είδους Ridge Regression και το μοντέλο «Multiple Linear Regression». Τα μοντέλα «Decision Tree» και «Boosting Regression» παρότι έχουν πολύ καλή προβλεπτική ικανότητα, δεν έχουν την ίδια ικανότητα πρόβλεψης όσο τα «Ridge» μοντέλα εξαιτίας της συνεχούς δομής των δεδομένων. Το σύνολο δεδομένων δηλαδή αποτελείται από συνεχείς μεταβλητές, δηλαδή μεταβλητές που παίρνουν τιμές σε ένα διάστημα πραγματικών αριθμών (-∞,∞). Σε συνεχή δεδομένα καλύτερη προσαρμογή έχουν τα «Ridge» μοντέλα που προσαρμόστηκαν. Το λιγότερα ικανοποιητικό αποτέλεσμα δίνει ο αλγόριθμος «Multilayer Perceptron Regression» με συντελεστή προσδιορισμού R^2 ίσο με 0,99516.

4.6 6η Μελέτη: Εντοπισμός και διάγνωση δυσλειτουργίας του κινητήρα των πλοίων χρησιμοποιώντας ένα σύνολο αλγορίθμων μηχανικής μάθησης

Η ασφάλεια των ταξιδιών των πλοίων και η ασφαλής πλοήγησή τους γενικότερα είναι ο πρωταρχικός στόχος κάθε ναυτιλιακής εταιρίας όταν είναι να πραγματοποιηθεί κάποιο δρομολόγιο. Κατά την διάρκεια ενός ταξιδιού υπάρχουν αρκετοί παράγοντες που επηρεάζουν την ασφαλή πλοήγηση του πλοίου. Ένας από τους κυριότερους παράγοντες αν όχι ο κυριότερος, είναι η δυσλειτουργία της κύριας μηχανής του. Κάποιες φορές εντοπίζονται προβλήματα στην κύρια μηχανή του πλοίου που έχουν ως αποτέλεσμα την κατανάλωση περαιτέρω καυσίμων, καθώς και την επισφαλή πλεύση του λόγω των καιρικών και θαλάσσιων συνθηκών που επικρατούν. Αυτή η δυσλειτουργία που τυχαίνει μερικές φορές να παρουσιάζει ο κύριος κινητήρας του, οδηγεί όχι μόνο σε αύξηση του οικονομικού κόστους για την ναυτιλιακή εταιρία αλλά και σε μη ασφαλή υλοποίηση του ταξιδιού και συνεπώς στην εκδήλωση ατυχήματος. Κύριοι λόγοι βλάβης της μηχανής του πλοίου είναι η οξείδωση της μηχανής, η αποσύνθεσή της, η θερμική καταπόνησή της λόγω της συνεχούς λειτουργίας της όπως και η παραμόρφωση κάποιων τμημάτων της μετά από κάποιο διάστημα λειτουργίας. Επομένως, ένα ακόμη ζήτημα για τις ναυτιλιακές εταιρίες είναι ο έγκαιρος εντοπισμός τέτοιων βλαβών του κινητήρα σε διάφορα ταξίδια που υλοποιούνται, έτσι ώστε να αποφευχθούν ανεπιθύμητα, απροσδόκητα και καταστροφικά περιστατικά και ατυχήματα.

Σε αυτή την μελέτη των *G. Tsaganos et al* (2018) σκοπός ήταν ο εντοπισμός και η έγκαιρη διάγνωση βλάβης της κύριας μηχανής του πλοίου ώστε να επιλυθεί το πρόβλημα γρήγορα και να μην συμβούν ή εμφανιστούν άσχημα συμβάντα στην συνέχεια του ταξιδιού. Επίσης με την έγκαιρη διάγνωση της βλάβης της κύριας μηχανής του πλοίου και την έγκαιρη

επισκευή της, εκτός από την ασφαλή περάτωση του ταξιδιού, πιθανόν θα αποφευχθεί ο κίνδυνος για παραπάνω κατανάλωση καυσίμων και θα μειωθούν συνεπώς οι εκπομπές ρύπων στην ατμόσφαιρα όπως και το οικονομικός κόστος του δρομολογίου για την ναυτιλιακή εταιρία. Η έρευνα βασίστηκε στον προσδιορισμό της διαφοροποίησης των τιμών κάποιων παραμέτρων όσων αφορά τις τιμές πίεσης και θερμοκρασίας της κύριας μηχανής του πλοίου. Δηλαδή αν κάποιοι παράμετροι που εξετάζονταν, εμφάνιζαν κάποια απόκλιση, που σημαίνει, τιμές εκτός των ορίων των τιμών πίεσης και θερμοκρασίας της μηχανής του πλοίου τότε έπρεπε να ελεγχθούν τα αίτια που οδήγησαν σε αυτή την απόκλιση των τιμών των παραμέτρων αυτών. Το δείγμα της μελέτης προήλθε με την βοήθεια της μεθόδου προσομοίωσης. Προσομοιώθηκαν σενάρια αστοχίας – βλάβης της κύριας μηχανής ενός πλοίου με μοντέλο μηχανής «MAN BW 7S60C» η οποία πρόκειται για τύπο δίχρονης μηχανής χαμηλής ταχύτητας, χρησιμοποιώντας τον προσομοιωτή μηχανής του τμήματος μηχανικών στην εμπορική ακαδημία του Ασπροπύργου που βρίσκεται στην Ελλάδα. Γενικά τα περισσότερα είδη πλοίων όπως τα φορτηγά πλοία, τα δεξαμενόπλοια κ.ά., χρησιμοποιούν τέτοιου τύπου μηχανή εξαιτίας της υψηλής απόδοσής τους σε λειτουργία υψηλής ισχύος καθώς και λόγω της ανοχής τους στην ποιότητα των καυσίμων. Συνολικά το δείγμα συμπεριλάμβανε 1000 εγγραφές σφαλμάτων της κύριας μηχανής. Οι αλγόριθμοι μηχανικής μάθησης που χρησιμοποιήθηκαν για τον εντοπισμό βλάβης στον κύριο κινητήρα (κύρια μηχανή) του πλοίου ήταν οι εξής:

1. Naïve Bayes
2. C 4.5
3. Simple Cart
4. Locally Weighted Regression
5. Multilayer Perceptron
6. Sequential minimal optimization
7. MODLEN
8. MultiBoost classifier
9. AdaBoost classifier
10. Decorate classifier
11. Ensemble models (συνδυαστικά μαζί κάποια από τα παραπάνω μοντέλα)

Η αξιολόγηση της επίδοσης των μοντέλων που εφαρμόστηκαν έγινε με γνώμονα τα μέτρα «Accuracy» και «F measure». Συγκρίνοντας τα συγκεκριμένα μέτρα για κάθε μοντέλο μηχανικής μάθησης, το συμπέρασμα ήταν ότι την καλύτερη απόδοση είχε ο αλγόριθμος «Simple Cart» με τιμή του «F μέτρου» ίση με 0,955 και τιμή «Accuracy» ίση με 95,5% = 0,955. Την χειρότερη επίδοση φαίνεται να είχε ο αλγόριθμος «Multilayer Perceptron» αφού η τιμή του μέτρου F ήταν ίση με 0,53 ενώ η τιμή «Accuracy» ήταν ίση με 54,5% = 0,545. Όταν έγινε συνδυασμός κάποιων αλγορίθμων (ensemble models) βρέθηκε ότι το μοντέλο «AdaBoost – Simple Cart» είχε την καλύτερη επίδοση με την τιμή του F μέτρου να είναι ίση με 0,965 και η τιμή «Accuracy» να είναι ίση με 96,6% = 0,966. Αυτό το «ενωμένο» μοντέλο «AdaBoost – Simple Cart» βελτίωσε στην ουσία την προβλεπτική ικανότητα του αλγορίθμου «Simple Cart» στο να εντοπίζει βλάβες στην κύρια μηχανή του πλοίου κατά 1,1%. Βέβαια πέρα από την βελτίωση της αποδοτικότητας του μοντέλου, η κατασκευή του συνδυασμένου μοντέλου «AdaBoost – Simple Cart» αύξησε και τον χρόνο που απαιτείται για την κατασκευή του.

4.7 7η Μελέτη: Μοντέλα πρόβλεψης ατυχημάτων και απρόοπτων περιστατικών κατά την διάρκεια δρομολόγιών κρουαζιερόπλοιων χρησιμοποιώντας τεχνικές μηχανικής μάθησης

Κατά την διάρκεια ταξιδιών κρουαζιερόπλοιων συνήθως δεν συμβαίνουν ατυχήματα και απρόοπτα περιστατικά. Όταν όμως συμβαίνουν ατυχήματα, αυτά τις περισσότερες φορές είναι καταστροφικά, προκαλώντας απώλειες ζωών, ρύπανση του περιβάλλοντος καθώς και υλικές ζημιές ή ολική ζημιά στα πλοία.

Σε αυτήν την μελέτη των *Zhaoqian Su et al* (2022), σκοπός ήταν να ενισχυθεί η ασφάλεια των ταξιδιών των κρουαζιερόπλοιων στην Κίνα χρησιμοποιώντας μοντέλα μηχανικής μάθησης που θα μπορούσαν να προβλέψουν τυχόν ατυχήματα ή απρόοπτα περιστατικά κατά την διάρκεια λειτουργίας των πλοίων στο διεθνές λιμάνι «Shanghai Wusongkou (SWICP)». Το σύνολο δεδομένων της μελέτης αποτελείται από συνολικά 1497 περιπτώσεις ατυχημάτων και τέτοιων απροσδόκητων περιστατικών στο λιμάνι «Shanghai Wusongkou (SWICP)» κατά την περίοδο μεταξύ 2012 και 2019. Για την πρόβλεψη ατυχημάτων κατά την διάρκεια λειτουργίας των πλοίων στο συγκεκριμένο κινέζικο λιμάνι χρησιμοποιήθηκαν οι τεχνικές:

- Linear Regression (LR)
- Decision Tree
- Random Forrest
- AdaBoost
- KNN Regressor
- Bootstrap Aggregating
- Extra Tree Regressor (ETR)
- Extreme Gradient Boosting (XGBoost)
- LightGBM Regressor
- Gradient Boost

Για την αξιολόγηση των παραπάνω μοντέλων χρησιμοποιήθηκαν τα μέτρα R^2 και «Root Mean Square Error (RMSE)». Τα καλύτερα μοντέλα με την υψηλότερη τιμή R^2 και την χαμηλότερη τιμή RMSE είχαν τα μοντέλα Linear Regression, KNN Regressor και Extra Tree regressor με τιμές R^2 ίσες με 0.79, 0.75 και 0.71 αντίστοιχα. Τα χειρότερα μοντέλα της μελέτης με τις πιο χαμηλές τιμές του συντελεστή προσδιορισμού R^2 και τις υψηλότερες τιμές RMSE είχαν τα μοντέλα Decision Tree, LightGBM και XGBoost. Τελικά για την μελέτη πρόβλεψης ατυχημάτων την πιο καλή προβλεπτική ικανότητα είχε το συνδυασμένο μοντέλο KNN+LR+ETR με τιμή R^2 ίση με 0,81 και τιμή RMSE ίση με 7,21.

4.8 8η Μελέτη: Εντοπισμός απάτης στην ναυτιλιακή βιομηχανία χρησιμοποιώντας αλγορίθμους μηχανικής μάθησης

Η συγκεκριμένη μελέτη των *Ganesan Subramaniam* και *Moamin A. Mahmoud* (2021) αποσκοπούσε στο να εντοπιστούν περιπτώσεις απάτης όπως λαθρεμπορίου και πλαστής δήλωσης φορτίων κατά την αποστολή των φορτίων στους προορισμούς τους. Για τις ανάγκες της μελέτης τα δεδομένα δημιουργήθηκαν μέσω προσομοίωσης που βασίστηκε σε μελέτη ενός διεθνούς

οργανισμού logistics και συγκεκριμένα στα ιστορικά του στοιχεία, που αφορούσαν την προέλευση και τον προορισμό των εμπορευμάτων για το διάστημα 2012 μέχρι 2017. Συνολικά χρησιμοποιήθηκαν 1500 δεδομένα φορτίων και παράμετροι που αφορούσαν το γεωγραφικό μήκος και πλάτος των τοποθεσιών προέλευσης και προορισμού των φορτίων. Στόχος των μοντέλων κατηγοριοποίησης που εφαρμόστηκαν ήταν να διαχωρίσουν τις αποστολές φορτίων σε περιπτώσεις απάτης ή μη απάτης αναλόγως με την τοποθεσία προέλευσης και προορισμού των φορτίων. Δηλαδή οι περιοχές προέλευσης και προορισμού είχαν προσδιοριστεί ως τοποθεσίες που γίνονται ή δεν γίνονται συχνά απάτες και αυτή η δίτιμη μεταβλητή που διαχώριζε με αυτό τον τρόπο τις περιοχές, ήταν η μεταβλητή που αποσκοπούσαν οι αλγόριθμοι να προβλέψουν όταν εισέρχονταν νέα δεδομένα. Οι αλγόριθμοι μηχανικής μάθησης που χρησιμοποιήθηκαν ήταν οι κάτωθι:

- Naïve Bayes
- Neural Network
- Decision Trees
- Logistic Regression
- Support Vector Machines (SVM)
- K-Nearest Neighbors (k-NN)

Την καλύτερη προβλεπτική ικανότητα στο να διαχωρίσει τις αποστολές φορτίων σε περιπτώσεις απάτης ή μη απάτης, φαίνεται να την είχε ο αλγόριθμος k-NN αφού το accuracy που πέτυχε ήταν ίσο με 98,4%. Όταν μάλιστα η παράμετρος k του αλγορίθμου k-NN ήταν ίση με 1 ή 2 τα αποτελέσματα ήταν πολύ ικανοποιητικά ως προς την προβλεπτική ικανότητα. Όταν όμως το k αυξανόταν κι άλλο, τότε το accuracy μειωνόταν.

4.9 9η Μελέτη: Εντοπισμός πλοίων στο λιμάνι της Αγίας Βαρβάρας χρησιμοποιώντας μεθόδους μηχανικής μάθησης

Αρκετά συχνά τα περιβαλλοντικά δεδομένα δεν είναι επαρκή για τον ακριβή εντοπισμό των πλοίων όταν αυτά εκτελούν δρομολόγια σε βαθιά νερά. Σε σχέση με τις παραδοσιακές μεθόδους για τον εντοπισμό πλοίων, οι μέθοδοι μηχανικής μάθησης φαίνεται να λειτουργούν καλύτερα. Μέχρι τώρα ο εντοπισμός ενός πλοίου που βρισκόταν σε ρηχά νερά γινόταν χρησιμοποιώντας μεθόδους μηχανικής μάθησης οι οποίοι μαθαίνουν μια σχέση διάδοσης την οποία λαμβάνουν μέσω της ακουστικής πίεσης των πλοίων σε κάθετη διάταξη και μέσω του συστήματος GPS τους. Όμως ο εντοπισμός αυτός δεν είναι τόσο εύκολος σε βαθιά ύδατα, ειδικά όταν τα περιβαλλοντικά δεδομένα είναι περιορισμένα.

Αυτή η μελέτη των *Haiqiang Niu et al* (2017) προσφέρεται να δώσει λύση στο πρόβλημα εντοπισμού πλοίων στο κανάλι της Αγίας Βαρβάρας όπου τα νερά είναι βαθιά, δηλαδή βάθους περίπου 550 με 600 μέτρα και δεν υπάρχουν επαρκή περιβαλλοντικά δεδομένα. Για την πρόβλεψη τοποθεσίας των πλοίων στο κανάλι της Αγίας Βαρβάρας εφαρμόστηκε ένα νευρωνικό δίκτυο εμπρόσθιας τροφοδότησης (Feed Forward Neural Network) το οποίο είχε 1 hidden layer και 2048 νευρώνες καθώς και η τεχνική κατηγοριοποίησης «Support Vector Machine». Για να εντοπιστούν τα πλοία, αρχικά, τα καταγεγραμμένα δεδομένα πίεσης που προέρχονται από τα πλοία, διαμορφώνονται σε κανονικοποιημένους δειγματικούς πίνακες συνδιακυμάνσεων, που έχουν σαν σκοπό να αφαιρεθεί η επίδραση των ευρών πηγών θορύβου από τα πλοία, και στην

συνέχεια διανυσματοποιούνται έτσι ώστε να δημιουργηθούν τα δεδομένα εισόδου των δύο μοντέλων που εφαρμόστηκαν. Για τον εντοπισμό των πλοίων ελέγχονται και προβλέπονται όλες οι πιθανές υποθαλάσσιες ακουστικές πηγές θορύβου που προέρχονται από τα πλοία. Για τις ανάγκες της μελέτης χρησιμοποιήθηκαν δεδομένα υποθαλάσσιου θορύβου από πλοία που είτε εισχωρούσαν είτε αποχωρούσαν από το λιμάνι «Los Angels» μεταξύ 7 και 20 Σεπτεμβρίου του 2016. Τα πλοία που διερχόντουσαν σε μία από τις δύο καλά καθορισμένες ναυτικές λωρίδες, χρησιμοποιήθηκαν ως πρωτογενείς ακουστικές πηγές στην μελέτη. Τελικά, τρεις διαδρομές πλοίων που καταγράφτηκαν σε διαφορετικούς χρόνους από ξεχωριστά πλοία, χρησιμοποιήθηκαν για την διαμόρφωση του συνόλου εκπαίδευσης (training data) και δοκιμής (test data) των δύο μεθόδων μηχανικής μάθησης. Τα δεδομένα της πρώτης διαδρομής όπου ήταν συνολικά 1956, χρησιμοποιήθηκαν για το σύνολο εκπαίδευσης των μοντέλων. Τα δεδομένα της δεύτερης και της τρίτης διαδρομής που ήταν συνολικά 260 και 300 αντίστοιχα, χρησιμοποιήθηκαν για το σύνολο δοκιμής (test data) των μοντέλων. Το σύνολο δοκιμής των αλγορίθμων ουσιαστικά διασπάστηκε σε δύο, το ένα απαρτιζόταν από τα 260 δεδομένα της 2ης διαδρομής και το άλλο απαρτιζόταν από τα 300 δεδομένα της 3ης διαδρομής. Στην εν λόγω μελέτη εξετάστηκαν δύο εύρη συχνοτήτων διάδοσης υποθαλάσσιου θορύβου από τα πλοία για τον εντοπισμό τους. Αυτό έγινε διότι αναλόγως την εμβέλεια του πλοίου, χανόταν το σήμα μετάδοσης του θορύβου που καταγραφόταν από αυτό. Έτσι λοιπόν, η μία ζώνη ήταν 53-200 Hz ενώ η άλλη ήταν 203-305 Hz με 3 Hz προσαύξηση. Αυτές οι δύο ζώνες είναι στην ουσία η μετάδοση σήματος (υποθαλάσσιου θορύβου) από τα πλοία καθώς και η απώλεια του σήματος αυτού, η οποία προσομοιώθηκε με την μορφή ενός μοντέλου κυματοδηγού. Για την αξιολόγηση της επίδοσης των δύο αλγορίθμων μηχανικής μάθησης ως προς την προβλεπτική ικανότητά τους χρησιμοποιήθηκε το στατιστικό μέτρο «Mean Absolute Percentage Error (MAPE. Οι αλγόριθμοι FNN και SVM προσπάθησαν να εντοπίσουν τα πλοία σε πολλές χρονικές στιγμές, με βάση τα δεδομένα τους από τα ακουστικά υποθαλάσσια σήματα του πλοίου σε σχέση με την διαδρομή που εκτέλεσε το πλοίο η οποία φαινόταν από την εμβέλεια του GPS του. Οι αλγόριθμοι FNN και SVM λοιπόν, για το πρώτο εύρος συχνοτήτων μετάδοσης θορύβου, είχαν τιμή MAPE ίση με 2,2% και 1,5% για τα δεδομένα δοκιμής της 2ης διαδρομής και είχαν τιμή MAPE ίση με 3,9% και 2,2% για τα δεδομένα δοκιμής της 3ης διαδρομής αντίστοιχα. Όσον αφορά το δεύτερο εύρος συχνοτήτων, οι αλγόριθμοι SVM και FNN είχαν τιμή MAPE ίση με 4,0% και 7,0% για τα δεδομένα δοκιμής της 2ης διαδρομής και είχαν τιμή MAPE ίση με 4,6% και 6,3% για τα δεδομένα δοκιμής της 3ης διαδρομής αντίστοιχα. Το συμπέρασμα ήταν ότι οι δύο μέθοδοι κατηγοριοποίησης που εφαρμόστηκαν, είχαν καλύτερη επίδοση για το πρώτο εύρος συχνοτήτων διάδοσης υποθαλάσσιου θορύβου (53-200 Hz), αφού ο θόρυβος και συνεπώς το σήμα που καταγράφεται από τα πλοία μεταδίδεται κανονικά χωρίς ιδιαίτερες απώλειες.

4.10 10η Μελέτη: Επιτήρηση της ασφάλειας των πλοίων σε περιπτώσεις ακραίων καιρικών φαινομένων με την βοήθεια μοντέλων μηχανικής μάθησης

Οι καιρικές συνθήκες είναι ίσως ο βασικότερος παράγοντας που επηρεάζει την έκβαση των ταξιδιών των πλοίων της εμπορικής ναυτιλίας. Σε αρκετές περιπτώσεις όπου υπάρχουν ακραία καιρικά φαινόμενα όπως τυφώνες, μπορεί να προκληθούν ατυχήματα τα οποία να προκαλέσουν είτε υλικές ζημιές στο πλοίο είτε φυσικά περιβαλλοντικές καταστροφές είτε και

απώλεια ζωών. Ο μόνος τρόπος για να αποφευχθούν τέτοια ατυχήματα λόγω ακραίων καιρικών φαινομένων είναι να αποφύγουν τα πλοία τις περιοχές με άσχημες καιρικές συνθήκες καθώς και την προβλεπόμενη πορεία αυτών των δυσμενών συνθηκών. Φυσικά η πορεία των καιρικών φαινομένων δεν μπορεί να προβλεφθεί με απόλυτη ακρίβεια και συχνά αποκλίνει και εμφανίζεται και σε άλλες περιοχές. Συνεπώς το ρίσκο των ταξιδιών παραμένει υψηλό λόγω της ελλιπούς ή και ανεπαρκούς πληροφορίας για τέτοιες δυνητικές και δυσμενείς συνθήκες. Για τον λόγο αυτόν, μια πιθανολογική και βασισμένη στον κίνδυνο μέθοδο παρακολούθησης της ασφάλειας του πλοίου θα ήταν χρήσιμη για παρέμβαση σε δυνητικά επικίνδυνες καταστάσεις και έγκαιρη ανταπόκριση σε απρόοπτα συμβάντα.

Σε αυτήν την μελέτη των *Andrew Rawson et al* (2021) σκοπός ήταν για την περίοδο των τυφώνων στην περιοχή του Ατλαντικού στις Ηνωμένες πολιτείες να προβλεφθούν πιθανά ατυχήματα. Εκπαιδεύτηκαν διάφοροι αλγόριθμοι κατηγοριοποίησης με την χρήση δεδομένων κίνησης των πλοίων, ιστορικών δεδομένων ατυχημάτων και δεδομένων καιρού, έτσι ώστε να μπορούν να κατηγοριοποιούνται τα πλοία που ταξιδεύουν εκείνη την περίοδο στην συγκεκριμένη περιοχή του Ατλαντικού σε υποψήφια για ατύχημα και όχι για ατύχημα. Χρησιμοποιήθηκαν δύο σύνολα δεδομένων, ένα που περιείχε ιστορικά δεδομένα ατυχημάτων και το οποίο χαρακτηρίστηκε ως η θετική κατηγορία υποψήφιου ατυχήματος και ένα σύνολο δεδομένων που περιείχε δεδομένα κίνησης των πλοίων που θεωρήθηκε ως η αρνητική κατηγορία δηλαδή να μην συμβεί ατύχημα. Στο σύνολο δεδομένων για κάθε περίπτωση υπήρχαν επτά μεταβλητές οι οποίες φαίνεται να επηρέαζαν τους σχετικούς με τον καιρό κινδύνους. Όσον αφορά τα δεδομένα ατυχημάτων τα οποία προήλθαν από το σύστημα «GISIS» του διεθνούς ναυτιλιακού οργανισμού (IMO) για την περίοδο μεταξύ Ιανουαρίου 2005 και Δεκέμβρη 2018 συμπεριλαμβανομένου, υπόψη ελήφθησαν εκείνα τα ατυχήματα τα οποία ήταν αποτέλεσμα ακραίων καιρικών συνθηκών. Συνολικά τέτοιου είδους ατυχήματα ήταν 207 τα οποία και μεταφέρθηκαν σε ένα άλλο, ξεχωριστό σύνολο δεδομένων και αφορούσαν αποκλειστικά φορτηγά πλοία και δεξαμενόπλοια. Από το σύνολο δεδομένων κίνησης των πλοίων, χρησιμοποιήθηκαν δεδομένα από το σύστημα «AIS» ώστε να μοντελοποιηθεί η πλοήγηση των πλοίων εντός της υπό μελέτη περιοχής. Τα δεδομένα εξήχθησαν για τον Αύγουστο, τον Σεπτέμβριο και τον Οκτώβριο για τα έτη 2016 έως 2017 για τις περιοχές μεταξύ Φλόριντας και Βοστώνης. Το τελικό σύνολο δεδομένων κίνησης περιλάμβανε 735.000 θέσεις πλοίων όπου η κάθε μία αντιπροσώπευε, συνολικά, μία ώρα διέλευσης. Το τελικό σύνολο δεδομένων λοιπόν περιείχε 735.000 θέσεις πλοίων και 207 περιπτώσεις ατυχημάτων λόγω δυσμενών καιρικών συνθηκών. Επειδή το σύνολο δεδομένων εκπαίδευσης θεωρήθηκε μη ισορροπημένο χρησιμοποιήθηκε μια μέθοδος εξισορρόπησης του συνόλου δεδομένων. Αυτή ήταν η τεχνική υπερδειγματοληψίας συνθετικής μειονότητας (SMOTE). Η χρήση αυτής της τεχνικής είχε σαν αποτέλεσμα το σύνολο δεδομένων εκπαίδευσης να τροποποιηθεί και έτσι από 587.267 μη ατυχήματα και 173 ατυχήματα να καταλήξει σε 587.440 περιπτώσεις δεδομένων τόσο ατυχημάτων όσο και μη ατυχημάτων. Τα μοντέλα κατηγοριοποίησης που χρησιμοποιήθηκαν στην συγκεκριμένη μελέτη ήταν τα:

- Support Vector Machines (SVM)
- Stochastic gradient descent – SVM (SGD)
- Random Forrest (RF)
- XGBoost (XGB)
- Logistic Regression (LR)
- Multilayer Perceptron (MLP)

Τα μέτρα αξιολόγησης των μοντέλων ήταν τα μέτρα «Accuracy», «Recall», «Precision», «F- Score», «AUC-ROC καμπύλη». Το καλύτερο μοντέλο για κατηγοριοποίηση των πλοίων σε πλοία που είναι υποψήφια για ατύχημα ή όχι, με βάση τα ακραία καιρικά φαινόμενα, φαίνεται πως έδινε ο αλγόριθμος «Stochastic gradient descent - Support Vector Machines» του οποίου οι τιμές των παραπάνω μέτρων αξιολόγησης φαίνονται παρακάτω στον πίνακα 6.

Precision	***Recall***	***Accuracy***	***F-Score***	***AUC-ROC***
0,003	0,95	0,92	0,006	0,98

Πίνακας 6: Μέτρα αξιολόγησης του μοντέλου SGD - SVM

Στα συμπεράσματα της μελέτης όσον αφορά την απόδοση των αλγορίθμων κατηγοριοποίησης τονίστηκε ότι πολλά πλοία στο σύνολο δεδομένων καταγράφηκαν να διέρχονται υπό κακές καιρικές συνθήκες, με σημαντικές ταχύτητες ανέμου, ωστόσο δεν σημειώθηκε ατύχημα. Επιπλέον, αρκετά ατυχήματα συνέβησαν σε συνθήκες λιγότερο σοβαρές από ό,τι θα αναμενόταν. Συνεπώς η πρόβλεψη της σωστής κλάσης από τους αλγορίθμους ήταν αρκετά απαιτητική για αυτό δόθηκε περισσότερο βάση στις τιμές των μέτρων «Accuracy» και «Recall». αυτόν Τελικά, καλύτερος για το συγκεκριμένο πρόβλημα ο αλγόριθμος SGD-SVM συγκριτικά με τον αλγόριθμο SVM, διότι στην ουσία βελτιστοποίησε την ταχύτητα εκτέλεσης του αλγορίθμου SVM δίνοντας παρόμοια αποτελέσματα.

4.11 11η Μελέτη: Μοντέλο συσταδοποίησης καταστάσεων πλοήγησης πλοίων για δοκιμές αποφυγής σύγκρουσης αυτόνομων πλοίων

Οι συγκρούσεις πλοίων είναι το πιο συχνό είδος ατυχημάτων στην θάλασσα καθώς αντιπροσωπεύει περισσότερο από το 50% των θαλάσσιων ατυχημάτων. Αυτές οι συγκρούσεις δεν προκαλούν μόνο υλικές ζημιές στα πλοία αλλά και περιβαλλοντικές καταστροφές ακόμα και απώλεια ζωών. Αποτελεί συνεπώς, μεγάλη πρόκληση για τον ναυτιλιακό τομέα, να μπορούν να αποφεύγονται τέτοιου είδους συγκρούσεις όσο το δυνατόν γίνεται περισσότερο. Για να επιτευχθεί αυτός ο στόχος έχει προταθεί και εφαρμοστεί ένα σύστημα αποφυγής συγκρούσεων μεταξύ πλοίων. Το σύστημα αποφυγής συγκρούσεων μεταξύ πλοίων πρόκειται για ένα από τα κυριότερα συστήματα του MASS που είναι τα αρχικά γράμματα των λέξεων του ονόματος «Maritime Autonomous Surface Ship». Αυτό το σύστημα αποφυγής συγκρούσεων, αναφέρει η εν λόγω μελέτη ότι επικυρώθηκε χρησιμοποιώντας πειράματα που βασίζονται σε σενάρια. Τα σενάρια αυτά όμως δημιουργήθηκαν αυθαίρετα ή βασίζονταν στους διεθνείς κανονισμούς για την πρόληψη συγκρούσεων στην θάλασσα. Η λειτουργία του συστήματος είναι να βρίσκει την τροχιά του κάθε ιδιόκτητου πλοίου και να ελέγχει αν υπάρχουν άλλα πλοία - στόχοι (ή αλλιώς λανθάνοντα πλοία) κοντά ώστε να αποφευχθεί μια πιθανή σύγκρουση. Βέβαια, τα αυτόνομα συστήματα αποφυγής συγκρούσεων βασίζονται στην δυναμική των πλοίων και προϋποθέτουν πως είναι γνωστή εκ των προτέρων, πράγμα που δεν ισχύει σε πολλές και διαφορετικές συνθήκες λειτουργίας των πλοίων. Οπότε αυτό οδηγεί σε σφάλματα του συστήματος αποφυγής συγκρούσεων μεταξύ πραγματικής τροχιάς των σκαφών και προβλεπόμενης τροχιάς τους.

Σκοπός αυτής της μελέτης των *Taewoong Hwang και Ik-Hyun Youn* (2021) ήταν να αναλυθούν διάφορες καταστάσεις πλοήγησης των σκαφών, ομαδοποιώντας την τροχιά του λανθάνοντος πλοίου (πλοίου - στόχου), συλλέγοντας δεδομένα δώδεκα μηνών, και πιο συγκεκριμένα από 1 Σεπτέμβρη 2019 μέχρι 31 Σεπτέμβρη 2020, στην περιοχή της δυτικής θάλασσας της Κορέας από το σύστημα AIS. Τα δεδομένα τροχιάς των ιδιόκτητων πλοίων και των πλοίων στόχων (ή λανθανόντων πλοίων), δηλαδή αυτών με τα οποία τα ιδιόκτητα πλοία ενδέχεται να συγκρουστούν μετατράπηκαν σε μια μεταβλητή που αντιπροσώπευε την κατάσταση πλοήγησης. Στην συνέχεια, εφαρμόστηκε ιεραρχική συσταδοποίηση για την ανάλυση της σύνθεσης και της αναλογίας της κατάστασης πλοήγησης. Για εξαγωγή δεδομένων για τα πλοία - στόχους χρησιμοποιήθηκε το ίδιο χρονικό εύρος με τα δεδομένα των ιδιόκτητων πλοίων. Δηλαδή, χρησιμοποιώντας το εύρος χρονοσειρών για κάθε σκάφος, εξήχθησαν δυναμικά δεδομένα, που αντιστοιχούν στο ίδιο χρονικό εύρος. Όταν η απόσταση μεταξύ των πλοίων ήταν μικρότερη από τρία ναυτικά μίλια τότε εξαγόντουσαν δυναμικά δεδομένα με τον υπολογισμό της απόστασης μεταξύ του ιδιόκτητου πλοίου και του πλοίου στόχου. Η υπολογισμένη απόσταση ήταν η Ευκλείδεια απόσταση χρησιμοποιώντας το γεωγραφικό μήκος και το γεωγραφικό πλάτος. Στην μελέτη χρησιμοποιήθηκε ένας συσσωρευτικός (agglomerative) αλγόριθμος ιεραρχικής συσταδοποίησης για την ανίχνευση των καταστάσεων πλοήγησης των σκαφών. Για τον υπολογισμό της απόστασης μεταξύ των ομάδων χρησιμοποιήθηκε η μέθοδος του μέσου συνδέσμου (Average Linkage), καθώς και η απόσταση του Hamming για την ομοιότητα των συστάδων. Ο συντελεστής Silhouette είχε τιμή ίση με 1 για αριθμό συστάδων ίσο με 327 οπότε τόσες συστάδες δημιουργήθηκαν. Συγκρίνοντας την ομοιότητα και την συχνότητα των συστάδων οι καταστάσεις πλοήγησης που μπορεί να συναντήσει ένα πλοίο κατά τη διάρκεια της πλοήγησής του χωρίστηκαν σε συνηθισμένες και έκτακτες καταστάσεις πλοήγησης με βάση την μέτρηση της ομοιότητας των συστάδων που δημιουργήθηκαν με την μέθοδο Hamming. Οι συνήθεις καταστάσεις πλοήγησης αντιπροσωπεύαν το 95,2% των συνολικών καταστάσεων πλοήγησης ενώ οι έκτακτες καταστάσεις πλοήγησης συνέβησαν σε ποσοστό 4,8%. Τέλος, οι 20 κορυφαίες καταστάσεις πλοήγησης, που αντιπροσωπεύαν το 75% των συνολικών καταστάσεων πλοήγησης, θα μπορούσαν να ταξινομηθούν μαζί με το ποσοστό εμφάνισής τους.

4.12 12η Μελέτη: Χρήση τεχνικών μηχανική μάθησης για την αξιολόγηση της περιβαλλοντικής απόδοσης των πλοίων

Σε αυτήν την μελέτη των *Kyriakos Skarlatos et al* (2023) γίνεται συνδυαστική χρήση των τεχνικών συσταδοποίησης και ανάλυσης σε κύριες συνιστώσες με σκοπό να εντοπιστούν κάποιες ενδείξεις ή διαφορετικά, κάποιοι δείκτες οι οποίοι να αντιπροσωπεύουν και να απεικονίζουν με απλό και κατανοητό τρόπο, την περιβαλλοντική απόδοση των πλοίων. Για να υλοποιηθεί η μελέτη, συλλέχθηκαν δεδομένα τα οποία και συνδυάστηκαν από δύο διαφορετικές πηγές. Η μία πηγή δεδομένων ήταν από τον οργανισμό MRV (EU Monitoring, Reporting, Verification) ο οποίος έχει δεδομένα από αναφορές εκπομπών διοξειδίου του άνθρακα για πλοία άνω των 5000 μικτών τόνων και ανεξαρτήτου σημαίας που δραστηριοποιούνται σε λιμάνια υπό την δικαιοδοσία σε λιμάνια οποιουδήποτε μέλους της Ευρωπαϊκής Ένωσης. Τα άλλα δεδομένα προήλθαν από μια «startup» επιχείρηση στην Ελλάδα με όνομα «27 Research», η οποία παρείχε δεδομένα με τα φυσικά χαρακτηριστικά των πλοίων τα οποία είχαν διεκπεραιώσει τουλάχιστον ένα ταξίδι εντός της Ευρωπαϊκής Ένωσης κατά την περίοδο 2018 με 2021. Τελικά το σύνολο δεδομένων

αποτελούταν από 2650 εγγραφές γενικών φορτηγών πλοίων και άλλες 62 εγγραφές από Κοντέινερ πλοία. Για κάθε πλοίο υπήρχαν συνολικά 14 μεταβλητές που λήφθηκαν υπόψη. Αφού έγινε η κατάλληλη προ επεξεργασία των δεδομένων για εντοπισμό ελλειπουσών τιμών και ακραίων τιμών συνεχίστηκε η μελέτη για συνολικά 2209 πλοία με την χρήση της ανάλυσης σε κύριες συνιστώσες (PCA). Από την χρήση της PCA προέκυψαν ότι σημαντικές ήταν οι τρεις πρώτες κύριες συνιστώσες καθώς εκείνες ήταν που εξηγούσαν το μεγαλύτερο ποσοστό της μεταβλητότητας του αρχικού συνόλου δεδομένων. Στην συνέχεια, οι τρεις πρώτες κύριες συνιστώσες ερμηνεύτηκαν και πιο συγκεκριμένα η πρώτη κύρια συνιστώσα αντιπροσώπευε τις φυσικές διαστάσεις του πλοίου, η δεύτερη κύρια συνιστώσα αντιπροσώπευε την ενεργειακή αποδοτικότητα του πλοίου ενώ η Τρίτη κύρια συνιστώσα απεικόνιζε την περιοχή λειτουργίας του πλοίου. Με βάση τις πρώτες τρεις κύριες συνιστώσες και την αντίστοιχη ερμηνεία τους, έγινε χρήση της τεχνικής της συσταδοποίησης και του αλγορίθμου K-Means συγκεκριμένα, για περαιτέρω ανάλυση του συνόλου δεδομένων. Η χρήση του αλγορίθμου K-Means αποσκοπούσε στην δημιουργία συστάδων από πλοία με παρόμοια χαρακτηριστικά όσον αφορά τις εκπομπές ρύπων, το μέγεθος, την κατανάλωση ενέργειας και άλλα κοινά. Τελικά δημιουργήθηκαν τέσσερις συστάδες όπου για την δημιουργία τους σημαντικό ρόλο έπαιξαν οι πρώτες δύο κύριες συνιστώσες δηλαδή οι φυσικές διαστάσεις και η ενεργειακή αποδοτικότητα των πλοίων, ενώ η τρίτη κύρια συνιστώσα δηλαδή η περιοχή λειτουργίας του πλοίου, είχε λιγότερη επιρροή στην δημιουργία αυτών των τεσσάρων συστάδων. Η πρώτη συστάδα ερμηνεύτηκε ως «μεγάλα και φιλικά προς το περιβάλλον πλοία» ενώ η δεύτερη συστάδα ερμηνεύτηκε ως «μικρά και φιλικά προς το περιβάλλον πλοία». Αντίστοιχα η τρίτη συστάδα ερμηνεύτηκε ως «μεγάλα και μη φιλικά προς το περιβάλλον πλοία» ενώ η τέταρτη συστάδα ερμηνεύτηκε ως «μικρά και μη φιλικά προς το περιβάλλον πλοία». Από την μελέτη βρέθηκε ότι, τα μικρά και φιλικά προς το περιβάλλον πλοία φαίνεται να λειτουργούν αποκλειστικά εντός της δικαιοδοσίας ενός κράτους μέλους κάτι που δεν ισχύει για τις υπόλοιπες κατηγορίες πλοίων. Επιπλέον, εντοπίστηκε σημαντικός αριθμός μικρών πλοίων με κακές περιβαλλοντικές επιδόσεις, τα οποία λειτουργούν αποκλειστικά εκτός της δικαιοδοσίας ενός κράτους μέλους της Ευρωπαϊκής Ένωσης.

ΚΕΦΑΛΑΙΟ 5^{Ο}

5. Εφαρμογές

5.1 Εφαρμογή – Εκπομπή αεριών διοξειδίου του άνθρακα από τα πλοία

Σε αυτό το κομμάτι της διπλωματικής εργασίας, θα προσεγγιστεί, θα αναλυθεί και θα εξεταστεί ένα μείζον πρόβλημα της ναυτιλίας που όπως έχει αναφερθεί και σε προηγούμενα κεφάλαια αλλά και σε μελέτες, είναι η εκπομπή ρύπων από τα πλοία. Στο συγκεκριμένο κομμάτι θα χρησιμοποιηθούν δεδομένα από πλοία τύπου μεταφοράς χύδην φορτίων (Bulk carriers) για τα οποία θα εκτιμηθεί το αν σε βάθος τεσσάρων ετών από ένα έτος αναφοράς, πρόκειται να είναι ρυπογόνα ή όχι. Αρχής γενομένης από το 2023, οι ναυτιλιακές εταιρίες θα είναι υποχρεωμένες να παραθέτουν στοιχεία σχετικά με την ενεργειακή τους απόδοση ώστε να φανεί αν υπάρχει κίνδυνος για εκπομπή πολλών ρύπων στην ατμόσφαιρα με σκοπό την λήψη μέτρων για την αποφυγή τέτοιων ενδεχομένων εκπομπής υπερβολικών αερίων διοξειδίου του άνθρακα. Στην ναυτιλία, μέσω διάφορων μελετών, έχουν βρεθεί, προταθεί και οριστεί δείκτες οι οποίοι μπορούν να εκτιμούν σε βάθος χρόνων αν ένα πλοίο θα εκπέμπει ρύπους ή όχι, ανάλογα με κάποιες παραμέτρους του. Ένας πολύ γνωστός δείκτης, ο οποίος αξιολογεί την ενεργειακή απόδοση των πλοίων και ο οποίος θα χρησιμοποιηθεί στα πλαίσια του συγκεκριμένου προβλήματος είναι ο δείκτης C.I.I. (Carbon Intensity Indicator). Ο δείκτης αυτός, ο οποίος δεν έχει ακόμα τεθεί σε εφαρμογή επισήμως, αξιολογεί το κατά πόσο τα πλοία είναι ενεργειακά αποδοτικά σε βάθος ενός ή και παραπάνω ημερολογιακών ετών ξεκινώντας από το εκάστοτε έτος αναφοράς, κάνοντας εκτιμήσεις για το 2023 και τα μετέπειτα έτη, και τα τοποθετεί σε πέντε κύριες κατηγορίες, τις A,B,C,D και E. Οι κατηγορίες A,B,C δηλώνουν ότι αν ένα πλοίο βρίσκεται σε μια από αυτές τις κατηγορίες όσον αφορά την ενεργειακή απόδοσή του, τότε αυτό σημαίνει ότι δεν ρυπαίνει. Αν όμως εκτιμηθεί ότι ένα πλοίο βρίσκεται στην κατηγορία D για τουλάχιστον τρία χρόνια στην σειρά ή στην E έστω και για μόνο ένα έτος τότε θεωρείται ρυπογόνο και συνεπώς θα πρέπει να ληφθούν διορθωτικά μέτρα για αποφυγή εκπομπών ρύπων στην ατμόσφαιρα όπως τέθηκε αυτός ο στόχος από την Ευρωπαϊκή Ένωση και τον Διεθνή Ναυτιλιακό Οργανισμό (IMO) (DNV).

Αυτή η εκτίμηση για το αν ένα πλοίο όσο περνούν τα έτη υπάρχει η περίπτωση να γίνει ρυπογόνο, βοηθάει τις ναυτιλιακές στο να αλλάξουν στρατηγικό πλάνο με σκοπό να αποφευχθεί αυτή η ρύπανση έτσι ώστε τα πλοία να μειώσουν του ρύπους που εκπέμπουν στην ατμόσφαιρα μέχρι το 2050 στο μισό. Γενικά με το πέρασμα των χρόνων και πλησιάζοντας στο 2030, τα όρια αξιολόγησης της ενεργειακής απόδοσης των πλοίων συνεχώς θα τείνουν να γίνονται και πιο αυστηρά.

Υπολογισμός δείκτη C.I.I

Για τον υπολογισμό του δείκτη C.I.I. υπάρχουν κάποιες κατευθυντήριες γραμμές που έχουν υιοθετηθεί και εφαρμόζονται από το 2019, αναλόγως το είδος πλοίων για τα οποία υπολογίζεται. Στην περίπτωση του συγκεκριμένου προβλήματος, επειδή χρησιμοποιούνται μόνο Bulk Carriers τα βήματα υπολογισμού του δείκτη C.I.I έχουν ως εξής:

- Attained C.I.I. = $\frac{CO2\ Emissions}{Deadweight\ X\ Annual\ DIstance\ Travelled} \bullet 10^6$
- C.I.I. reference year = $a \bullet Capacity^{-C}$

Όσον αφορά τον υπολογισμού του C.I.I. reference year, υπάρχουν ήδη οι τιμές των συντελεστών a και c αναλόγως με την χωρητικότητα σε dwt του κάθε πλοίου. Οι τιμές τους δίνονται στο παρακάτω σχήμα. Επίσης ο όρος «capacity» αναφέρεται στο deadweight των πλοίων.

Ship Type		Capacity	a	c
Bulk Carrier	DWT ≥ 279,000	279,000	4745	0.622
	DWT < 279,000	DWT	4745	0.622

Σχήμα 20: Τιμές των συντελεστών a και c για τον υπολογισμού του δείκτη C.I.I για το έτος αναφοράς (C.I.I. ref)

Πηγή σχήματος: (https://www.classnk.or.jp/)

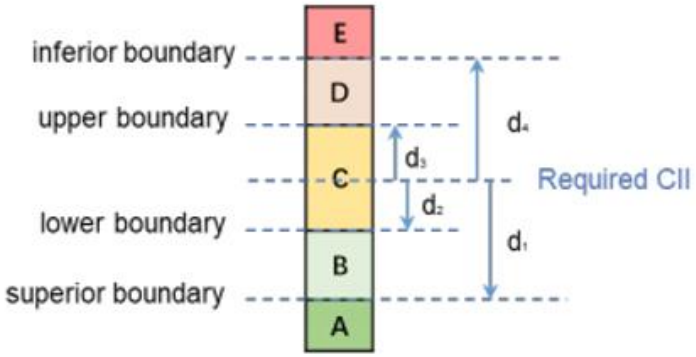
- Required C.I.I. = $\frac{100-Z}{100} \bullet$ C.I.I. ref, όπου Z στην συγκεκριμένη σχέση, είναι η τιμή του συντελεστή μείωσης (reduction factor) του δείκτη C.I.I. για κάθε νέο έτος, ξεκινώντας από το 2023 σε σχέση με το έτος αναφοράς. Παρακάτω, στον πίνακα 7 φαίνονται και οι τιμές του συντελεστή μείωσης.

Έτος	Συντελεστής μείωσης Z
2023	5%
2024	7%
2025	9%
2026	11%

2027	**
2028	**
2029	**
2030	**

Πίνακας 7: Τιμές συντελεστή μείωσης για κάθε έτος από το 2023 και μετέπειτα

Στην συνέχεια χρησιμοποιώντας κάποιους ορισμένους συντελεστές d_1 , d_2 , d_3 , d_4 , προσδιορίζονται τα όρια των βαθμολογιών για τις εκπομπές ρύπων από τα πλοία έτσι ώστε να βρεθεί η κατηγορία στην οποία ανήκουν με το πέρασμα των χρόνων. Τα συγκεκριμένα όρια φαίνονται στο παρακάτω σχήμα 21 ενώ οι αντίστοιχες τιμές των παραπάνω συντελεστών για τα πλοία Bulk Carriers φαίνονται στον πίνακα 8.



Σχήμα 21: : Όρια κατηγοριών για τα πλοία αναλόγως την εκπομπή ρύπων
Πηγή σχήματος: (https://www.classnk.or.jp/)

Είδος Πλοίου	d_1	d_2	d_3	d_4
Bulk Carrier	0,86	0,94	1,06	1,18

Πίνακας 8: Τιμές συντελεστών ορίων για τις εκπομπές ρύπων από τα πλοία

Τέλος, για να βγει το συμπέρασμα σε ποια κατηγορία «ρύπων» ανήκει κάποιο πλοίο υπολογίζεται η σχέση:

$$\frac{Attained\ C.I.I.}{Required\ C.I.I.}$$

Όπου ανάλογα με την τιμή της παραπάνω σχέσης και σε ποιο εύρος τιμών συντελεστών ορίων βρίσκεται αυτή, γίνεται η αξιολόγηση των πλοίων όσον αφορά την κατηγορία που ανήκουν με βάση τους ρύπους που εκπέμπουν.

Περιγραφή συνόλου δεδομένων

Το αρχικό σύνολο δεδομένων αποτελείται από συνολικά 2193 εγγραφές και 23 μεταβλητές. Τα δεδομένα προέρχονται από την επίσημη ιστοσελίδα του MRV (Monitoring Reporting and Verification) σε συνδυασμό με κάποια πλοία τύπου Bulk Carriers από μια ναυτιλιακή εταιρία που αφορούν το έτος 2021 (έτος αναφοράς). Οι μεταβλητές που περιέχονται στο σύνολο δεδομένων παρουσιάζονται στον παρακάτω πίνακα.

Περιγραφή μεταβλητών

Α/Α	**Μεταβλητή**	**Περιγραφή**
1	Reporting Period	Περίοδος αναφοράς
2	Total Fuel consumption [m tonnes]	Συνολική ετήσια κατανάλωση καυσίμων
3	Technical efficiency	Τεχνική αποδοτικότητα
4	Ship type	Τύπος Πλοίου
5	Total CO2 emissions [m tonnes]	Συνολικές ετήσιες εκπομπές ρύπων
6	CO2 emissions from all voyages between ports under a MS jurisdiction [m tonnes]	Εκπομπές CO2 από όλα τα ταξίδια μεταξύ λιμένων υπό τη δικαιοδοσία των κρατών μελών Ε.Ε [εκ. τόνοι]
7	CO2 emissions from all voyages which departed from ports under a MS jurisdiction [m tonnes]	Ετήσιες εκπομπές ρύπων CO2 από όλα τα πλοία που αποχώρησαν από λιμάνια που είναι υπό την δικαιοδοσία των κρατών μελών της Ε.Ε
8	CO2 emissions from all voyages to ports under a MS jurisdiction [m tonnes]	Εκπομπές CO2 από όλα τα ταξίδια σε λιμάνια υπό τη δικαιοδοσία των κρατών μελών [εκ. τόνοι]
9	CO2 emissions which	Εκπομπές CO2 που

	occurred within ports under a MS jurisdiction at berth [m tonnes]	σημειώθηκαν εντός λιμένων υπό τη δικαιοδοσία των κρατών μελών σε αγκυροβόλιο [εκ. τόνοι]
10	Annual Total time spent at sea [hours]	Συνολικός ετήσιος χρόνος που πέρασε ένα πλοίο στην θάλασσα (ώρες)
11	Annual average Fuel consumption per distance [kg / n mile]	Μέση ετήσιακατανάλωση καυσίμων ανά μίλι (kg/ n mile)
12	Annual average Fuel consumption per transport work (mass) [g / m tonnes Â· n miles]	Μέση ετήσια κατανάλωση καυσίμων ανά μεταφορικό έργο
13	Annual average CO2 emissions per distance [kg CO2 / n mile]	Μέση ετήσια εκπομπή ρύπων CO2 ανά απόσταση
14	Annual average CO2 emissions per transport work (mass) [g CO2 / m tonnes Â· n miles]	Μέση ετήσια εκπομπή ρύπων CO2 ανά μεταφορικό έργο
15	Total time spent at sea [hours]	Συνολικός χρόνος που βρισκόταν στην θάλασσα το πλοίο (ώρες)
16	Distance	Απόσταση που διένυσε το πλοίο

17	ves_dwt	Deadweight πλοίου
18	ves_draft	ορίζει την απόσταση μεταξύ της ίσαλης γραμμής και του χαμηλότερου σημείου του κύτους του πλοίου
19	ves_loa	Το μήκος του πλοίου
20	ves_beam	Πλάτος πλοίου
21	ves_depth	Βάθος του πλοίου
22	ves_capacity_grain	Η χωρητικότητα του πλοίου που μετράται πλευρικά προς το εξωτερικό των πλαισίων και κατακόρυφα από τις κορυφές των δεξαμενών έως την κορυφή των δοκών κάτω από το κατάστρωμα
23	ves_main_engine_kw	Η προωθητική δύναμη της κύριας μηχανής του πλοίου σε kw

Όλες οι μεταβλητές του αρχικού συνόλου δεδομένων είναι συνεχείς ποσοτικές με εξαίρεση μόνο τις μεταβλητές «Ship type» η οποία είναι ποιοτική κατηγορική και την μεταβλητή «Reported Period» η οποία είναι ποσοτική διακριτή.

Διερευνητική ανάλυση δεδομένων – Exploratory data analysis

Σκοπός μας όταν εισάγουμε τα δεδομένα μας, είναι να τα δούμε, να τα κατανοήσουμε, δηλαδή η κάθε μεταβλητή σε τι αναφέρεται καθώς να δούμε και τι είδους μεταβλητή είναι η κάθε μία που υπάρχει στο σύνολο δεδομένων όπως και τις αντίστοιχες τιμές της. Πολλές φορές όταν χρησιμοποιούμε ένα σύνολο δεδομένων, υπάρχουν εγγραφές (rows or entries) οι οποίες εμπεριέχουν εσφαλμένες τιμές, ελλιπείς τιμές ή και διπλοεγγραφές (duplicates) σε κάποια χαρακτηριστικά (attributes or columns). Οπότε πρωταρχικός μας στόχος είναι ο καθαρισμός των δεδομένων μας, έτσι ώστε να μπορέσουμε μετά να τα χρησιμοποιήσουμε και να βγάλουμε αξιόπιστα και έγκυρα συμπεράσματα με όσο το δυνατόν λιγότερη μεροληψία (bias). Θέλουμε με άλλα λόγια να βγάλουμε αμερόληπτα αποτελέσματα, οπότε το κομμάτι της προ επεξεργασίας των δεδομένων χρειάζεται ιδιαίτερη προσοχή. Αρχικά όπως είπαμε έχουμε 23 μεταβλητές και 2193 εγγραφές. Οι στήλες που χρειαζόμαστε για την ανάλυση που πρόκειται να κάνουμε σχετικά με τον δείκτη C.I.I. και τι εκτιμήσεις του ξεκινώντας από το 2023 φαίνονται στον παρακάτω πίνακα 9. Επόμενο βήμα, είναι να ελέγξουμε αν έχουμε ελλείπουσες τιμές, διπλοεγγραφές, ή μη λογικές τιμές στα δεδομένα μας. Παρακάτω, στον πίνακα 10, διακρίνουμε πόσες ελλιπείς τιμές υπάρχουν και ποιες μεταβλητές τις έχουν.

A/A	Μεταβλητή
1	Total CO2 emissions [m tonnes]
2	Total Fuel consumption [m tonnes]
3	Distance
4	ves_dwt

Πίνακας 9: Μεταβλητές που χρειάζονται για την εκτίμηση του δείκτη C.I.I.

Μεταβλητή	Σύνολο ελλειπουσών τιμών
Annual average Fuel consumption per distance [kg / n mile]	251
Annual average Fuel consumption per transport work (mass) [g / m tonnes Â· n miles]	273

Annual average CO2 emissions per distance [kg CO2 / n mile]	251
Annual average CO2 emissions per transport work (mass) [g CO2 / m tonnes Â· n miles]	273
Distance	251

Πίνακας 10: Μεταβλητές που έχουν ελλιπείς τιμές και το σύνολο αυτών

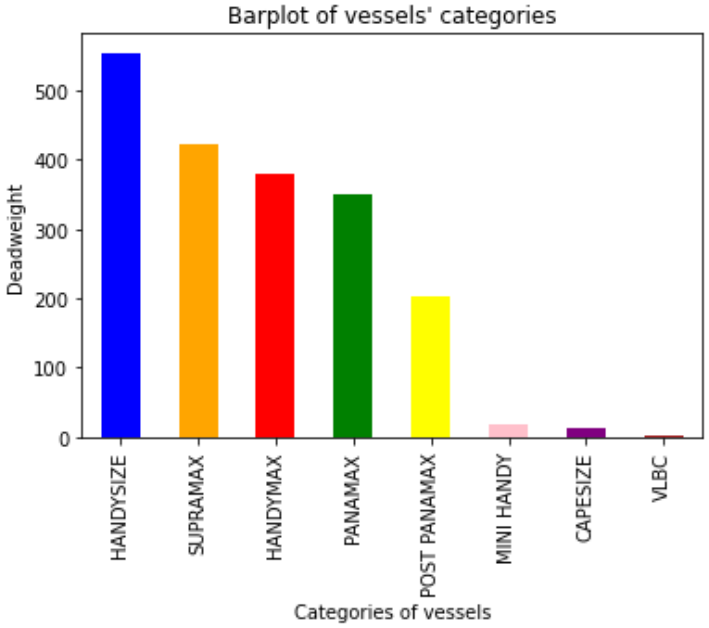
Παρατηρούμε λοιπόν, ότι πέντε μόνο μεταβλητές από τις 23 περιέχουν ελλείπουσες τιμές, οι οποίες είτε θα πρέπει να αφαιρεθούν από το σύνολο δεδομένων είτε με μια διαδικασία που είναι ευρέως γνωστή ως «Imputation» να αντικατασταθούν αυτές οι τιμές με κάποιες άλλες τιμές. Αναλόγως με τα δεδομένα που είναι διαθέσιμα, γίνεται επιλογή για το αν οι εγγραφές με τις ελλείπουσες τιμές πρέπει να απομακρυνθούν ή να αντικατασταθούν με την διαδικασία του «Imputation». Βέβαια, στο συγκεκριμένο κομμάτι της ανάλυσής μας, μας ενδιαφέρει η μεταβλητή «Distance». Επομένως επειδή το σύνολο δεδομένων έχει αρκετές εγγραφές, θεωρήθηκε πως 251 εγγραφές δεν θα επηρεάσουν αρκετά το αποτέλεσμα όσο το να αντικαθιστούσαμε για παράδειγμα με την μέση τιμή των τιμών της συγκεκριμένης στήλης, εφαρμόζοντας δηλαδή την διαδικασία «imputation». Οπότε τελικά οι εγγραφές της συγκεκριμένης στήλης που είχαν ελλείπουσες τιμές, διαγράφηκαν από το σύνολο δεδομένων και έτσι τώρα το σύνολο αυτό αποτελείται από συνολικά 1942 εγγραφές πλοίων. Να σημειωθεί εδώ ότι η απομάκρυνση των ελλιπών τιμών έγινε με βάση την στήλη «Distance» επειδή οι αντίστοιχες ελλείπουσες τιμές των μεταβλητών «Annual average Fuel consumption per distance [kg / n mile]» και «Annual average CO2 emissions per distance [kg CO2 / n mile]» βρισκόντουσαν στην ίδια γραμμή με αυτές της μεταβλητής «Distance». Επιπλέον οι μεταβλητές «Annual average Fuel consumption per transport work (mass) [g / m tonnes Â· n miles]» και «Annual average CO2 emissions per transport work (mass) [g CO2 / m tonnes Â· n miles]» αφαιρέθηκαν από το σύνολο δεδομένων επειδή δεν θέλαμε να υπάρχουν σαν παράμετροι στην ανάλυση. Επόμενο βήμα, είναι να ελεγχθεί αν οι μεταβλητές «Total CO2 emissions», «ves_dwt» και «Distance» έχουν τιμή 0 σε κάποιες εγγραφές. Τυχόν εγγραφές που έχουν τιμή 0 για τις συγκεκριμένες εγγραφές δεν θα πρέπει να ληφθούν υπόψη στην εκτίμηση των ρύπων για τα συγκεκριμένα πλοία μιας και μερικά αποτελέσματα είτε θα τείνουν στο άπειρο είτε θα τείνουν στο 0. Τέτοιες τιμές δεν παρατηρούνται στο σύνολο δεδομένων οπότε η ανάλυση του προβλήματος μπορεί να συνεχιστεί. Το επόμενο πράγμα που πρέπει να εξετάσουμε είναι το αν υπάρχουν «μη λογικές» και ακραίες τιμές στα δεδομένα μας, δηλαδή τιμές εκτός του εύρους τιμών των δεδομένων μας. Σημαντική βοήθεια στον σκοπό μας, προσφέρει η περιγραφική στατιστική και τα περιγραφικά μέτρα. Εξετάζοντας τα κύρια περιγραφικά μέτρα των μεταβλητών μας μπορούμε να δούμε το εύρος τιμών (range), την μέγιστη (Max) και ελάχιστη τιμή (Min), καθώς και μέση τιμή (Mean) και φυσικά τυπική απόκλιση (Standard Deviation) και τεταρτημόρια (Quantiles). Ακολούθως στον πίνακα 11 φαίνονται τα κύρια περιγραφικά μέτρα για τις τέσσερις μεταβλητές «Total CO2 emissions [m tonnes]», «Total Fuel consumption [m tonnes]», «Distance» και «ves_dwt».

Μεταβλητή	Range	Min	Max	Mean	Stand. Deviation	25%	75%	Skewness	Kurtosis
Total CO2 emissions [m tonnes]	20150,8	73,34	20224,1	4028,93	2623,57	2180,74	5275,15	1,55589	4,24452
Total Fuel consumption [m tonnes]	6458,8	23,07	6481,88	1279,39	833,602	693,183	1677,73	1,5677	4,34569
Distance	66820,3	80,9997	66901,3	14975,6	9630,79	8010,51	19940,5	1,31703	2,63095
ves_dwt	226876	6708	233584	50099,8	21900,6	33896,2	61413,5	1,64404	8,19958

Πίνακας 11: Κύρια περιγραφικά μέτρα για τις τέσσερις μεταβλητές «Total CO2 emissions [m tonnes]», «Total Fuel consumption [m tonnes]», «Distance» και «ves_dwt»

Παρατηρούμε ότι για παράδειγμα για την μεταβλητή «Total CO2 emissions [m tonnes]» που δείχνει τον συνολικό αριθμό εκπομπών ρύπων CO2 ανά μίλι τόνου , ότι η ελάχιστη τιμή (min) που παίρνει είναι η τιμή 73,34 και η μέγιστη τιμή (max) της είναι η τιμή 20224,1. Ο μέσος (mean) συνολικός αριθμός εκπομπών ρύπων CO2 ισούται με 4028,93 ανά μίλι τόνου. Το 1ο τεταρτημόριο (25%) συνολικού αριθμού ρύπων CO2 ισούται με 2180,74, ενώ το 3ο τεταρτημόριο (75%) ισούται με 5275,15. Ο συντελεστής ασσυμετρίας (skewness) ισούται με 1,55589 ενώ ο συντελεστής κύρτωσης (kurtosis) ισούται με 4,2445. Συμπεραίνουμε δηλαδή ότι η κατανομή της μεταβλητής «Total CO2 emissions [m tonnes]» παρουσιάζει θετική ασσυμετρία και η κατανομή της χαρακτηρίζεται λεπτόκυρτη.

Σημαντικά συμπεράσματα, όσον αφορά την κατανομή που ακολουθούν τα δεδομένα αλλά και για τον εντοπισμό ακραίων τιμών, μπορούν να εξαχθούν με την βοήθεια των διαγραμμάτων όπως το θηκόγραμμα και το ιστόγραμμα. Επιπλέον για ποιοτικές μεταβλητές μπορούν για οπτική αναπαράσταση των δεδομένων να χρησιμοποιηθεί είτε το γράφημα πίτας (pie chart) είτε το λεγόμενο «ραβδόγραμμα (barplot)». Αν και στο σύνολο δεδομένων μας δεν έχουμε ποιοτικές μεταβλητές προς το παρόν, για την μεταβλητή «ves_dwt» κατασκευάστηκε η μεταβλητή «ves_dwt_category» η οποία δείχνει τις κατηγορίες πλοίων αναλόγως το deadweight τους. Στο παρακάτω σχήμα 22, διακρίνονται οι κατηγορίες πλοίων με βάση το deadweight. Οι κατηγορίες αυτές φτιάχτηκαν με γνώμονα τις κατηγορίες πλοίων όπως παρουσιάστηκαν συνοπτικά στο κεφάλαιο 2.

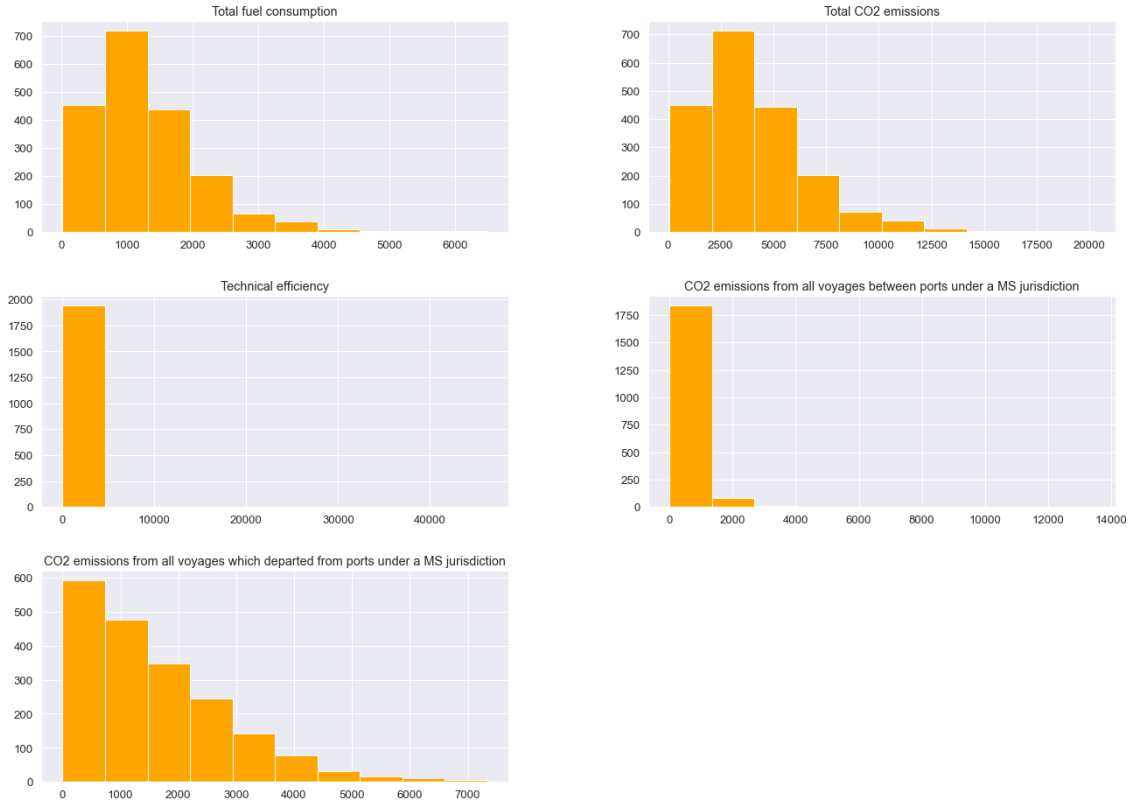


Σχήμα 22: Ραβδόγραμμα με τις κατηγορίες πλοίων με βάση το dwt

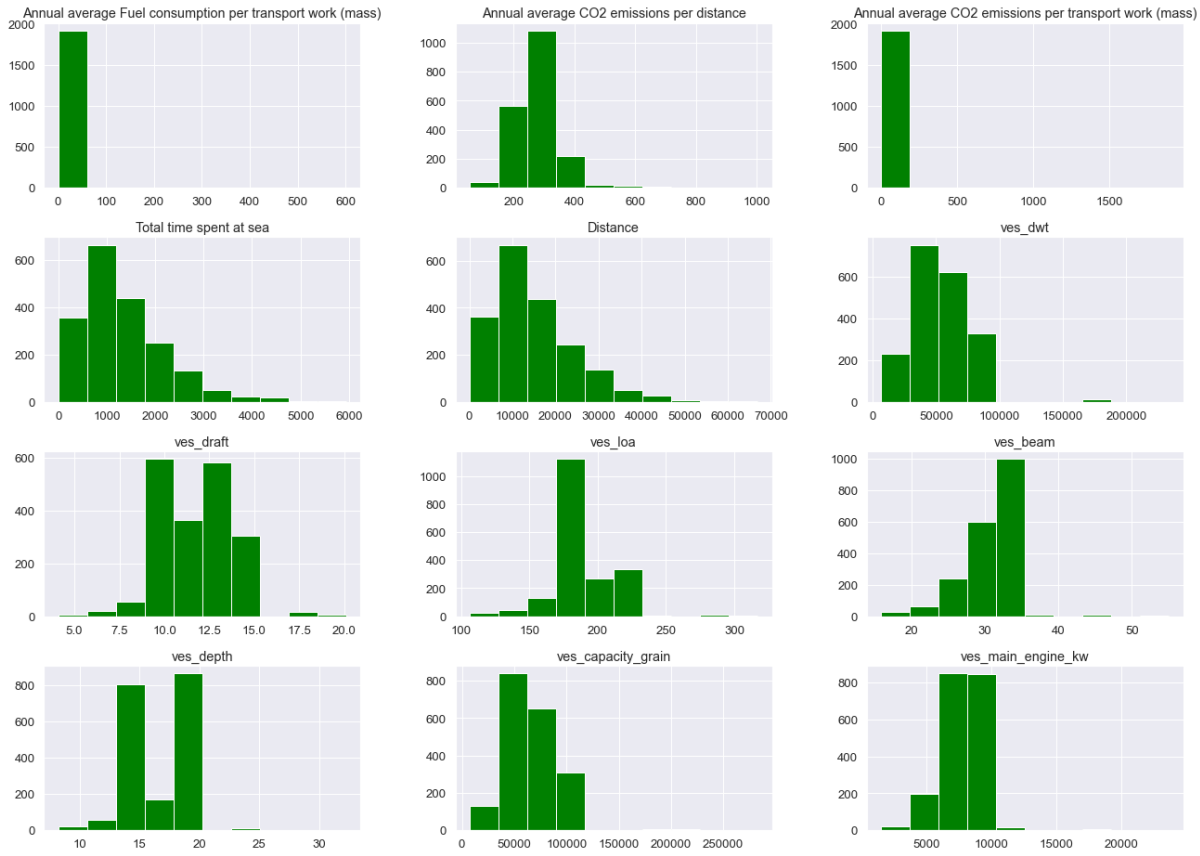
Όπως φαίνεται και στην παραπάνω εικόνα, στο σύνολο δεδομένων, τα περισσότερα πλοία ανήκουν στην κατηγορία «Handysize» ενώ αρκετά ανήκουν στις κατηγορίες «Supramax», «Handymax» και «Panamax» αντίστοιχα. Πιο συγκεκριμένα, 555 πλοία είναι τύπου Handysize δηλαδή το 28,57% του συνόλου δεδομένων ενώ 422 είναι τύπου Supramax δηλαδή το 21,7% και 349 είναι τύπου Panamax, δηλαδή το 17,9%.

Στην συνέχεια της ανάλυσης, γίνεται έλεγχος των ποσοτικών δεδομένων για την κατανομή που ακολουθούν αλλά και για τυχόν ακραίες τιμές. Αρχικά στην εικόνα 5 βλέπουμε το λεγόμενο γράφημα πυκνότητας το οποίο δείχνει την μορφή της κατανομής που ακολουθεί η μεταβλητή «ves_dwt». Όπως στην στατιστική έτσι και στην στατιστική μηχανική μάθηση έχει μεγάλη σημασία το τι κατανομή ακολουθούν οι διάφορες μεταβλητές. Η εύρεση της κατανομής που ακολουθούν τα δεδομένα παίζει ιδιαίτερο ρόλο για το ποιες τεχνικές θα χρησιμοποιηθούν ή για το πως θα μετασχηματιστούν τα δεδομένα για ανήκουν στο ίδιο εύρος τιμών έτσι ώστε οι εκτιμήσεις που γίνονται από τους αλγορίθμους να μην επηρεάζονται από το μεγάλο εύρος τιμών των δεδομένων και για να ακολουθούν μια συγκεκριμένη κατανομή. Επίσης όταν γίνεται κάποιος έλεγχος κάποιας υπόθεσης, το αν τα δεδομένα ακολουθούν κανονική κατανομή ή όχι είναι απαραίτητο να είναι γνωστό για την εφαρμογή κατάλληλων ελέγχων. Από το σχήμα 23 φαίνεται ότι η μεταβλητή «ves_dwt» δεν ακολουθεί κανονική κατανομή. Παρακάτω, στα σχήματα 24, 25 και 26 βλέπουμε τα ιστογράμματα για όλες τις ποσοτικές μεταβλητές του συνόλου δεδομένων.

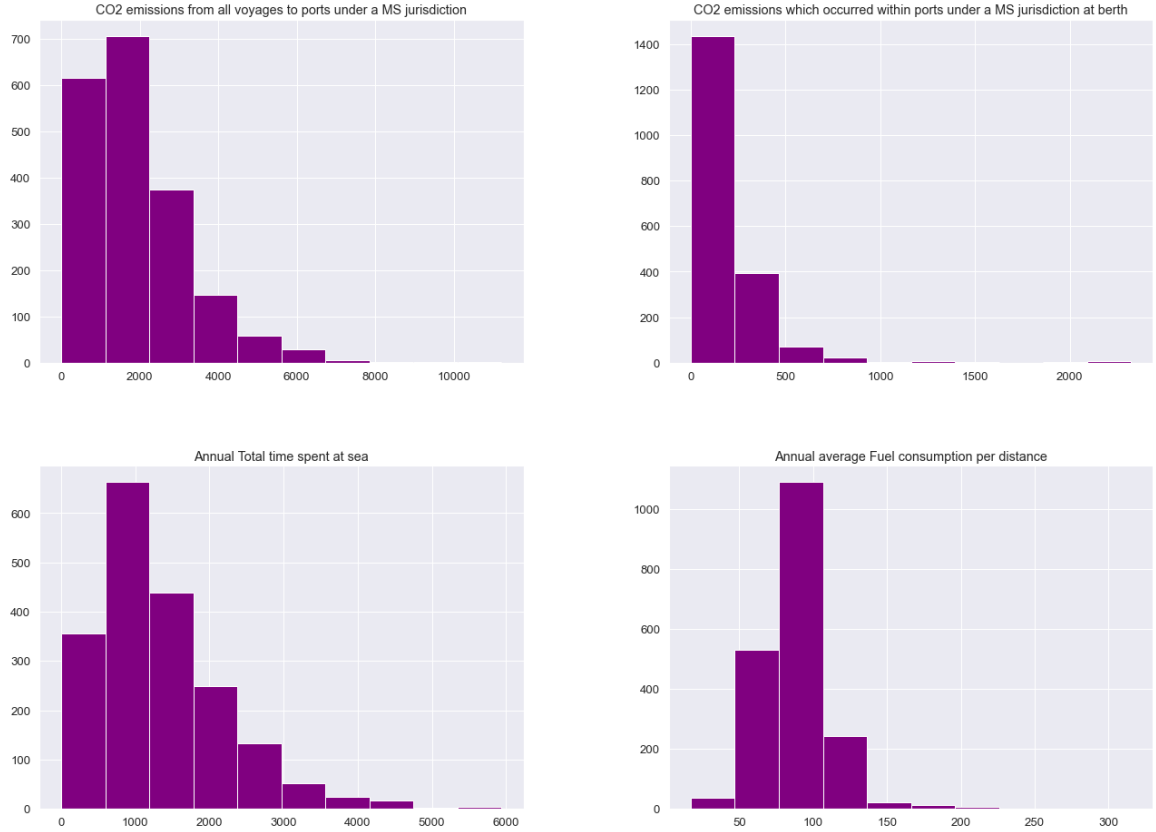
Σχήμα 23: Γράφημα πυκνότητας για την μεταβλητή «ves_dwt»



Σχήμα 24: Ιστογράμματα ποσοτικών μεταβλητών του συνόλου δεδομένων (Ά σύνολο μεταβλητών)

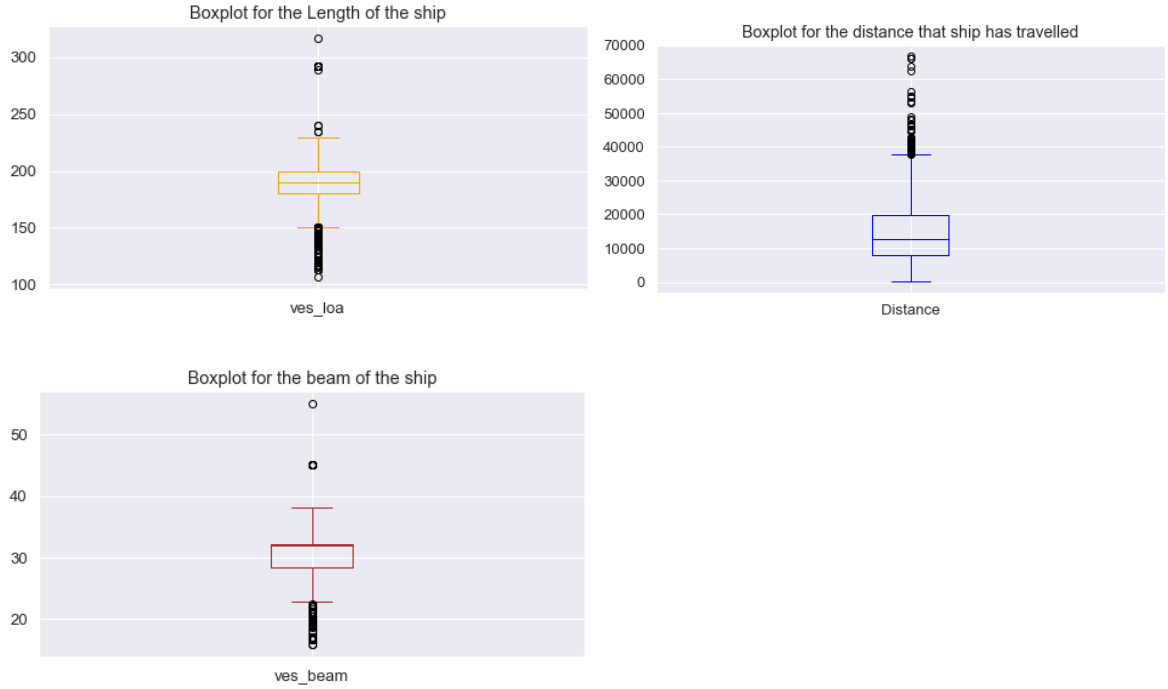


Σχήμα 25: Ιστογράμματα ποσοτικών μεταβλητών του συνόλου δεδομένων (Ά Β σύνολο μεταβλητών)



Σχήμα 26: Ιστογράμματα ποσοτικών μεταβλητών του συνόλου δεδομένων (Γ σύνολο μεταβλητών)

Από τα παραπάνω διαγράμματα φαίνεται τα δεδομένα να μην ακολουθούν την κανονική κατανομή. Για να αποδειχθεί όμως αυτή η απεικόνιση των δεδομένων και ότι δεν ακολουθούν κανονική κατανομή, θα πρέπει να εφαρμοστεί κάποιος έλεγχος κανονικότητας. Στην διεθνή βιβλιογραφία, υπάρχουν αρκετοί έλεγχοι κανονικότητας των δεδομένων όπως ο έλεγχος των «Kolmogorov – Smirnov», ή ο έλεγχος «Shapiro – Wilk» ή ο έλεγχος «Lilliefors», και άλλοι. Εφαρμόζοντας τον έλεγχο Kolmogorov – Smirnov για όλες τις ποσοτικές συνεχείς μεταβλητές του συνόλου διαπιστώθηκε πως καμία δεν ακολουθούσε την κανονική κατανομή αφού η τιμή p-value του ελέγχου σε όλες τις περιπτώσεις ήταν 0 με αποτέλεσμα να απορριφθεί η μηδενική υπόθεση που υποστηρίζει ότι τα δεδομένα ακολουθούν κανονική κατανομή. Επόμενο βήμα είναι να βρεθούν ακραίες παρατηρήσεις στο σύνολο δεδομένων. Ένας τρόπος οπτικού εντοπισμού ακραίων παρατηρήσεων όπως αναφέρθηκε και προηγουμένως είναι τα θηκογράμματα. Στην συνέχεια λοιπόν παρουσιάζονται τα θηκογράμματα μερικών μεταβλητών που εμφανίστηκαν αρκετές ακραίες τιμές. Οι οριζόντιες γραμμές οι οποίες διακρίνονται πέραν των δύο οριζοντίων πλευρών του παραλληλογράμμου σε αποστάσεις ίσες το πολύ με μιάμιση φορά το ενδοτεταρτημοριακό εύρος της κατανομής, δηλαδή 1,5(Q3-Q1), ονομάζονται φράκτες (whiskers). Τιμές της κατανομής που είναι έξω από τις περιοχές που ορίζονται από τους φράκτες θεωρούνται ακραίες παρατηρήσεις ή εναλλακτικά «παράτυπα» σημεία (outliers).

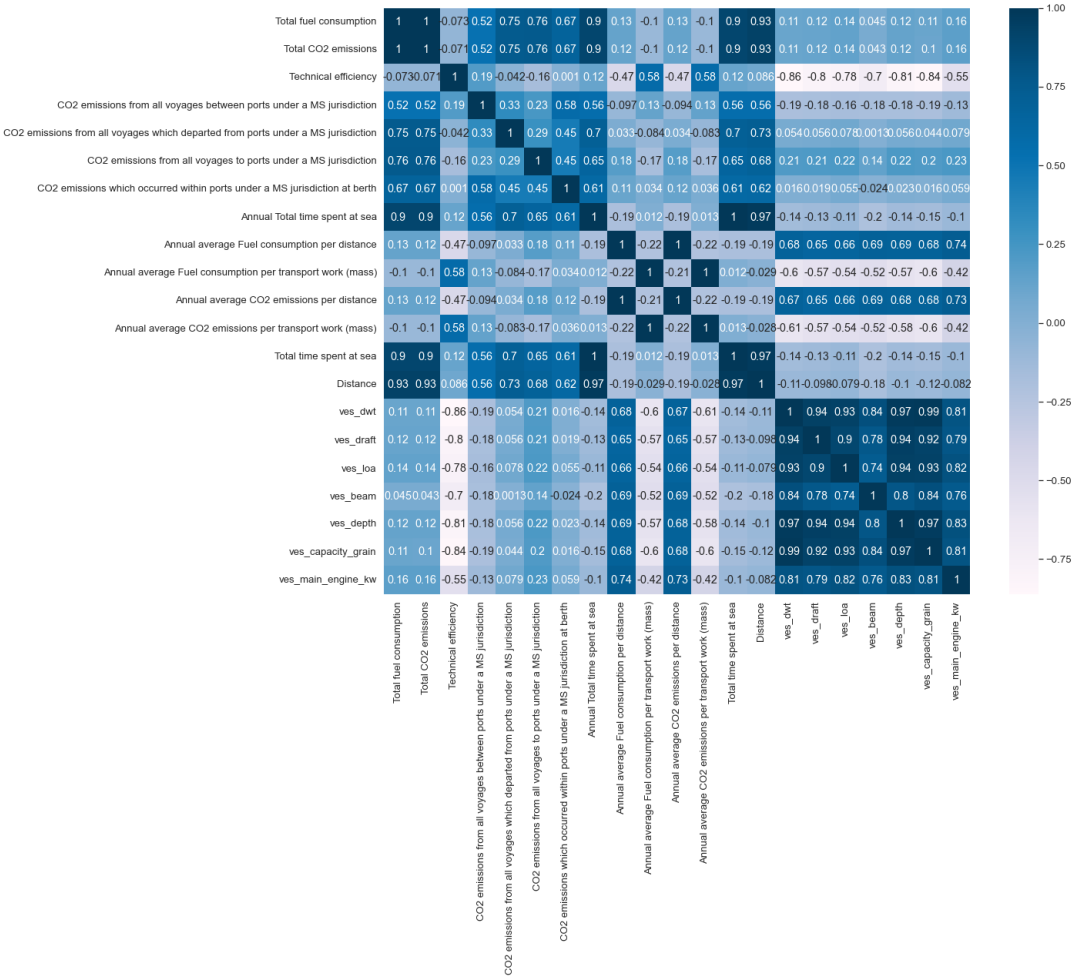


Σχήμα 27: Θηκογράμματα των μεταβλητών «ves_loa», «ves_beam», «Distance»

Γενικά, στο σύνολο δεδομένων υπήρχαν αρκετές μεταβλητές που είχαν πολλές ακραίες τιμές και άλλες μεταβλητές όπως οι «ves_dwt» και «ves_depth» που είχαν λίγες ακραίες τιμές δηλαδή πιο συγκεκριμένα 13 ακραίες παρατηρήσεις η μεταβλητή «ves_dwt» και 16 η μεταβλητή «ves_depth» αντίστοιχα. Για τις μεταβλητές «Distance», «Total fuel consumption» και «Total CO2 emissions» εντοπίστηκαν 54, 66 και 67 ακραίες παρατηρήσεις αντίστοιχα. Επειδή πλέον το σύνολο δεδομένων δεν είναι ιδιαίτερα μεγάλο αφού ήδη έχουν αφαιρεθεί 251 εγγραφές και έχουν μείνει 1942, θεωρήθηκε ότι οι εγγραφές των μεταβλητών με τις ακραίες παρατηρήσεις έπρεπε να παραμείνουν στο σύνολο δεδομένων γιατί η διαγραφή τους, πιθανόν να οδηγούσε σε μεροληπτικά αποτελέσματα αφού θα έλειπε σημαντική πληροφορία από τα δεδομένα.

Στην συνέχεια της ανάλυσης, υπολογίζονται οι συσχετίσεις μεταξύ των μεταβλητών. Υπολογίζεται ο συντελεστής συσχέτισης του Spearman αφού τα δεδομένα δεν ακολουθούν κανονική κατανομή. Ο συντελεστής συσχέτισης του Spearman που συμβολίζεται με r_s παίρνει τιμές στο διάστημα [-1,1]. Η ερμηνεία του, είναι ακριβώς η ίδια με αυτή του συντελεστή συσχέτισης του Pearson και δεν επηρεάζεται από τις μονάδες μέτρησης των μεταβλητών. Συσχετίσεις αρνητικές δείχνουν ότι όσο αυξάνεται ή μειώνεται η τιμή της μιας μεταβλητής τόσο μειώνεται ή αυξάνεται αντίστοιχα η τιμή της άλλης μεταβλητής. Όταν ο συντελεστής συσχέτισης παίρνει τιμές θετικές, αυτό σημαίνει ότι όταν αυξάνεται ή μειώνεται η τιμή της μιας μεταβλητής τότε αυξάνεται ή μειώνεται αντίστοιχα η τιμή της άλλης μεταβλητής. Εάν η τιμή του συντελεστή συσχέτισης ισούται με το -1 ή το +1 τότε οι δύο μεταβλητές εμφανίζουν εντελώς θετική ή εντελώς αρνητική γραμμική σχέση. Τιμή του συντελεστή συσχέτισης ίση με 0, υποδηλώνει ότι οι μεταβλητές είναι γραμμικά ασυσχέτιστες. Σκοπός της εύρεσης συσχετίσεων είναι να εντοπιστούν τυχόν πολύ μεγάλες συσχετίσεις είτε αρνητικές είτε θετικές οι οποίες μπορεί να επηρεάσουν την

διαδικασία μοντελοποίησης. Πολλές φορές, ένα πρόβλημα που παρατηρείται σε αρκετά μοντέλα πρόβλεψης είναι αυτό της πολυσυγγραμικότητας. Πολυσυγγραμικότητα εμφανίζεται στα δεδομένα όταν δύο ή περισσότερες ανεξάρτητες μεταβλητές είναι πολύ υψηλά συσχετισμένες μεταξύ τους. Όταν οι ανεξάρτητες μεταβλητές είναι υψηλά συσχετισμένες τότε το πόσο σημαντική είναι μια μεταβλητή για το μοντέλο είναι δύσκολο να βρεθεί γιατί συγχέονται οι επιδράσεις και επιπλέον δημιουργούνται αμφίβολες σχέσεις. Συνεπώς η πολυσυγγραμικότητα μπορεί να οδηγήσει σε εσφαλμένα συμπεράσματα οπότε για αυτό είναι αναγκαίο να βρεθούν οι



συσχετίσεις.

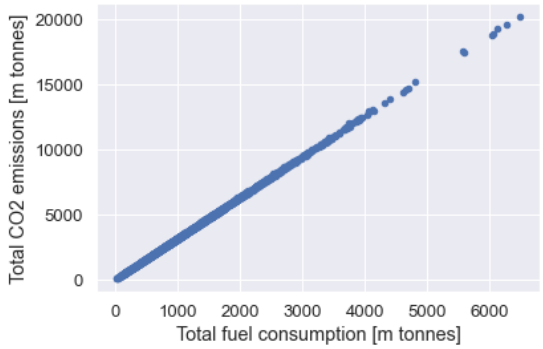
Σχήμα 28: Πίνακας συσχετίσεων Spearman μεταξύ των μεταβλητών

Από το σχήμα 28 φαίνεται ότι υπάρχουν αρκετά ζευγάρια που εμφανίζουν υψηλές συσχετίσεις μεταξύ τους, είτε υψηλά αρνητικές συσχετίσεις. Υψηλές θεωρούνται οι συσχετίσεις που έχουν συντελεστή συσχέτισης του Spearman μεγαλύτερο από 0,7 ή συντελεστή συσχέτισης του Spearman μικρότερο από -0,7. Γενικά παρατηρήθηκε στο σύνολο δεδομένων ότι μεταβλητές που αναφέρονταν σε καταναλώσεις καυσίμου και σε εκπομπές ρύπων είχαν υψηλές συσχετίσεις (> 0.7) μεταξύ τους. Επίσης παρατηρήθηκε σχεδόν τέλεια θετική συσχέτιση μεταξύ των μεταβλητών «Total_CO2_emissions [m tonnes]» και «Total fuel consumption [m tonnes]» με

συντελεστή συσχέτισης spearman που βρέθηκε ίσος με 0,999 (p-value = 0,000) πράγμα φυσιολογικό διότι όταν αυξάνεται η κατανάλωση καυσίμου, αυξάνονται και οι ρύποι που εκπέμπονται από τα πλοία (Σχήμα 29). Άξιο αναφοράς είναι ακόμα ότι υψηλή συσχέτιση εμφανίζεται μεταξύ του συνολικού χρόνου που πέρασε κάποιο πλοίο στην θάλασσα και της κατανάλωσης καυσίμου (r_s > 0,8 και p-value = 0,000) αλλά και της εκπομπής ρύπων (r_s > 0,8 και p-value = 0,000). Παρακάτω, στον πίνακα 12, φαίνονται μερικά από τα ζευγάρια μεταβλητών εκτός από ζευγάρια κατανάλωσης καυσίμου και εκπομπής ρύπων που είχαν στατιστικώς σημαντικά, υψηλές συσχετίσεις.

Μεταβλητή 1	Μεταβλητή 2	Τιμή r_s	p-value
Ves_loa	Ves_beam	0,75	0,000
Ves_draft	Ves_beam	0,764	0,000
Ves_main_engine_kw	Ves_beam	0,775	0,000
Ves_draft	Ves_main_engine_kw	0,789	0,000
Ves_loa	Ves_main_engine_kw	0,815	0,000
Ves_depth	Ves_beam	0,82	0,000
Ves_dwt	Ves_beam	0,83	0,000
Ves_capacity_grain	Ves_beam	0,832	0,000
Ves_loa	Ves_draft	0,855	0,000
Ves_dwt	Ves_main_engine_kw	0,863	0,000
Ves_loa	Ves_depth	0,882	0,000
Ves_dwt	Ves_draft	0,905	0,000
Total_fuel_consumption	Distance	0,911	0,000
Total_CO2_emissions	Distance	0,912	0,000

Πίνακας 12: Συντελεστής συσχέτισης Spearman και p-value για ζευγάρια μεταβλητών



Σχήμα 29: Διάγραμμα διασποράς μεταξύ συνολικής κατανάλωσης καυσίμου και συνολικών εκπομπών ρύπων

Επόμενο στάδιο, είναι να ελεγχθεί το αν τα δεδομένα «πάσχουν» από το φαινόμενο της πολυσυγγραμικότητας, πράγμα που μπορεί να οδηγήσει σε εσφαλμένες εκτιμήσεις και συμπεράσματα όπως αναφέρθηκε και προηγουμένως. Για να εξακριβωθεί το αν υπάρχει το συγκεκριμένο φαινόμενο και για να εντοπιστούν οι «προβληματικές» και υψηλά συσχετισμένες

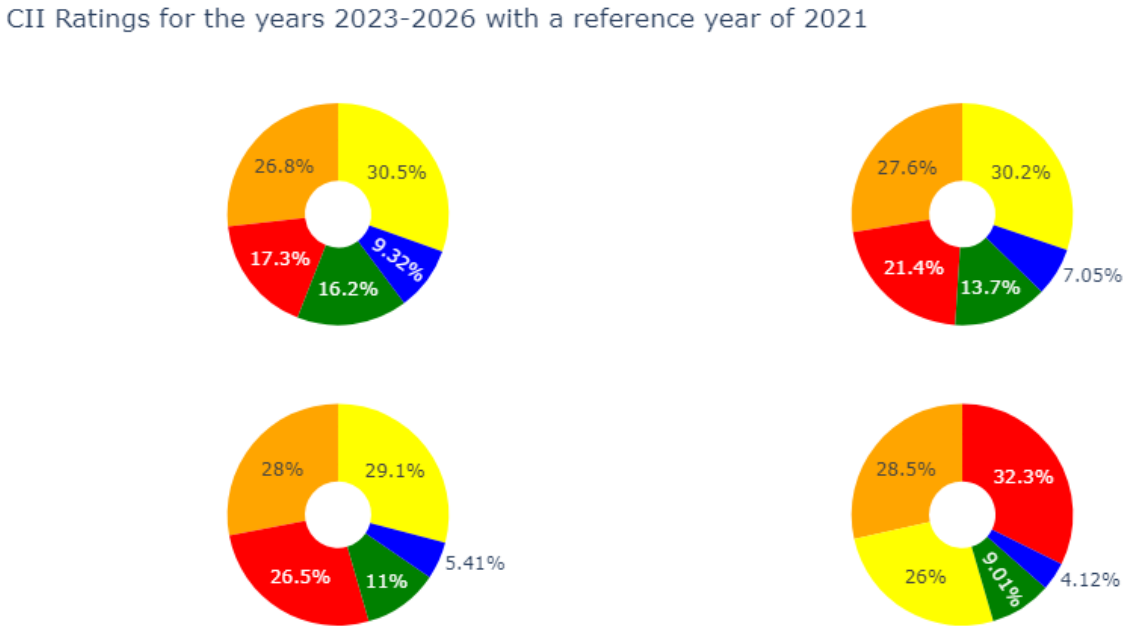
μεταβλητές που το δημιουργούν, χρησιμοποιούνται οι τιμές του γνωστού δείκτη από την διεθνή βιβλιογραφία με όνομα VIF (Variance Inflation Factor) (Εφαρμοσμένη στατιστική και στατιστική μηχανική μάθηση, 2021). Τιμές υψηλές του δείκτη VIF, δηλαδή τιμές μεγαλύτερες του 10 μεταξύ μεταβλητών (An Introduction to Statistical Methods and Data Analysis, 2015), δείχνουν ότι δύο ή παραπάνω μεταβλητές είναι συσχετισμένες μεταξύ τους και οι επιδράσεις τους πιθανών να συγχέονται. Σε μια τέτοια περίπτωση για να επιλυθεί το πρόβλημα της πολυσυγγραμικότητας θα πρέπει μια ή και παραπάνω από τις ανεξάρτητες μεταβλητές να απομακρυνθούν από το μοντέλο πρόβλεψης. Συνήθως, από εκείνες τις υψηλά συσχετισμένες μεταβλητές, όπου κάποια πρέπει να απομακρυνθεί από το μοντέλο πρόβλεψης και η άλλη να παραμείνει, στο μοντέλο παραμένει εκείνη η μεταβλητή που παρέχει την περισσότερη πληροφορία. Εκτελώντας αρκετές δοκιμές μέσω εφαρμογής ενός πολλαπλού γραμμικού μοντέλου παλινδρόμησης για να εκτιμηθούν οι συνολικές εκπομπές ρύπων, και για να γίνει σύγκριση της τιμής του συντελεστή προσδιορισμού R^2, όταν σε κάθε δοκιμή αφαιρούνταν κάποιες από τις υψηλά συσχετισμένες μεταβλητές, διαπιστώθηκε ότι για να μειωθούν οι πολύ υψηλοί δείκτες VIF που εμφανίστηκαν σε πολλές περιπτώσεις μεταβλητών με τιμή πάνω από 100, στο τελικό μοντέλο έπρεπε να παραμείνουν μόνο οι μεταβλητές «Technical efficiency», «CO2 emissions from all voyages to ports under a MS jurisdiction», «CO2 emissions which occurred within ports under a MS jurisdiction at berth», «Distance» και «ves_dwt». Οι τελικές τιμές του δείκτη VIF για αυτό το τελικό σύνολο μεταβλητών φαίνεται στον παρακάτω πίνακα 13. Στην ουσία η μεταβλητή «ves_dwt» αντικατέστησε τις υπόλοιπες παραμέτρους που αφορούσαν τις διαστάσεις του πλοίου με τις οποίες εμφάνιζε υψηλή συσχέτιση. Οι άλλες δύο μεταβλητές σχετικά με τις εκπομπές ρύπων παρέμειναν με χαμηλές τιμές VIF στην θέση άλλων μεταβλητών που είχαν υψηλές τιμές VIF. Τέλος η μεταβλητή «Distance» παρέμεινε στην θέση των μεταβλητών που αφορούσαν καταναλώσεις καυσίμου και εκπομπών ρύπων με τις οποίες εμφάνιζε υψηλότερη συσχέτιση. Η τιμή του συντελεστή προσδιορισμού R^2 όταν χρησιμοποιούνταν όλες οι μεταβλητές του συνόλου δεδομένων πέραν των τεσσάρων μεταβλητών που αφαιρέθηκαν εξαρχής όπως αναφέρθηκε προηγουμένως, ήταν ίση με 1. Φτάνοντας στο τελικό μοντέλο με ανεξάρτητες μεταβλητές αυτές που φαίνονται στον πίνακα 13, η τιμή του συντελεστή προσδιορισμού R^2 βρέθηκε ίση με 0,974 ενώ δεν θεωρήθηκε στατιστικά σημαντική η μεταβλητή «Technical Efficiency» (p-value >0,05) οπότε μπορεί να μην χρησιμοποιηθεί σε περίπτωση που γινόταν ανάλυση για πρόβλεψη των συνολικών εκπομπών διοξειδίου του άνθρακα. Το μοντέλο με τις συγκεκριμένες μεταβλητές εφαρμόζοντας τον έλεγχο ANOVA θεωρήθηκε στατιστικά σημαντικό (p-value < 0 ,000).

Μεταβλητή	VIF
Technical efficiency	1,000
CO2 emissions from all voyages to ports under a MS jurisdiction	5,977
CO2 emissions which occurred within ports under a	2,725

MS jurisdiction at berth	
Distance	6,872
ves_dwt	2,881

Πίνακας 13: Τελικές τιμές δείκτη VIF για τις εναπομείναντες μεταβλητές

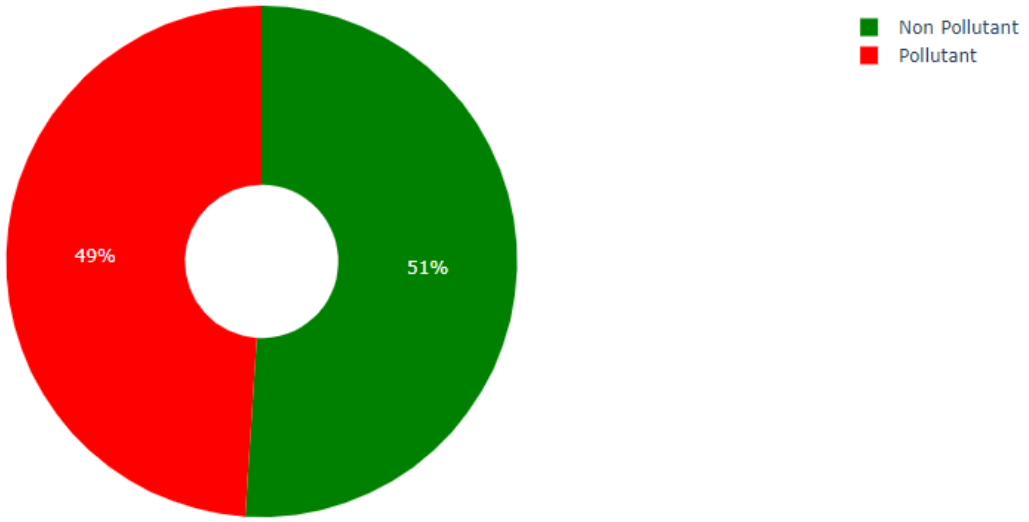
Στην συνέχεια, δημιουργούνται οι καινούργιες μεταβλητές «Attained_CII» και «Required_CII» και «CII_ref» όπως αναφέρθηκε στην αρχή της συγκεκριμένης ενότητας. Έχοντας υπολογίσει πλέον, τις κατηγορίες που ανήκουν τα πλοία ανάλογα με τις τιμές που προκύπτουν από το αποτέλεσμα της διαίρεσης «Attained C.I.I / Required C.I.I.» για κάθε έτος από το 2023 μέχρι και το 2026, εξάγονται κάποια τελικά συμπεράσματα σχετικά με τις εκπομπές ρύπων.



Σχήμα 30: Αξιολόγηση των πλοίων σε κατηγορίες με βάση τις τιμές του δείκτη C.I.I. για τα έτη 2023 – 2026, με έτος αναφοράς, το έτος 2021. Πάνω αριστερά είναι το γράφημα για το έτος 2023, πάνω δεξιά είναι το γράφημα για το έτος 2024, κάτω αριστερά είναι το γράφημα για το 2025 ενώ κάτω δεξιά είναι το γράφημα για το 2026. Με μπλε χρώμα είναι η κατηγορία «A», με πράσινο χρώμα είναι η κατηγορία «B», με κίτρινο χρώμα είναι η κατηγορία «C», με πορτοκαλί χρώμα είναι η κατηγορία «D» ενώ με κόκκινο χρώμα είναι η κατηγορία «E».

Από το παραπάνω σχήμα 30 παρατηρείται ότι όσο αυξάνονται τα έτη, τόσο αυξάνεται το ποσοστό να τείνει ένα πλοίο να γίνεται πιο ρυπογόνο από ότι ήταν το προηγούμενο έτος. Δηλαδή βλέπουμε πως το ποσοστό της κατηγορίας «E» τείνει να αυξάνεται χρόνο με τον χρόνο. Παρατηρείται επίσης ότι τα περισσότερα πλοία, με εξαίρεση το έτος 2026, φαίνεται να έχουν τιμή C.I.I. που τα κατατάσσει στην κατηγορία «C» με το ποσοστό τους περίπου να κυμαίνεται κάθε χρόνο μεταξύ 29% και 31%. Πλοία που το έτος 2023 κατατάσσονταν στην κατηγορία «C» ήταν πολύ πιθανό να γίνουν ρυπογόνα μέσα στα επόμενα τρία έτη αφού θα χρειαζόταν να

καταταχθούν είτε για τις επόμενες τρεις χρονιές στην κατηγορία «D», ή για μια χρονιά από τις επόμενες τρεις, να καταταχθούν στην κατηγορία «E». Από το παρακάτω σχήμα 31 παρατηρείται ότι για το συγκεκριμένο σύνολο δεδομένων για το 2021, το 49% των πλοίων τύπου «Bulk carriers» θα χρειαστεί να αλλάξει στρατηγική έτσι ώστε να μην κινδυνεύουν να γίνουν ρυπογόνα μέσα στα επόμενα χρόνια. Γενικά, ξεκινώντας τις εκτιμήσεις από το 2023 μέχρι και το 2026, φαίνεται να υπάρχει σχεδόν ισορροπία μεταξύ των πλοίων που ενδέχεται να ρυπαίνουν λόγω των εκπομπών αερίων διοξειδίου του άνθρακα και εκείνων που δεν θα ρυπαίνουν αν ακολουθήσουν την ίδια στρατηγική και πλάνο ναυσιπλοΐας.



Σχήμα 31: Γράφημα για ποσοστό πλοίων που θεωρούνται ρυπογόνα με βάση τις τιμές C.I.I. και πλοίων που θεωρούνται μη ρυπογόνα

Ένα ενδιαφέρον ερώτημα που τίθεται σε αυτό το μέρος είναι το αν μπορεί να εξακριβωθεί αν διαφέρουν οι μέσες συνολικές εκπομπές ρύπων στις δύο κατηγορίες, δηλαδή σε αυτή των μη ρυπογόνων πλοίων και σε αυτή των ρυπογόνων. Για να ελεγχθεί το αν διαφέρουν οι μέσες συνολικές εκπομπές διοξειδίου του άνθρακα στις δύο κατηγορίες, χρησιμοποιήθηκε ο μη παραμετρικός έλεγχος για δύο ανεξάρτητα δείγματα «Mann – Whitney» αφού εφαρμόζοντας τον έλεγχο «Shapiro – Wilk» για κανονικότητα των δεδομένων στους δύο πληθυσμούς, διαπιστώθηκε πως η υπόθεση ότι τα δεδομένα ακολουθούν κανονική κατανομή απορρίπτεται σε επίπεδο σημαντικότητας 5% με p-value ίσο με 0. Επειδή το p-value του ελέγχου «Mann – Whitney» βρέθηκε ίσο με 0,001 το συμπέρασμα σε επίπεδο σημαντικότητας 5% ήταν ότι οι μέσες συνολικές εκπομπές διοξειδίου του άνθρακα διαφέρουν στις κατηγορίες ρυπογόνων και μη ρυπογόνων πλοίων.

5.2 Σύγκριση μοντέλων Κατηγοριοποίησης για ταξινόμηση των πλοίων σε ρυπογόνα και μη ρυπογόνα με βάση τις μεταβλητές του συνόλου δεδομένων

Σε αυτό το δεύτερο κομμάτι του προβλήματος, θα προσαρμοστούν και θα συγκριθούν διάφορα μοντέλα κατηγοριοποίησης των δεδομένων στις κατηγορίες ρυπογόνων και μη ρυπογόνων πλοίων με βάση διάφορα χαρακτηριστικά του συνόλου δεδομένων. Μια τέτοια ανάλυση μπορεί να εξάγει χρήσιμα συμπεράσματα σχετικά με το ποιοι αλγόριθμοι κατηγοριοποίησης είναι σε θέση να προβλέπουν ικανοποιητικά και με αρκετά μεγάλη ακρίβεια το αν τα πλοία τύπου «Bulk carriers» μιας ναυτιλιακής εταιρίας ενδέχεται να είναι ρυπογόνα ή όχι μέχρι το 2026, όπως έχουν εκτιμηθεί αυτές οι δύο κατηγορίες με γνώμονα τον δείκτη C.I.I, με βάση κάποιες παραμέτρους όπως οι εκπομπές διοξειδίου του άνθρακα ή η συνολική κατανάλωση καυσίμου από τα πλοία, το μήκος του πλοίου, η συνολική ετήσια απόσταση που έχει διανύσει το πλοίο κ.ά. Με νέα δεδομένα να ρέουν προς τις ναυτιλιακές καθημερινά σχετικά με τις καταναλώσεις των πλοίων, τις εκπομπές αερίων διοξειδίου του άνθρακα και δεδομένα σχετικά με τεχνικά χαρακτηριστικά των πλοίων, αυτή η ανάλυση μπορεί να βοηθήσει τις ναυτιλιακές στο να αναγνωρίζουν άμεσα αν ένα νέο πλοίο του είδους «Bulk Carrier» ενδέχεται μέσα σε διάστημα κάποιων ετών να είναι ρυπογόνο ή όχι, σύμφωνα και με τις εκτιμήσεις του δείκτη C.I.I, έτσι ώστε αν χρειαστεί, να μπορεί άμεσα η ναυτιλιακή εταιρία να αλλάξει στρατηγικό πλάνο ως προς την εκμετάλλευση του πλοίου για να μην εκπέμπει επικίνδυνα πολλούς ρύπους στην ατμόσφαιρα. Η εξαρτημένη μεταβλητή που χρησιμοποιείται για την συγκεκριμένη ανάλυση είναι αυτή που βρέθηκε στο ερώτημα α), δηλαδή η νέα δίτιμη μεταβλητή «ves_pollution» που δημιουργήθηκε στο προηγούμενο ερώτημα. Τιμή 0 της μεταβλητής αυτής δηλώνει «μη ρυπογόνο» πλοίο ενώ η τιμή 1 δηλώνει ρυπογόνο πλοίο. Ως ανεξάρτητες μεταβλητές χρησιμοποιήθηκαν αυτές που υπήρχαν στο αρχικό σύνολο δεδομένων με εξαίρεση τις μεταβλητές «Ship type», «Reporting period», «Annual average Fuel consumption per transport work (mass) [g / m tonnes Â· n miles]» και «Annual average CO2 emissions per transport work (mass) [g CO2 / m tonnes Â· n miles]» οι οποίες δεν λήφθηκαν υπόψιν. Στα πλαίσια της συγκεκριμένης ανάλυσης χρησιμοποιήθηκαν 4 στατιστικοί δείκτες (Accuracy, Precision, Recall, F - Score) για την αξιολόγηση των μοντέλων κατηγοριοποίησης. Οι αλγόριθμοι κατηγοριοποίησης που χρησιμοποιήθηκαν ήταν οι ακόλουθοι 6:

- Gaussian Naive Bayes
- Logistic Regression
- Support Vector Machines
- Random Forest
- Decision Trees
- K Nearest Neighbors

Προεπεξεργασία των δεδομένων (Data Preprocessing)

Για να βρεθεί ένα μοντέλο που να μπορεί με «μεγάλη» επιτυχία να κατηγοριοποιεί σωστά τα δεδομένα σε κατηγορίες, θα πρέπει το σύνολο δεδομένων να «καθαριστεί», ενδεχομένως να μειωθεί και να περιέχει λιγότερες μεταβλητές από το αρχικό σύνολο και τέλος να μετασχηματιστεί, όπως αναφέρθηκε και στο προηγούμενο κομμάτι της ανάλυσης έτσι ώστε τα μοντέλα πρόβλεψης να έχουν βελτιωμένη απόδοση. Σημαντικό λοιπόν κομμάτι, είναι να βρεθούν εκείνες οι ανεξάρτητες μεταβλητές που έχουν την σπουδαιότερη επιρροή στην εύρεση της σωστής κατηγορίας που ανήκουν τα δεδομένα. Αυτό γίνεται διότι τα δεδομένα που μπορούν να

έχουν στην διάθεσή τους οι ναυτιλιακές και γενικά όλοι οργανισμοί έχουν κάποιο κόστος. Μια ναυτιλιακή επιχείρηση μπορεί να μην είναι σε θέση να κατέχει πολλά δεδομένα και παραμέτρους και με βάση το κεφάλαιο που διαθέτει να μην μπορεί να εξασφαλίσει ένα πλούσιο σύνολο δεδομένων και μεταβλητών. Συνεπώς θα πρέπει να αρκεστεί σε ένα μικρότερο σύνολο παραμέτρων και με βάση αυτό το σύνολο να εξάγει συμπεράσματα. Υπάρχουν διάφορες τεχνικές και αλγόριθμοι που μπορούν να επιτύχουν αυτήν την επιλογή χαρακτηριστικών όπως η μέθοδος «Lasso», «Select K best», η μέθοδος με τους συντελεστές συσχέτισης κ.ά. Εφαρμόζοντας 5 διαφορετικές μεθόδους επιλογής χαρακτηριστικών, συνδυάζοντας τα αποτελέσματά τους, επιλέχθηκαν οι 11 καλύτερες μεταβλητές για την πρόβλεψη του πότε ένα πλοίο θα θεωρείται ρυπογόνο ή μη ρυπογόνο. Οι πέντε μέθοδοι επιλογής χαρακτηριστικών που χρησιμοποιήθηκαν ήταν οι «Extra Trees Classifer», «Select K best», «Lasso Regression», «Logistic Regression με Lasso κανονικοποίηση» και «Random Forest Classifier». Να σημειωθεί εδώ ότι πολλές μεταβλητές όπως αυτές με τα φυσικά χαρακτηριστικά ενός πλοίου, είναι υψηλά συσχετισμένες μεταξύ τους οπότε θα έπρεπε κάποιες να απομακρυνθούν. Όμως οι περισσότερες ναυτιλιακές και πιο συγκεκριμένα τα δεδομένα που ρέουν προς αυτές, αφορούν τέτοια χαρακτηριστικά όπως το νεκρό βάρος (deadweight) του πλοίου, το βύθισμα, το μήκος τους και φυσικά τις μέσες ή ετήσιες εκπομπές ρύπων τους και τις μέσες ή ετήσιες καταναλώσεις καυσίμου. Συνεπώς θεωρήθηκε αναγκαίο να διατηρηθούν στο σύνολο δεδομένων και να χρησιμοποιηθούν στα μοντέλα πρόβλεψης. Στον παρακάτω πίνακα 14 παρουσιάζονται οι μεταβλητές του συνόλου δεδομένων και δίπλα το πόσες φορές προέκυψαν συνολικά, εφαρμόζοντας τις πέντε τεχνικές επιλογής χαρακτηριστικών. Η μεταβλητή «Annual average CO2 emissions» προτιμήθηκε αντί της «Total CO2 emissions» επειδή προέκυψε περισσότερες φορές. Με ανοικτό πράσινο χρώμα, είναι σημειωμένες οι μεταβλητές που χρησιμοποιήθηκαν για την κατηγοριοποίηση.

Μεταβλητές	Φορές που προέκυψε
Annual average CO2 emissions per distance	5
Annual average Fuel Consumption per distance	3
CO2 emissions from all voyages between MS ports	3
Distance	4
ves_dwt	5
ves_draft	3
ves_depth	3
ves_loa	5
ves_main_engine_kw	5
ves_capacity_grain	3
Technical Efficiency	3

Total Fuel consumption	1
Total CO2 emissions	3
Total Time spent at sea	0
CO2 emissions from all voyages departed MS port	0
CO2 emissions from all voyages to MS ports	1
Ves_beam	2
Annual Total time spent at sea	0
CO2 emissions within MS port at berth	2

Πίνακας 14: Μεταβλητές που προέκυψαν για τα μοντέλα κατηγοριοποίησης μετά από επιλογή χαρακτηριστικών με πέντε διαφορετικές μεθόδους

Έχοντας ήδη αφαιρέσει τις ελλείπουσες τιμές από τα δεδομένα στο προηγούμενο κομμάτι της ανάλυσης του συνόλου δεδομένων, επόμενο βήμα είναι να μετασχηματιστούν τα δεδομένα έτσι ώστε να μην επηρεάζονται τα μοντέλα μηχανικής μάθησης από το εύρος τιμών των δεδομένων. Για να γίνει αυτό, δύο μέθοδοι χρησιμοποιούνται ευρέως. Είτε η μέθοδος τυποποίησης των δεδομένων «Standard Scaling» και η μέθοδος κανονικοποίησης των δεδομένων «Normalization». Το ποια μέθοδος χρησιμοποιείται για την κλιμάκωση των δεδομένων, εξαρτάται από την φύση του προβλήματος.

Η μέθοδος μετασχηματισμού των δεδομένων «Normalization» χρησιμοποιείται όταν η κατανομή των δεδομένων δεν ακολουθεί κανονική κατανομή. Η συγκεκριμένη μέθοδος «scaling (κλιμάκωσης)» των δεδομένων παίρνει τιμές μεταξύ 0 και 1 και μπορεί να εφαρμοστεί και σε ποιοτικές μεταβλητές, πέρα από ποσοτικές. Αλλιώς ονομάζεται και «Min – Max» scaling. Επίσης τα αποτελέσματα του μετασχηματισμού των δεδομένων με την συγκεκριμένη μέθοδο, επηρεάζονται από πιθανά outliers που υπάρχουν στις μεταβλητές οπότε αυτό είναι ένα μειονέκτημά της. Δηλαδή για να την εφαρμόσουμε καλό θα ήταν πρώτα να διώξουμε τα outliers.

Αντιθέτως, η μέθοδος μετασχηματισμού των δεδομένων «Standardization», χρησιμοποιείται όταν η κατανομή των δεδομένων ακολουθεί κανονική κατανομή. Σε αυτή την περίπτωση η μέθοδος δεν επηρεάζεται από πιθανή ύπαρξη outliers που υπάρχουν στα δεδομένα. Σε αντίθεση με την μέθοδο «Normalization», η μέθοδος τυποποίησης μετατρέπει το εύρος τιμών των μεταβλητών να είναι μεταξύ -1 και 1 και εφαρμόζεται μόνο σε ποσοτικές μεταβλητές. Στο συγκεκριμένο σύνολο δεδομένων αν και διαπιστώθηκε ότι τα δεδομένα δεν ακολουθούν κανονική κατανομή, τα πολλά outliers που εντοπίστηκαν στο σύνολο των ανεξάρτητων μεταβλητών, έκριναν χρησιμότερη την κλιμάκωση των δεδομένων μέσω της μεθόδου «Standard Scaling», δηλαδή σύμφωνα με την σχέση:

$$Z_i = \frac{Xi - \mu_i}{\sigma_i},$$

όπου μ_i είναι η μέση τιμή της i – οστής μεταβλητής και σ_i είναι η τυπική απόκλιση της i – οστής μεταβλητής.

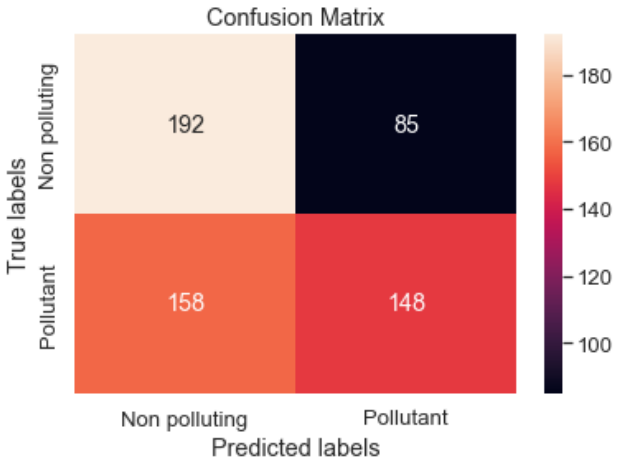
Αποτελέσματα κατηγοριοποίησης

Παρακάτω, στον πίνακα 15 φαίνεται η αποδοτικότητα των μεθόδων κατηγοριοποίησης που εφαρμόστηκαν με τις αντίστοιχες τιμές των μέτρων αξιολόγησής τους. Για την εκπαίδευση των αλγορίθμων κατηγοριοποίησης μηχανικής μάθησης χρησιμοποιήθηκε το 70% των δεδομένων για το σύνολο δεδομένων εκπαίδευσης και το 30% για το σύνολο δεδομένων ελέγχου.

Μοντέλο	Accuracy	Precision	Recall	F - Score	Χρόνος εκτέλεσης αλγορίθμου
Gaussian Naive Bayes	70,66%	74,71%	69,57%	68,64%	0,1709 sec
Logistic Regression	97,25%	97,22%	97,3%	97,25%	0,1898 sec
Support Vector Machines	97,42%	97,39%	97,46%	97,42%	0,1928 sec
Random Forest	97,25%	97,22%	97,31%	97,25%	0,4437 sec
Decision Trees	96,56%	96,53%	96,62%	96,56%	0,1459 sec
K Nearest Neighbors	86,96%	88,38%	87,46%	86,92%	0,1708 sec

Πίνακας 15: Αξιολόγηση μοντέλων κατηγοριοποίησης που εφαρμόστηκαν για το σύνολο δεδομένων που προέκυψε από τις μεθόδους επιλογής χαρακτηριστικών

Από τον πίνακα 15 φαίνεται ότι χρησιμοποιώντας για την εκπαίδευση των μοντέλων, τις μεταβλητές που προέκυψαν από την εφαρμογή των μεθόδων επιλογής χαρακτηριστικών, με εξαίρεση το μοντέλο κατηγοριοποίησης «Gaussian Naive Bayes» τα υπόλοιπα μοντέλα κατηγοριοποίησης πετυχαίνουν περίπου την ίδια υψηλή ακρίβεια με ποσοστό 97% περίπου. Ελαφρώς μικρότερη ακρίβεια πέτυχε το μοντέλο κατηγοριοποίησης «Decision Trees» με ποσοστό ακρίβειας 96,56% ενώ το ποσοστό ανάκλησης ήταν 96,53%. Τον υψηλότερο χρόνο εκτέλεσης χρειάζεται το μοντέλο «Random Forest» αφού για να εκπαιδευτεί και να κάνει τις προβλέψεις χρειάστηκε 0,4437 δευτερόλεπτα. Τον λιγότερο χρόνο χρειάζεται το μοντέλο των δέντρων αποφάσεων με μόλις 0,1459 δευτερόλεπτα για να εκτελεστεί.. Για το μοντέλο «Support Vector Machines» που φαίνεται να έχει την υψηλότερη ακρίβεια, εξειδικευμένη ακρίβεια και ανάκληση, ο αντίστοιχος πίνακας σύγχυσης φαίνεται στο παρακάτω σχήμα 32.



Σχήμα 32: Confusion matrix του μοντέλου κατηγοριοποίησης «Support Vector Machines»

Από το παραπάνω σχήμα το συμπέρασμα ήταν ο συγκεκριμένος αλγόριθμος μπορεί να προβλέπει ότι 192 πλοία δεν πρόκειται να είναι ρυπογόνα και όντως στην πραγματικότητα δεν είναι ρυπογόνα. Επίσης μπορεί και προβλέπει πως 148 πλοία θα είναι ρυπογόνα και στην πραγματικότητα όντως είναι ρυπογόνα. Επιπλέον, η τιμή 85 στον πίνακα σύγχυσης δείχνει ο κατηγοριοποιητής προβλέπει πως 85 πλοία θεωρούνται ρυπογόνα ενώ στην πραγματικότητα δεν θα είναι ρυπογόνα. Αν αντί για το σύνολο δεδομένων που χρησιμοποιήθηκε για την εκπαίδευση των μοντέλων κατηγοριοποίησης, χρησιμοποιούταν το σύνολο μεταβλητών που προέκυψε από τον πίνακα συσχετίσεων σε συνδυασμό με την απαλοιφή της πολυσυγγραμικότητας, δηλαδή από το σύνολο μεταβλητών που φαίνεται στον πίνακα 13, τα αποτελέσματα και η αξιολόγηση της αποδοτικότητας των αλγορίθμων κατηγοριοποίησης διακρίνονται στον παρακάτω πίνακα 16.

Μοντέλο	Accuracy	Precision	Recall	F - Score
Gaussian Naive Bayes	54,02%	57,25%	52,11%	42,83%
Logistic Regression	57,97%	58,83%	58,49%	57,73%
Support Vector Machines	58,31%	59,18%	58,84%	58,08%
Random Forest	72,04%	72,28%	72,25%	72,03%
Decision Trees	66,72%	66,90%	66,89%	66,72%
K Nearest Neighbors	59,17%	62,57%	60,27%	57,64%

Πίνακας 16: Αξιολόγηση μοντέλων κατηγοριοποίησης που εφαρμόστηκαν για το σύνολο δεδομένων που προέκυψε από τον πίνακα συσχετίσεων και τις τιμές VIF των μεταβλητών

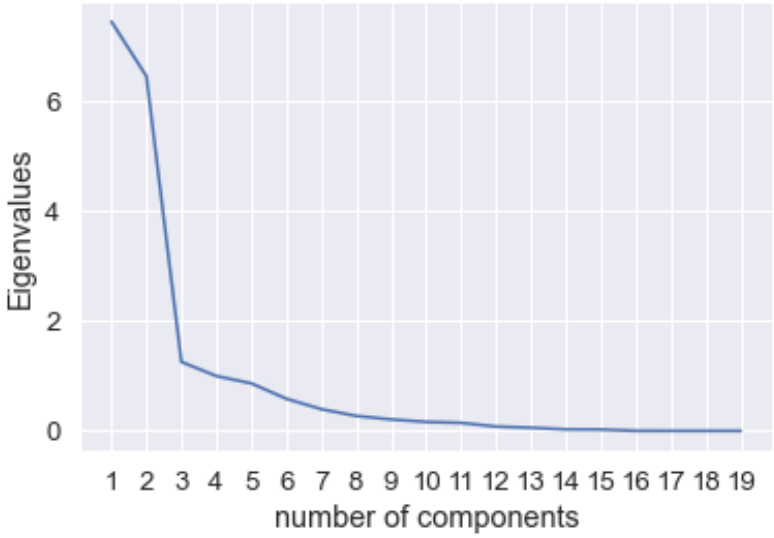
Στο συγκεκριμένο σύνολο μεταβλητών παρατηρούνται διαφοροποιήσεις στην απόδοση των μοντέλων κατηγοριοποίησης. Ο καλύτερος αλγόριθμος κατηγοριοποίησης φαίνεται να είναι ο αλγόριθμος «Random Forest» με ποσοστό ακρίβειας ίσο με 72,04% και με F-Score ίσο με 72,03%. Ο κατηγοροποιητής που έχει την χειρότερη προσαρμογή στα συγκεκριμένα δεδομένα φαίνεται να είναι ο κατηγοριοποιητής «Gaussian Naive Bayes». Αν και οι μεταβλητές του

συγκεκριμένου συνόλου δεδομένων είναι λίγες, δεν πάσχουν από το φαινόμενο της πολυσυγγραμικότητας οπότε η αξιολόγηση των μοντέλων μηχανικής μάθησης που προσαρμόστηκαν και οι αντίστοιχες τιμές των μέτρων αξιολόγησης θεωρούνται αρκετά αξιόπιστες. Συνεπώς από τα δύο σύνολα δεδομένων και την εφαρμογή των παραπάνω μοντέλων κατηγοριοποίησης, μπορεί να εξαχθεί το συμπέρασμα ότι ο αλγόριθμος «Random Forest» μπορεί με αρκετά καλή ακρίβεια να προβλέπει αν ένα πλοίο «Bulk carrier» ενδέχεται να είναι ρυπογόνο ή μη ρυπογόνο καθώς τα έτη περνούν για να μπορούν οι ναυτιλιακές να γνωρίζουν άμεσα αν θα χρειαστεί να αλλάξουν στρατηγικό πλάνο ή όχι. Ικανοποιητικά αποτελέσματα παρουσιάζει και ο αλγόριθμος «Decision Trees» όπου το πλεονέκτημά του σε σχέση με τον κατηγοριοποιητή «Random Forest» είναι ότι εκτελείται πιο γρήγορα. Ο αλγόριθμος που παρουσιάζει την χειρότερη απόδοση και στα δύο σύνολα δεδομένων είναι ο αλγόριθμος «Gaussian Naive Bayes». Για το σύνολο των 11 μεταβλητών μπορεί επίσης να χρησιμοποιηθεί για την πρόβλεψη κατηγοριών και ο αλγόριθμος «Logistic Regression» αφού έχει πολύ καλή απόδοση. Μεταξύ των αλγορίθμων «Random Forest» και «Logistic Regression», τα αποτελέσματα του αλγορίθμου «Logistic Regression» είναι πιο εύκολα ερμηνεύσιμα οπότε για το συγκεκριμένο σύνολο μεταβλητών θεωρείται πιο εύχρηστος.

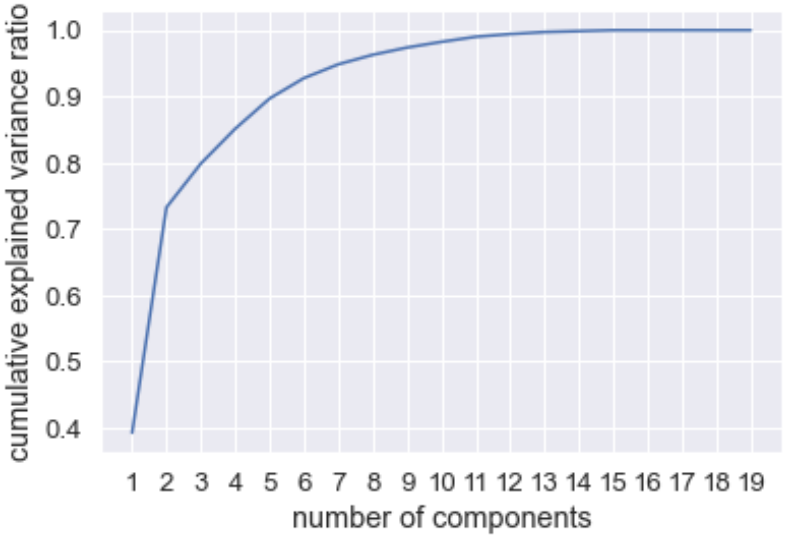
5.3 Εφαρμογή PCA και μιας τεχνικής Συσταδοποίησης στο συνόλο δεδομένων για εξαγωγή κρυφής και χρήσιμης γνώσης

Για το τελευταίο κομμάτι της ανάλυσης, χρησιμοποιήθηκε το αρχικό σύνολο δεδομένων εξαιρώντας τις δύο μεταβλητές «Annual average Fuel consumption per transport work (mass) [g / m tonnes Â· n miles]» και «Annual average CO2 emissions per transport work (mass) [g CO2 / m tonnes Â· n miles]». Η συσταδοποίηση χρησιμοποιείται με σκοπό την εξαγωγή κρυφής γνώσης από τα δεδομένα δημιουργώντας ομάδες από παρατηρήσεις που έχουν την μεγαλύτερη ομοιότητα και την μικρότερη απόσταση μεταξύ τους. Εφαρμόζοντας τεχνικές συσταδοποίησης στο σύνολο δεδομένων μπορούν να προκύψουν κρυφά πρότυπα και να εντοπιστεί κρυφή πληροφορία που δεν είναι εύκολα ανιχνεύσιμη. Όπως αναλύθηκε και στο α) μέρος του προβλήματος, τα δεδομένα και συγκεκριμένα πολλές από τις μεταβλητές του συνόλου δεδομένων είναι υψηλά συσχετισμένες μεταξύ τους. Για τον λόγο αυτό, θεωρήθηκε σημαντικό, πριν την εφαρμογή της συσταδοποίησης να χρησιμοποιηθεί η τεχνική της ανάλυσης σε κύριες συνιστώσες (PCA). Η PCA σκοπεύει όπως αναλύθηκε και στο κεφάλαιο 3, να αντικαταστήσει το αρχικό σύνολο μεταβλητών με ένα μικρότερο σύνολο ασυσχέτιστων μεταβλητών τις συνιστώσες που προκύπτουν από γραμμικούς συνδυασμούς των αρχικών μεταβλητών. Αυτοί οι γραμμικοί συνδυασμοί των αρχικών μεταβλητών που προκύπτουν, θα είναι ασυσχέτιστοι μεταξύ τους και θα περιέχουν ένα μεγάλο ποσοστό της πληροφορίας εξηγείται από τις αρχικές μεταβλητές. Για την εφαρμογή της ανάλυσης σε κύριες συνιστώσες προηγήθηκε τυποποίηση («Standardized») των δεδομένων. Η απόφαση για τον αριθμό των κύριων συνιστωσών που θα διατηρηθούν μπορεί να εξαχθεί είτε οπτικά μέσω του λεγόμενου διαγράμματος «scree – plot» που παρουσιάζεται στο σχήμα 33 είτε με την χρήση των ιδιοτιμών που έχει η κάθε κύρια συνιστώσα. Όσον αφορά τις ιδιοτιμές των κύριων συνιστωσών, υπάρχει ένα κριτήριο που επισημαίνει ότι αν έχει προηγηθεί τυποποίηση των δεδομένων, στο μοντέλο πρέπει να διατηρηθούν οι κύριες συνιστώσες που

έχουν ιδιοτιμή μεγαλύτερη της μονάδας, αν και αυτό το κριτήριο δεν πρέπει να χρησιμοποιείται πάντα τυφλά μιας και μπορεί να υπάρχουν κύριες συνιστώσες που εξηγούν μεγάλο κομμάτι της συνολικής διακύμανσης των αρχικών δεδομένων αλλά έχουν ιδιοτιμή μικρότερη της μονάδας.



Σχήμα 33: Scree plot για τις κύριες συνιστώσες



Σχήμα 34: Γράφημα της συνολικής διακύμανσης που εξηγείται από τις κύριες συνιστώσες

Από το scree plot του σχήματος 33, φαίνεται ότι οι τρεις πρώτες κύριες συνιστώσες πρέπει να διατηρηθούν αφού η καμπύλη μειώνεται συνεχώς και ξαφνικά αρχίζει στην τρίτη κύρια συνιστώσα να παρουσιάζει κλίση. Επίσης από το συγκεκριμένο σχήμα φαίνεται πως οι τρεις πρώτες κύριες συνιστώσες έχουν τιμή μεγαλύτερη της μονάδας. Πιο συγκεκριμένα η πρώτη συνιστώσα έχει ιδιοτιμή ίση με 7,46, η δεύτερη κύρια συνιστώσα έχει τιμή ίση με 6,46 ενώ η τρίτη κύρια συνιστώσα έχει τιμή 1,25. Στο σχήμα 34 φαίνεται η συνολική μεταβλητότητα που εξηγείται από τις κύριες συνιστώσες. Η πρώτη κύρια συνιστώσα εξηγεί το 39,2% της συνολικής διακύμανσης των δεδομένων, η δεύτερη κύρια συνιστώσα εξηγεί το 34,01% της συνολικής διακύμανσης ενώ η τρίτη κύρια συνιστώσα εξηγεί το 6,6% της συνολικής διακύμανσης. Μαζί οι πρώτες τρεις κύριες συνιστώσες εξηγούν το 79,81 της συνολικής διακύμανσης των δεδομένων. Τα φορτία αυτά για τις τρεις πρώτες κύριες συνιστώσες φαίνονται και στον παρακάτω πίνακα 17.

Μεταβλητή	PC1	PC2	PC3
X1_std	0,0520	0,3843	-0,0115
X2_std	0,0512	0,3845	-0,0114
X3_std	-0,0086	-0,0065	-0,0277
X4_std	-0,0666	0,219	0,4881
X5_std	0,0348	0,2975	-0,1816
X6_std	0,1032	0,2835	-0,2105
X7_std	-0,0034	0,2518	0,4763
X8_std	-0,0802	0,3718	-0,0806
X9_std	0,2827	-0,0040	0,4294
X10_std	0,2811	-0,0039	0,4322
X11_std	-0,0802	0,3718	-0,0806
X12_std	-0,0644	0,3802	-0,0921
X13_std	0,3505	0,0172	-0,1077
X14_std	0,3355	0,0118	-0,1033
X15_std	0,3346	0,0312	-0,1299
X16_std	0,3200	-0,0282	-0,0626
X17_std	0,3481	0,0020	-0,0817
X18_std	0,3479	0,0083	-0,1162
X19_std	0,3310	0,0222	0,0180
Διακύμανση που εξηγείται	39,2%	34,01%	6,6%

Πίνακας 17: Φορτία των τριών πρώτων κύριων συνιστωσών χρησιμοποιώντας το σύνολο των τυποποιημένων δεδομένων

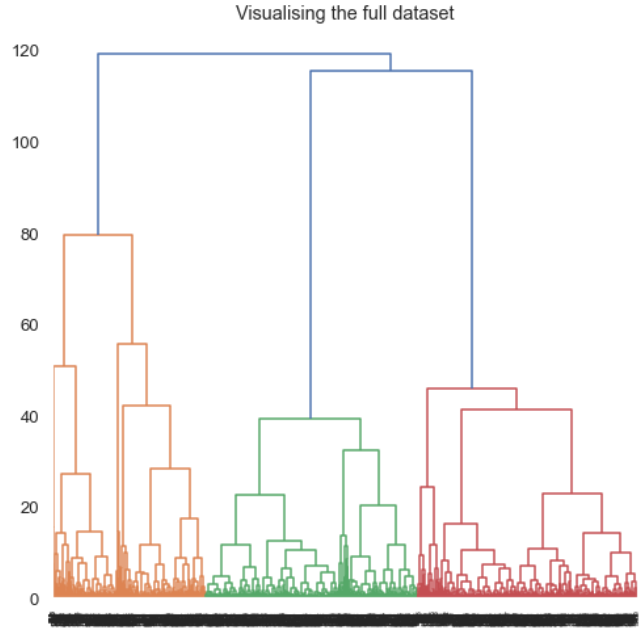
Αδιαφορώντας για το πρόσημο των τιμών, μεγάλες κατά απόλυτη τιμή, τιμές των φορτίων των κύριων συνιστωσών υποδηλώνουν ότι οι μεταβλητές έχουν σπουδαία επιρροή στην αντίστοιχη κύρια συνιστώσα. Σύμφωνα λοιπόν με τις τιμές του παραπάνω πίνακα φαίνεται πως η πρώτη κύρια συνιστώσα συνδέεται περισσότερο με τις μεταβλητές X13_std – X19_std

(χρωματισμένες με πράσινο) που στην πραγματικότητα είναι οι τυποποιημένες μεταβλητές του αρχικού συνόλου δεδομένων οι οποίες αντιπροσωπεύουν φυσικά χαρακτηριστικά του πλοίου και πιο συγκεκριμένα τις διαστάσεις του. Δηλαδή οι τυποποιημένες μεταβλητές X13_std – X19_std είναι στην ουσία οι μεταβλητές «ves_dwt», «ves_loa», «ves_draft», «ves_beam», «ves_capacity_grain», «ves_depth», «ves_main_engine_kw». Η δεύτερη κύρια συνιστώσα φαίνεται να σχετίζεται περισσότερο με τις μεταβλητές X1_std, X2_std, X8_std, X11_std, και X12_std. Οι τυποποιημένες αυτές μεταβλητές αντιπροσωπεύουν τις μεταβλητές «Total fuel consumption», «Total CO2 emissions», «Annual Total time spent at sea», « Total time spent at sea» και « Distance». Συνεπώς η δεύτερη κύρια συνιστώσα συνδέεται με τις συνολικές καταναλώσεις καυσίμου, τις συνολικές εκπομπές αερίων CO2 καθώς και με την λειτουργικότητα του πλοίου όσον αφορά τις αποστάσεις που διένυσε και τον χρόνο ταξιδιού του. Η τρίτη κύρια συνιστώσα σχετίζεται με τις μεταβλητές X4_std, X7_std, X9_std και X10_std οι οποίες αντιπροσωπεύουν τις τυποποιημένες τιμές των μεταβλητών «CO2 emissions from all voyages between ports under a MS jurisdiction», «CO2 emissions which occurred within ports under a MS jurisdiction at berth», « Annual average Fuel consumption per distance» και «Annual average CO2 emissions per distance» αντίστοιχα. Δηλαδή η τρίτη κύρια συνιστώσα πιθανόν να περιγράφει τις εκπομπές ρύπων και την κατανάλωση καυσίμου από τα πλοία ανά γεωγραφική περιοχή. Οι τιμές των φορτίων για την πρώτη κύρια συνιστώσα είναι θετικές, πράγμα που δείχνει ότι πλοία με μεγάλες διαστάσεις θα έχουν αντίστοιχα υψηλές και θετικές τιμές στην πρώτη κύρια συνιστώσα ενώ πλοία με μικρές διαστάσεις θα έχουν μικρές τιμές. Για την δεύτερη κύρια συνιστώσα παρατηρούνται θετικές τιμές φορτίων για τις μεταβλητές που αναφέρθηκαν οι οποίες σχετίζονται πολύ με την συγκεκριμένη κύρια συνιστώσα. Συνεπώς πλοία με υψηλές καταναλώσεις καυσίμου, υψηλές εκπομπές ρύπων και υψηλό συνολικό χρόνο που βρισκόταν στην θάλασσα, θα έχει ως αποτέλεσμα υψηλές τιμές στην δεύτερη κύρια συνιστώσα, ενώ πλοία με μικρές καταναλώσεις καυσίμου και χαμηλές εκπομπές ρύπων θα επιφέρουν χαμηλές τιμές στην δεύτερη κύρια συνιστώσα. Τέλος και για την τρίτη κύρια συνιστώσα παρατηρούνται θετικές τιμές φορτίων για τις μεταβλητές που αναφέρθηκαν προηγουμένως. Οπότε πλοία με υψηλές εκπομπές CO2 και κατανάλωση καυσίμου σε περιοχές εντός των κρατών μελών της Ευρωπαϊκής Ένωσης θα έχουν μεγάλες τιμές στην τρίτη κύρια συνιστώσα ενώ πλοία με μειωμένη κατανάλωση καυσίμου και με μειωμένες εκπομπές CO2 εντός των κρατών μελών της Ευρωπαϊκής Ένωσης θα έχουν αντίστοιχα μικρές τιμές στην τρίτη κύρια συνιστώσα. Έχοντας δει με ποιες μεταβλητές συνδέονται και σχετίζονται οι τρεις πρώτες τρεις κύριες συνιστώσες, η ερμηνεία τους θα μπορούσε να ήταν η εξής:

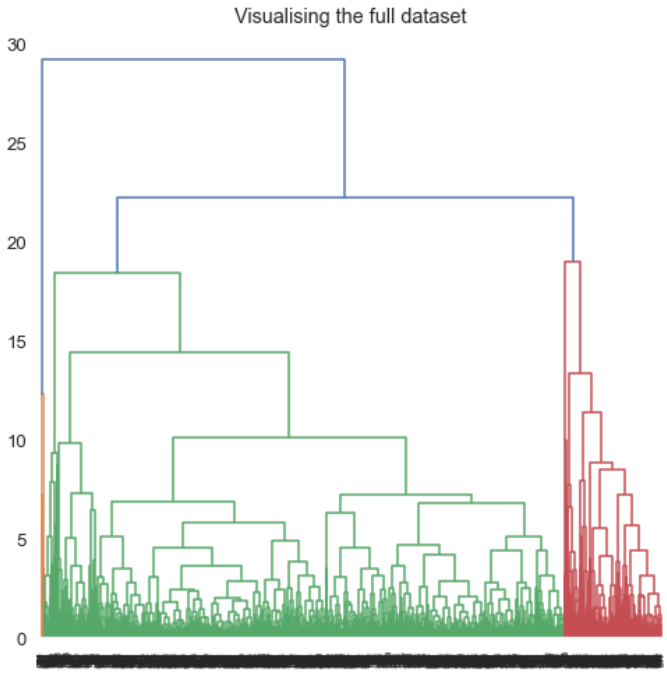
- $1^{η}$ Κύρια συνιστώσα: Διαστάσεις του πλοίου
- $2^{η}$ Κύρια συνιστώσα: Περιβαλλοντική απόδοση του πλοίου
- $3^{η}$ Κύρια συνιστώσα: Περιβαλλοντική απόδοση του πλοίου αναλόγως την περιοχή που υλοποίησε δρομολόγια

Ένα λοιπόν χρήσιμο συμπέρασμα από την εφαρμογή κύριων συνιστωσών ήταν ότι οι τρεις πρώτες κύριες συνιστώσες βοήθησαν στην ερμηνεία τριών νέων μη μετρήσιμων μεταβλητών που αντιπροσωπεύουν τις διαστάσεις των πλοίων και την περιβαλλοντική τους απόδοση τόσο σε χρόνο όσο και σε περιοχή που διεκπεραιώθηκαν τα ταξίδια.

Στο επόμενο κομμάτι της ανάλυσης, εφαρμόστηκε η ιεραρχική μέθοδος συσταδοποίησης για την περαιτέρω εξερεύνηση του συνόλου δεδομένων. Η συσταδοποίηση εφαρμόστηκε στις τρεις διαμορφωμένες συνιστώσες, δηλαδή στους τρεις παράγοντες που προέκυψαν από την μέθοδο PCA με σκοπό να ομαδοποιηθούν τα πλοία σε συστάδες ανάλογα με τις διαστάσεις και τα περιβαλλοντικά τους χαρακτηριστικά σε σχέση με τις καταναλώσεις τους, τον χρόνο διεκπεραίωσης των ταξιδιών και την περιοχή που εκτελούν τα δρομολόγιά τους. Επειδή δεν είναι γνωστός εκ των προτέρων ο αριθμός των συστάδων που πρέπει να δημιουργηθούν, ένας τρόπος για να εξακριβωθεί το πόσες συστάδες δημιουργούνται είναι μέσω του δενδρογράμματος. Για τον υπολογισμό των αποστάσεων μεταξύ των παρατηρήσεων χρησιμοποιήθηκε η μέθοδος «ward» μιας και χρησιμοποιείται πολύ συχνά στην πράξη λόγω των καλών ιδιοτήτων που έχει όπως το να δημιουργεί ομάδες με παρόμοιο αριθμό παρατηρήσεων. Στο παρακάτω σχήμα 35, φαίνεται το δενδρόγραμμα της ιεραρχικής συσταδοποίησης. Από το σχήμα προκύπτει ότι έχουν δημιουργηθεί τέσσερις συστάδες. Εφαρμόζοντας ιεραρχική συσταδοποίηση και με την μέθοδο «Furthest Neighbor» που φαίνεται στο σχήμα 36, προέκυψαν τα ίδια αποτελέσματα σχετικά με τον αριθμό συστάδων που δημιουργούνται. Φυσικά για να βρεθεί πιο αξιόπιστα ο αριθμός των συστάδων και πότε εξάγονται τα καλύτερα αποτελέσματα, μπορούν να χρησιμοποιηθούν οι δείκτες «Silhouette Coefficient» και «Davies-Bouldin». Για τα ζευγάρια των τριών συνιστωσών, δημιουργήθηκαν συστάδες για αριθμούς συστάδων από 2 μέχρι και 6, και κατασκευάστηκαν τα αντίστοιχα διαγράμματα που δείχνουν τις τιμές των δεικτών «Silhouette Coefficient» και «Davies Bouldin» ανά περίπτωση. Επίσης υπολογίστηκαν οι ακριβείς τιμές των συγκεκριμένων δεικτών αξιολόγησης της συσταδοποίησης καθώς και ο δείκτης «Calinski – Harabasz».

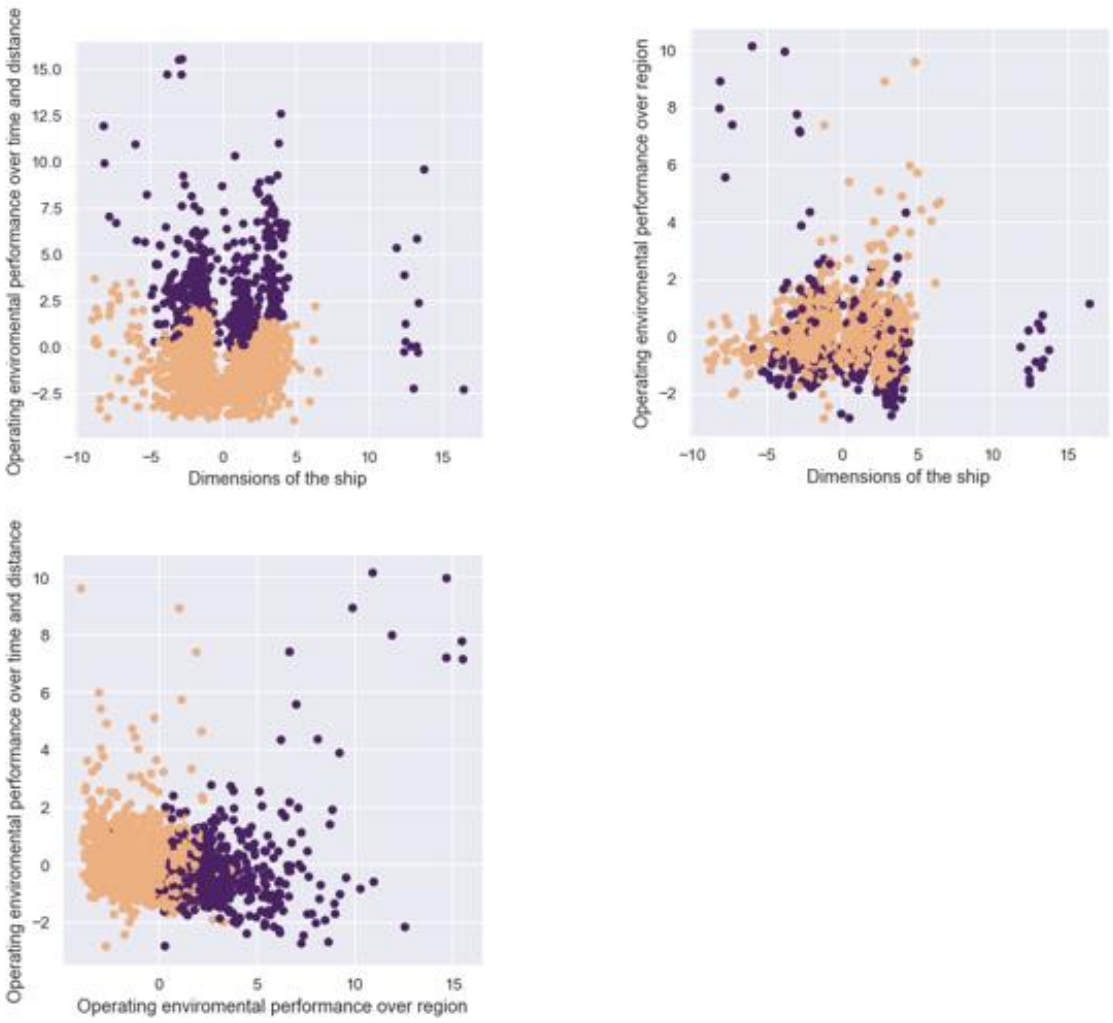


Σχήμα 35: Δενδρόγραμμα χρησιμοποιώντας την μέθοδο «Ward»



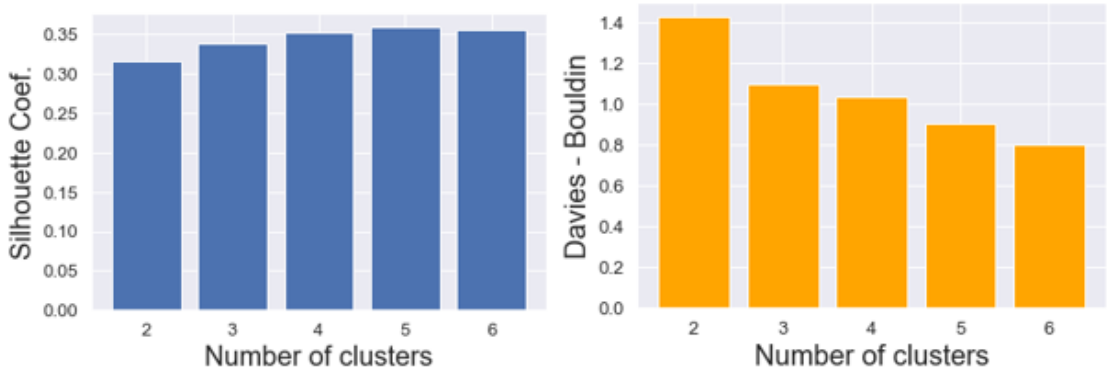
Σχήμα 36: Δενδρόγραμμα χρησιμοποιώντας την μέθοδο «Furthest Neighbor»

Στο παρακάτω σχήμα 37 φαίνονται οι 2 δημιουργημένες συστάδες για τα ζευγάρια συνιστωσών. Δεν παρατηρείται και πολύ καλός διαχωρισμός των παρατηρήσεων παρά μόνο στην περίπτωση που χρησιμοποιείται η πρώτη και η δεύτερη κύρια συνιστώσα δηλαδή οι διαστάσεις του πλοίου και η ενεργειακή του απόδοση αναλόγως τον χρόνο και την απόσταση που διένυσε.



Σχήμα 37: Οι δύο δημιουργημένες συστάδες για τα τρία ζευγάρια συνιστωσών. Το πάνω αριστερά σχήμα αφορά την πρώτη και την δεύτερη κύρια συνιστώσα ενώ το πάνω δεξιά σχήμα αφορά την πρώτη και την τρίτη κύρια συνιστώσα. Το κάτω αριστερά σχήμα αφορά την δεύτερη και την τρίτη κύρια συνιστώσα

Στο επόμενο σχήμα 38 φαίνονται οι τιμές των δεικτών «Silhouette coefficient» και «Davies – Bouldin» για κάθε αριθμό συστάδων για τις τρεις πρώτες κύριες συνιστώσες που χρησιμοποιήθηκαν. Παρατηρείται ότι καλύτερη συσταδοποίηση υλοποιείται για τέσσερις συστάδες. Οι ακριβείς τιμές των δεικτών αυτών καθώς και του δείκτη «Calinski – Harabasz» φαίνονται στον πίνακα 18 παρακάτω.



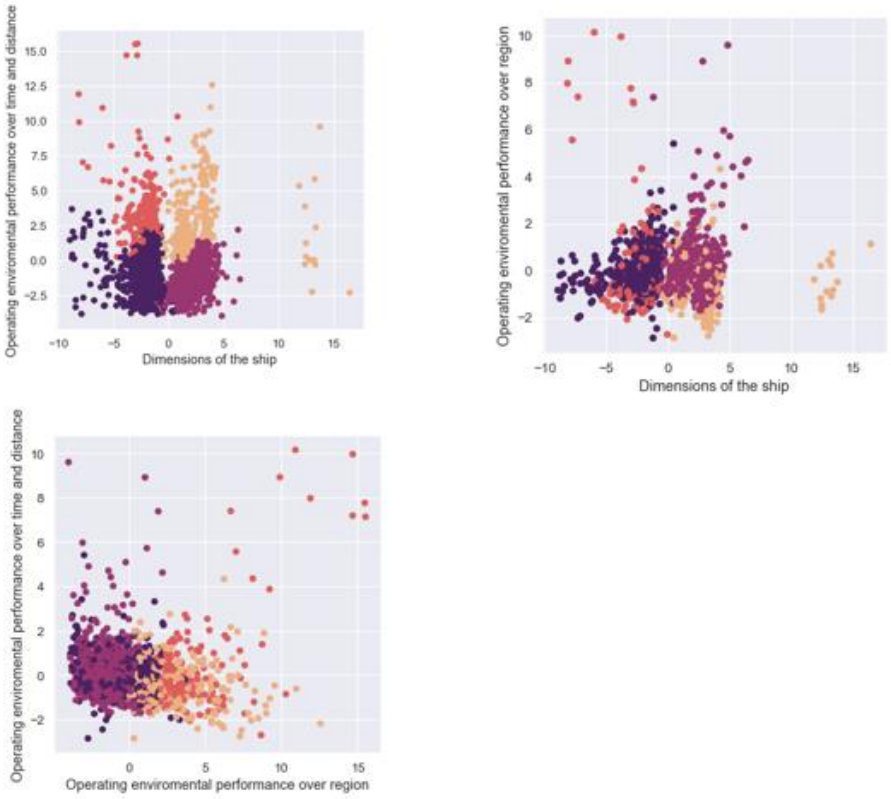
Σχήμα 38: Διαγραμματική απεικόνιση τιμών των δεικτών «Silhouette coefficient» και «Davies - Bouldin» για τον αριθμό συστάδων που δημιουργούνται χρησιμοποιώντας τις πρώτες τρεις κύριες συνιστώσες

Ζευγάρι συνιστωσών	Αριθμός συστάδων	Silhouette Coefficient	Davies Bouldin	Calinski Harabasz
PC1 & PC2 & PC3	2	0,316	1,424	615,606
PC1 & PC2 & PC3	3	0,337	1,093	848,906
PC1 & PC2 & PC3	4	0,351	1,037	870,436
PC1 & PC2 & PC3	5	0,359	0,901	811,657
PC1 & PC2 & PC3	6	0,355	0,798	786,890

Πίνακας 18: Τιμές δεικτών «Silhouette coefficient», «Davies – Bouldin» και «Calinski – Harabasz» για τις τρεις πρώτες κύριες συνιστώσες ανά αριθμό συστάδων

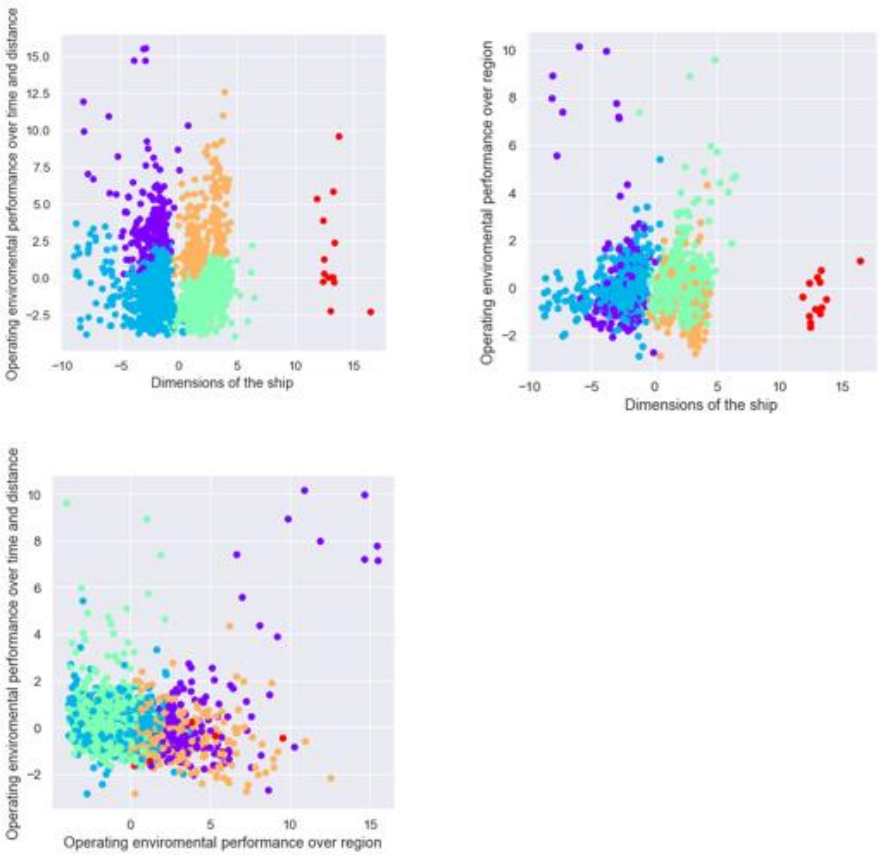
Ο δείκτης «Silhouette coefficient» παίρνει την υψηλότερη τιμή για αριθμό συστάδων ίσο με 5 ενώ ο δείκτης «Calinski – Harabasz» παίρνει την υψηλότερη τιμή για αριθμό συστάδων ίσο με 4. Την μικρότερη τιμή ο δείκτης «Davies – Bouldin» την έχει για αριθμό συστάδων επίσης ίσο με 6. Συνεπώς η ομαδοποίηση παρόμοιων παρατηρήσεων πραγματοποιείται σε 4,5 ή και 6 συστάδες. Στην συνέχεια, στο σχήμα 39, φαίνονται οι τέσσερις δημιουργημένες συστάδες για τα ζευγάρια των κυρίων συνιστωσών. Γενικά με εξαίρεση το ζευγάρι της πρώτης και της δεύτερης κύριας συνιστώσας, φαίνεται έντονα το φαινόμενο των υπερκαλυπτόμενων συστάδων που είναι γνωστό με τον όρο «overlapping», όπου υποδηλώνει ότι πιθανότητα κάποιες παρατηρήσεις δεν έχουν τοποθετηθεί σε σωστές συστάδες. Γενικά μπορεί να φανεί από τα σχήματα των δύο συστάδων ότι υπάρχει καλός σχετικά διαχωρισμός των πλοίων ως προς τις διαστάσεις τους και την περιβαλλοντική απόδοσής τους αναλόγως τους ρύπους που εκπέμπουν, την κατανάλωση καυσίμου, τον συνολικό χρόνο ταξιδιού τους και την συνολική απόσταση που διένυσαν. Επιπλέον φαίνεται να γίνεται ικανοποιητικός διαχωρισμός των πλοίων σε δύο συστάδες χρησιμοποιώντας την περιβαλλοντική τους απόδοση αναλόγως τον χρόνο και την απόσταση που διένυσαν και την περιβαλλοντική τους απόδοση αναλόγως την περιοχή που βρισκόντουσαν. Για τις 4 συστάδες παρατηρείται καλός διαχωρισμός σχετικά μόνο ως προς τις φυσικές διαστάσεις του πλοίου και την περιβαλλοντική τους απόδοση αναλόγως του συνολικού χρόνου ταξιδιού τους και την συνολική απόσταση που διένυσαν, δηλαδή καλός διαχωρισμός επιτυγχάνεται για τις πρώτες δύο κύριες συνιστώσες και όχι για τα υπόλοιπα ζευγάρια συνιστωσών. Παρατηρώντας τις

δύο δημιουργημένες συστάδες αλλά και τις τέσσερις, τον μεγαλύτερο ρόλο στον καλύτερο διαχωρισμό συστάδων φαίνεται να τον έχει η δεύτερη κύρια συνιστώσα δηλαδή η περιβαλλοντική απόδοση των πλοίων αναλόγως του συνολικού χρόνου ταξιδιού τους και την συνολική απόσταση που διένυσαν ενώ την δεύτερη «μεγαλύτερη» επιρροή φαίνεται να την έχουν οι διαστάσεις των πλοίων.



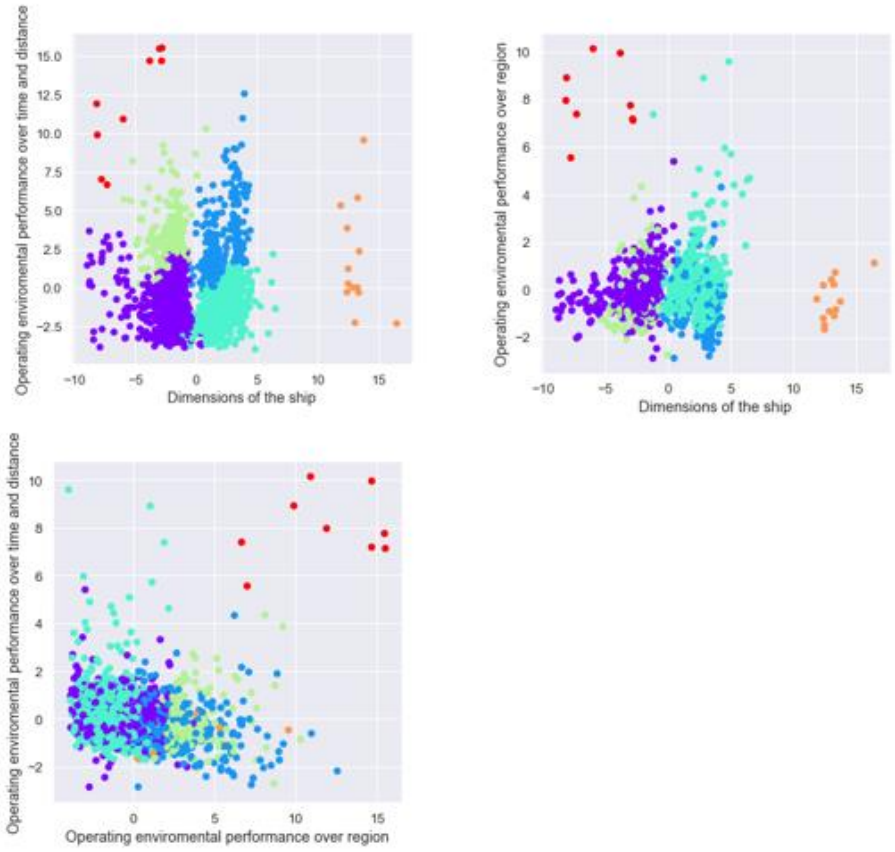
Σχήμα 39: Οι τέσσερις δημιουργημένες συστάδες για τα τρία ζευγάρια συνιστωσών. Το πάνω αριστερά σχήμα αφορά την πρώτη και την δεύτερη κύρια συνιστώσα ενώ το πάνω δεξιά σχήμα αφορά την πρώτη και την τρίτη κύρια συνιστώσα. Το κάτω αριστερά σχήμα αφορά την δεύτερη και την τρίτη κύρια συνιστώσα.

Στο παρακάτω σχήμα 40, φαίνονται οι πέντε δημιουργημένες συστάδες. Και σε αυτήν την περίπτωση, καλός διαχωρισμός των συστάδων παρατηρείται μόνο στην πάνω αριστερή εικόνα που χρησιμοποιούνται οι διαστάσεις του πλοίου και η ενεργειακή του απόδοση αναλόγως τις εκπομπές ρύπων, την κατανάλωση καυσίμου, τον χρόνο και την απόσταση που διένυσε. Παρατηρήσεις με κόκκινο χρώμα οι οποίες αποτελούν μια συστάδα, πρόκειται πιθανόν για outliers μιας και οι τιμές αυτές βρίσκονται αρκετά μακριά από τις υπόλοιπες και είναι ελάχιστες. Συνεπώς η ύπαρξη των ακραίων τιμών πιθανόν να επηρεάζει την δημιουργία των συστάδων και για αυτό τον λόγο να φαίνεται αναγκαία η δημιουργία πάνω από τέσσερις συστάδες.



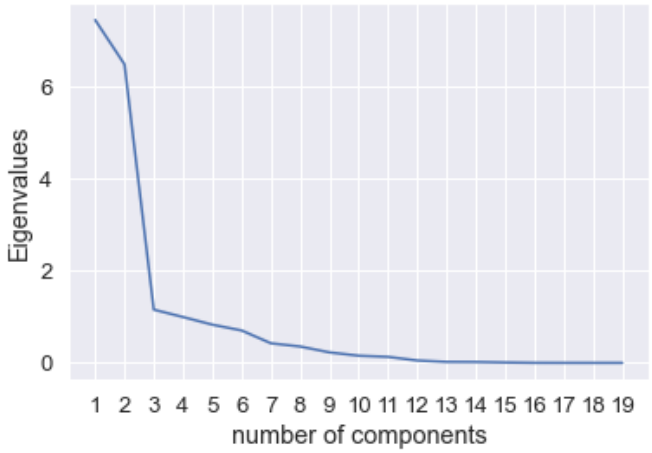
Σχήμα 40: Οι πέντε δημιουργημένες συστάδες για τα τρία ζευγάρια συνιστωσών. Το πάνω αριστερά σχήμα αφορά την πρώτη και την δεύτερη κύρια συνιστώσα ενώ το πάνω δεξιά σχήμα αφορά την πρώτη και την τρίτη κύρια συνιστώσα. Το κάτω αριστερά σχήμα αφορά την δεύτερη και την τρίτη κύρια συνιστώσα

Παρακάτω στο σχήμα 41, φαίνονται οι έξι δημιουργημένες συστάδες. Και σε αυτήν την περίπτωση φαίνεται οι ακραίες παρατηρήσεις να επηρεάζουν την δημιουργία των συστάδων με αποτέλεσμα να δημιουργούνται επιπλέον συστάδες με ελάχιστο αριθμό παρατηρήσεων. Πάλι καλή συσταδοποίηση επιτυγχάνεται όταν χρησιμοποιούνται η πρώτη και η δεύτερη κύρια συνιστώσα. Στις άλλες περιπτώσεις συνιστωσών παρατηρείται ξανά, έντονα, το φαινόμενο της υπερκάλυψης των συστάδων. Συνεπώς το συμπέρασμα φαινομενικά είναι, ότι καλύτερη και πιο ουσιώδης συσταδοποίηση, επιτυγχάνεται όταν ο αριθμός τον συστάδων είναι ίσος με 4. Ο δείκτης «Silhouette Coefficient» δεν διαφέρει σημαντικά μεταξύ των τεσσάρων, των πέντε και των έξι συστάδων. Η ενδεχόμενη απομάκρυνση των ακραίων τιμών, ίσως επιφέρει πιο μεγάλες διαφοροποιήσεις στον συγκεκριμένο δείκτη ανάλογα με τον αριθμό των συστάδων. Οπότε για να αποδειχθεί ο ισχυρισμός περί τεσσάρων συστάδων πρέπει να πραγματοποιηθεί και ανάλυση χωρίς την ύπαρξη των ακραίων παρατηρήσεων.



Σχήμα 41: Οι έξι δημιουργημένες συστάδες για τα τρία ζευγάρια συνιστωσών. Το πάνω αριστερά σχήμα αφορά την πρώτη και την δεύτερη κύρια συνιστώσα ενώ το πάνω δεξιά σχήμα αφορά την πρώτη και την τρίτη κύρια συνιστώσα. Το κάτω αριστερά σχήμα αφορά την δεύτερη και την τρίτη κύρια συνιστώσα

Προκειμένου να αποδειχθεί λοιπόν ο ισχυρισμός για την δημιουργία τεσσάρων συστάδων, ξαναγίνονται οι υπολογισμοί των κυρίων συνιστωσών αλλά και η δημιουργία των συστάδων για τα δεδομένα που δεν περιέχουν τα outliers. Η απομάκρυνση των ακραίων τιμών που φαίνεται να επηρέαζαν την δημιουργία συστάδων έγινε με την βοήθεια των Z - σκορ. Για να ανιχνευθούν ακραίες τιμές χρησιμοποιώντας τα Z - σκορ , τέθηκε ένα όριο για οποιεσδήποτε τιμές εκτός των +3 ή -3 τυπικών αποκλίσεων από τη μέση τιμή. Οποιεσδήποτε τιμές εκτός των ορίων αυτών θεωρήθηκαν ακραίες τιμές και αφαιρέθηκαν από το σύνολο δεδομένων. Μετά την απομάκρυνση των ακραίων τιμών, στο σύνολο παρέμειναν συνολικά 1789 εγγραφές πλοίων. Για τον αριθμό τον κυρίων συνιστωσών που θα χρησιμοποιηθούν, πάλι, στο παρακάτω σχήμα 42, φαίνεται το scree plot. Από το scree plot παρατηρείται πως χρήσιμες είναι οι πρώτες τρεις κύριες συνιστώσες. Πράγματι, οι 3 πρώτες κύριες συνιστώσες εξηγούν μαζί το 80% περίπου της συνολικής μεταβλητότητας των αρχικών δεδομένων όπως δηλαδή και στην περίπτωση που υπήρχαν ακραίες παρατηρήσεις στο σύνολο δεδομένων. Επόμενο βήμα είναι να βρεθούν τα φορτία των τριών κυρίων συνιστωσών για να εξακριβωθεί με ποιες μεταβλητές σχετίζονται. Τα φορτία αυτά παρατηρούνται στον πίνακα 19 στην συνέχεια.



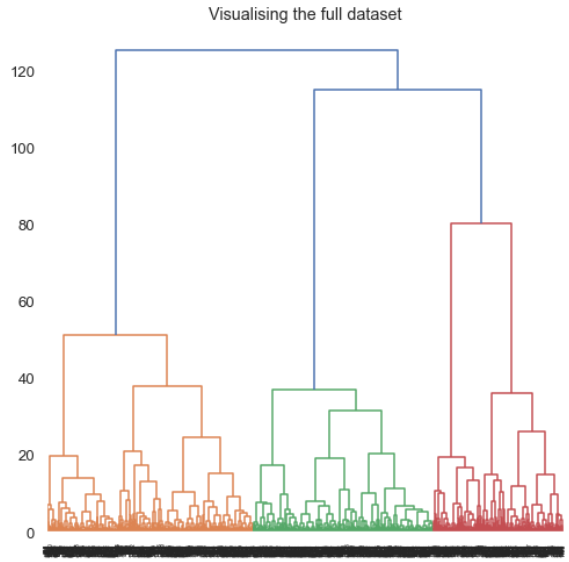
Σχήμα 42: Scree plot για δεδομένα χωρίς outliers

Μεταβλητή	PC1	PC2	PC3
X1_std	0,012	0,387	0,002
X2_std	0,012	0,387	0,003
X3_std	0,029	-0,017	0,115
X4_std	0,079	0,1929	0,471
X5_std	0,031	0,2966	-0,089
X6_std	-0,039	0,2974	-0,125
X7_std	0,031	0,2644	0,393
X8_std	0,130	0,354	-0,105
X9_std	-0,287	0,041	0,4614
X10_std	-0,286	0,041	0,465
X11_std	0,138	0,354	-0,105
X12_std	0,118	0,363	-0,128
X13_std	-0,348	0,066	-0,159
X14_std	-0,335	0,062	-0,157
X15_std	-0,317	0,088	-0,168
X16_std	-0,307	0,018	-0,046
X17_std	-0,349	0,057	-0,119
X18_std	-0,349	0,061	-0,145
X19_std	-0,319	0,071	0,047
Διακύμανση που εξηγείται	39,24%	34,017%	6,61%

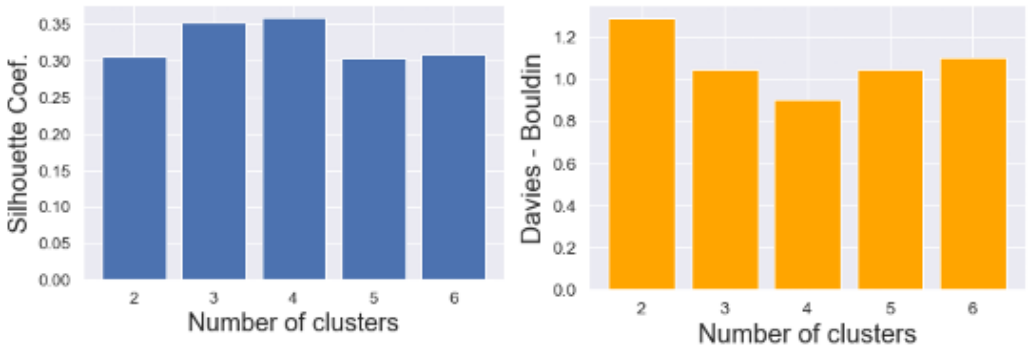
Πίνακας 19: Φορτία των τριών πρώτων κύριων συνιστωσών χρησιμοποιώντας το σύνολο των τυποποιημένων δεδομένων χωρίς outliers

Η ερμηνεία των κυρίων συνιστωσών είναι ίδια σαν της περίπτωσης του συνόλου δεδομένων που περιείχε ακραίες παρατηρήσεις. Οι κύριες συνιστώσες συνδέονται με τις αντίστοιχες ίδιες τυποποιημένες μεταβλητές. Η μόνη διαφοροποίηση που παρατηρείται είναι ότι τα φορτία της πρώτης κύριας συνιστώσας για τις μεταβλητές με τις οποίες σχετίζεται πολύ, είναι αρνητικά..

Στην συνέχεια εφαρμόζεται ιεραρχική συσταδοποίηση με βάση τις πρώτες κύριες συνιστώσες . Στο σχήμα 43 διακρίνεται το δενδρόγραμμα με την μέθοδο «Ward» ενώ στο σχήμα 44 φαίνονται γραφικά οι τιμές των δεικτών «Silhouette coefficient» και «Davies - Bouldin» για αριθμός συστάδων από 2 μέχρι 6. Οι ακριβείς τιμές των δεικτών αυτών και του δείκτη «Calinski – Harabasz» φαίνονται στον πίνακα 20.



Σχήμα 43: Δενδρόγραμμα με την μέθοδο «Ward» για το σύνολο δεδομένων χωρίς outliers χρησιμοποιώντας τις τρεις πρώτες κύριες συνιστώσες



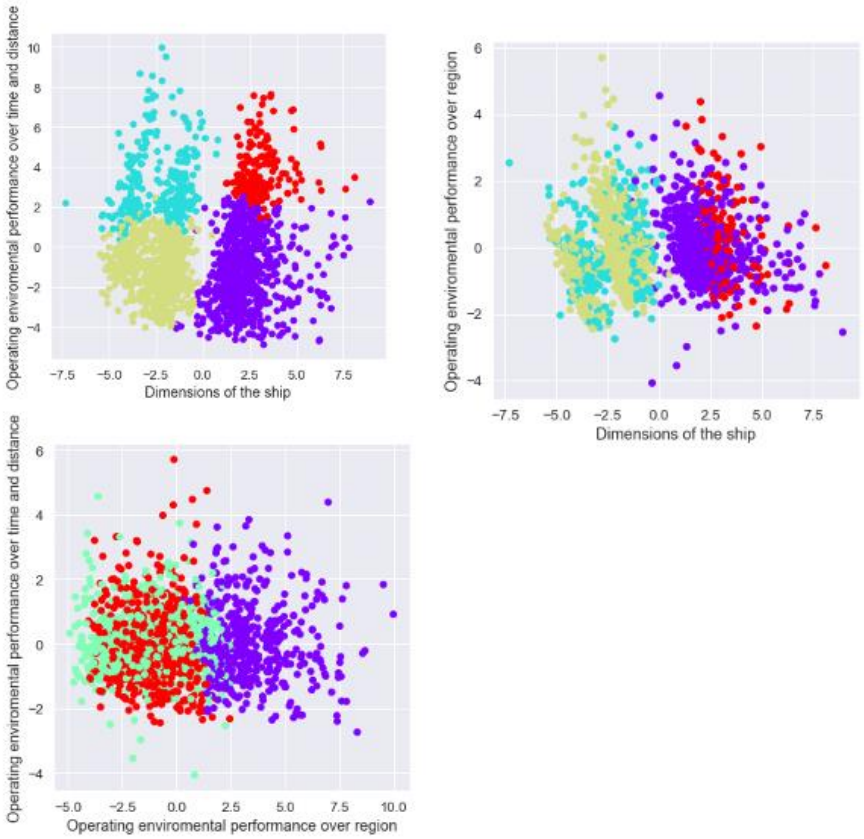
Σχήμα 44: Γραφική αναπαράσταση δεικτών «Silhouette coefficient» και «Davies – Bouldin» για τον αριθμό συστάδων με βάση τις πρώτες τρεις κύριες συνιστώσες για το σύνολο δεδομένων χωρίς outliers

Ζευγάρι	Αριθμός	Silhouette	Davies Bouldin	Calinski

συνιστωσών	συστάδων	Coefficient		Harabasz
PC1 & PC2 & PC3	2	0,305	1,286	732,102
PC1 & PC2 & PC3	3	0,352	1,047	1029,909
PC1 & PC2 & PC3	4	0,359	0,903	1128,749
PC1 & PC2 & PC3	5	0,303	1,044	1057,136
PC1 & PC2 & PC3	6	0,309	1,099	963,270

Πίνακας 20: Τιμές δεικτών «Silhouette coefficient», «Davies – Bouldin» και «Calinski – Harabasz» για τις τρεις πρώτες κύριες συνιστώσες ανά αριθμό συστάδων όταν δεν υπάρχουν outliers στο σύνολο δεδομένων

Παρατηρώντας το σχήμα 44 αλλά και τον πίνακα 20, προκύπτει ότι ο πιο κατάλληλος αριθμός συστάδων για τα συγκεκριμένα δεδομένα είναι τέσσερις συστάδες. Επομένως, σχηματίζονται 4 συστάδες για τα τρία ζευγάρια των τριών κυρίων συνιστωσών. Τα αποτελέσματα της συσταδοποίησης φαίνονται στο σχήμα 46 παρακάτω.



Σχήμα 45: Οι τέσσερις δημιουργημένες συστάδες για τα τρία ζευγάρια συνιστωσών. Το πάνω αριστερά σχήμα αφορά την πρώτη και την δεύτερη κύρια συνιστώσα ενώ το πάνω δεξιά σχήμα αφορά την πρώτη και την τρίτη κύρια συνιστώσα. Το κάτω αριστερά σχήμα αφορά την δεύτερη και την τρίτη κύρια συνιστώσα (χωρίς την ύπαρξη των outliers)

Από το σχήμα 45 παρατηρείται ότι, καλός διαχωρισμός επιτυγχάνεται όταν χρησιμοποιούνται οι πρώτες δύο κύριες συνιστώσες που αναφέρονται στις διαστάσεις του πλοίου και την περιβαλλοντική απόδοση του πλοίου αντίστοιχα. Τελικά στην πρώτη συστάδα εντοπίστηκαν 716 πλοία, στην δεύτερη συστάδα βρέθηκαν 280 πλοία, στην τρίτη συστάδα υπήρχαν 624 πλοία ενώ στην τέταρτη συστάδα υπήρχαν 169 πλοία. Σύμφωνα με τις διαστάσεις των πλοίων μέσα σε κάθε συστάδα και σύμφωνα με τις τιμές των εκπομπών ρύπων σε κάθε συστάδα τα ονόματα των συστάδων είναι:

- $1^{η}$ Συστάδα: Μικρών διαστάσεων πλοία που είναι φιλικά προς το περιβάλλον
- $2^{η}$ Συστάδα: Μεγάλων διαστάσεων πλοία που δεν είναι φιλικά προς το περιβάλλον
- $3^{η}$ Συστάδα: Μεγάλων διαστάσεων πλοία που είναι φιλικά προς το περιβάλλον
- $4^{η}$ Συστάδα: Μικρών διαστάσεων πλοία που δεν είναι φιλικά προς το περιβάλλον

Παρατηρώντας το σχήμα 45 και το πάνω αριστερά γράφημα, η πρώτη συστάδα είναι χρωματισμένη με λαχανί χρώμα, η δεύτερη συστάδα είναι χρωματισμένη με κόκκινο χρώμα, η τρίτη συστάδα είναι χρωματισμένη με μωβ χρώμα, ενώ η $4^{η}$ συστάδα είναι χρωματισμένη με γαλάζιο χρώμα. Επομένως το πόρισμα της ανάλυσης ήταν ότι δημιουργήθηκαν 4 συστάδες για το σύνολο δεδομένων με βάση τις διαστάσεις και την περιβαλλοντική απόδοση των πλοίων.

ΚΕΦΑΛΑΙΟ 6^{Ο}

6. Συμπεράσματα

Η παρούσα διπλωματική εργασία αφορούσε την βιομηχανία της ναυτιλίας, την λειτουργία της, τα προβλήματα που αντιμετωπίζει ο συγκεκριμένος κλάδος καθώς και τις νέες προκλήσεις που καθημερινά εμφανίζονται στο συγκεκριμένο χώρο. Έγινε αναφορά σε διάφορα προβλήματα που ταλανίζουν τις ναυτιλιακές επιχειρήσεις σχετικά με τις εκπομπές ρύπων, τις καταναλώσεις καυσίμου των πλοίων, την βελτιστοποίηση των ταξιδιών αλλά και την εύρεση χρήσιμης γνώσης μέσα από τον τεράστιο όγκο δεδομένων που έχουν στην διάθεσή τους οι ναυτιλιακές εταιρίες. Έγινε περιγραφή για το πως η στατιστική και η πληροφορική και πιο συγκεκριμένα, η στατιστική μηχανική μάθηση μπορεί να βοηθήσει σημαντικά στην εύρεση νέων λύσεων, καινοτόμων και οικονομικών τρόπων ταξιδιού καθώς και στην εύρεση κρυφών προτύπων και σχέσεων μέσα από έναν τεράστιο όγκο δεδομένων που αφορούν την ναυτιλία. Αναφέρθηκαν και επεξηγήθηκαν με αρκετή λεπτομέρεια διάφορες τεχνικές μηχανικής μάθησης, παρουσιάστηκαν μελέτες που στηρίχθηκαν σε αληθινά δεδομένα και προβλήματα του ναυτιλιακού κλάδου και στο τέλος εφαρμόστηκαν μερικές τεχνικές μηχανικής μάθησης όπως κατηγοριοποίηση και συσταδοποίηση πάνω σε αληθινά δεδομένα για την εξαγωγή χρήσιμων, αξιόπιστων αποτελεσμάτων και αποφάσεων.

Η πρώτη μελέτη που έγινε στα πλαίσια της διπλωματικής εργασίας αφορούσε τις εκπομπές διοξειδίου του άνθρακα από τα πλοία και σκοπό είχε την εκτίμηση για το αν ένα πλοίο που μετέφερε χύδην φορτία, δηλαδή τύπου «Bulk Carrier» θα θεωρούταν ρυπογόνο ή μη ρυπογόνο σε βάθος τεσσάρων ετών, με τις εκτιμήσεις να ξεκινούν από το 2023 και να συνεχίζονται μέχρι και το 2026. Τα πλοία που χρησιμοποιήθηκαν για την ανάλυση περιείχαν στοιχεία για την περίοδο τους έτους 2021, σχετικά με τα περιβαλλοντική τους απόδοση και τις φυσικές διαστάσεις τους. Τα δεδομένα προήλθαν από το επίσημο site του MRV σε συνδυασμό με στοιχεία πλοίων που αφορούσαν τις διαστάσεις τους και τα οποία προήλθαν από μια ναυτιλιακή εταιρία. Η εκτίμηση για το εάν ένα πλοίο ενδέχεται να είναι ρυπογόνο ή όχι είναι ένα υψίστης σημασίας πρόβλημα που καλούνται οι ναυτιλιακές εταιρίες να επιλύσουν μιας και από το 2023 θα πρέπει να παραθέτουν περιβαλλοντικά στοιχεία σχετικά με τους ρύπους που εκπέμπουν στην ατμόσφαιρα τα πλοία τους και αναλόγως ρύπανσης ή μη ρύπανσης θα χρειάζεται να αλλάξουν ή αντίστοιχα να μην αλλάξουν στρατηγικό πλάνο ως προς την διαχείριση των πλοίων. Για αυτή την εκτίμηση χρησιμοποιήθηκε ο γνωστός δείκτης από την διεθνή βιβλιογραφία που ονομάζεται C.I.I. (Carbon Intensity Indicator). Ο λόγος εφαρμογής αυτού του δείκτη είναι η μεγάλη πρόκληση και στόχος που έχει τεθεί από τον διεθνή οργανισμό ναυτιλίας (IMO) και από την Ευρωπαϊκή Ένωση, που είναι η μείωση των εκπομπών ρύπων από τα πλοία στο μισό μέχρι το 2050. Σύμφωνα με τις τιμές αυτού του δείκτη για κάθε έτος, τα πλοία ταξινομούνται σε μια από πέντε κατηγορίες A, B, C, D και E αναλόγως τις εκπομπές ρύπων τους. Εφαρμόζοντας την ισχύ αυτού του δείκτη για πλοία που είχαν περιβαλλοντικά και άλλα στοιχεία για τα ημερολογιακά έτη 2019 και μετέπειτα, και ξεκινώντας τις προβλέψεις από το 2023 μέχρι το 2026, αν ένα πλοίο

ταξινομείται στην κατηγορία D για τρία συνεχόμενα έτη ή στην κατηγορία E έστω και για ένα έτος, τότε θα θεωρείται ρυπογόνο, εναλλακτικά θα θεωρείται μη ρυπογόνο. Υπολογίζοντας λοιπόν αυτό τον δείκτη βρέθηκε ότι από τα 1942 πλοία που εξετάστηκαν, το 49% αυτών μέχρι και το 2026 θα ήταν ρυπογόνα ενώ το υπόλοιπο 51% θα ήταν μη ρυπογόνα. Αυτή η ταξινόμηση των πλοίων σε δύο κατηγορίες με βάσει τις εκτιμήσεις του δείκτη C.I.I. για τους ρύπους που εκπέμπουν, έδωσε την ευκαιρία και το έναυσμα για την διεκπεραίωση μιας δεύτερης σημαντικής μελέτης στον χώρο της ναυτιλίας.

Η δεύτερη μελέτη βασίστηκε στις εκτιμήσεις του δείκτη C.I.I. σχετικά με τις εκπομπές ρύπων των πλοίων. Ουσιαστικά δημιουργήθηκε μια καινούργια μεταβλητή η οποία περιείχε την κατηγορία που ανήκαν τα πλοία. Είτε τα πλοία θα ήταν ρυπογόνα, είτε μη ρυπογόνα. Μεγάλο ενδιαφέρον λοιπόν, παρουσίαζε για τις ναυτιλιακές εταιρίες, το να προβλέπεται για νέα δεδομένα πλοίων που αφορούν τις εκπομπές τους, τις καταναλώσεις τους ή και τις διαστάσεις τους, το αν με βάση τέτοιου είδους παραμέτρους, επρόκειτο αυτά τα πλοία να είναι ρυπογόνα ή μη ρυπογόνα. Για την διεξαγωγή λοιπόν αυτής της μελέτης εφαρμόστηκαν διάφοροι αλγόριθμοι κατηγοριοποίησης οι οποίοι εκπαιδεύτηκαν πάνω σε δύο σύνολα σημαντικών παραμέτρων που προέκυψαν είτε μέσω μεθόδων επιλογής χαρακτηριστικών είτε μέσω του πίνακα συσχετίσεων. Με βάση την εκπαίδευσή τους πάνω σε αυτά τα δύο σύνολα παραμέτρων, οι αλγόριθμοι κατηγοριοποίησης έκαναν προβλέψεις για νέα δεδομένα πλοίων που εισέρχονταν στην ναυτιλιακή εταιρία. Για αυτά τα δύο σύνολα μεταβλητών που χρησιμοποιήθηκαν, έγινε σύγκριση μεταξύ των τεχνικών κατηγοριοποίησης για το ποια τεχνική κατηγοριοποιούσε με περισσότερη ακρίβεια και πιο αξιόπιστα τα πλοία σε ρυπογόνα και μη ρυπογόνα. Την καλύτερη απόδοση στο σύνολο μεταβλητών που προέκυψαν από τις διάφορες μεθόδους επιλογής χαρακτηριστικών, την είχαν οι αλγόριθμοι κατηγοριοποίησης «Logistic Regression», «Support Vector Machines», «Random Forest Classifier» και ο αλγόριθμος «Decision Trees». Η ακρίβεια (accuracy) των παραπάνω τεχνικών κατηγοριοποίησης ήταν ίση με 97,25%, 97,42%, 97,25%, 96,56% και 86,96% αντίστοιχα. Επίσης το F-Score των παραπάνω αλγορίθμων για το συγκεκριμένο σύνολο βρέθηκε ίσο με 97,25%, 97,42%, 97,25%, 96,56% και 86,92% αντίστοιχα. Καλύτερος και πιο χρήσιμος θεωρήθηκε ο αλγόριθμος κατηγοριοποίησης «Logistic Regression» επειδή γενικά η λογιστική παλινδρόμηση είναι πιο απλή στην εφαρμογή και η ερμηνεία των αποτελεσμάτων που προκύπτουν είναι πολύ πιο εύκολη. Για το σύνολο μεταβλητών που προέκυψε από τον πίνακα συσχετίσεων και τις τιμές του δείκτη VIF σε περιπτώσεις μεταβλητών που εμφάνιζαν υψηλή συσχέτιση, ο καλύτερος αλγόριθμος κατηγοριοποίησης των δεδομένων ήταν ο «Random Forest Classifier» με ποσοστό ακρίβειας 72,04% και ποσοστό F-Score ίσο με 72,25%. Ικανοποιητική απόδοση εμφάνισε και ο αλγόριθμος «Decision Trees» με ποσοστό ακρίβειας ίσο με 66,72% και αντίστοιχο F-Score ίσο με 66,72%. Το μειονέκτημα του αλγορίθμου «Random Forest» είναι ότι χρειάζεται 0,4437 δευτερόλεπτα για να εκτελεστεί ενώ οι υπόλοιποι αλγόριθμοι χρειάζονται το ένα τρίτο του χρόνου του αλγορίθμου κατηγοριοποίησης «Random Forest».

Η τρίτη και τελευταία μελέτη είχε στόχο την εύρεση προτύπων και σχέσεων μέσα στο σύνολο δεδομένων, του οποίου οι μεταβλητές εμφάνιζαν υψηλή συσχέτιση μεταξύ τους. Για τον λόγο αυτό εφαρμόστηκε η μέθοδος των κύριων συνιστωσών και ιεραρχική συσταδοποίηση. Αρχικά χρησιμοποιήθηκε η μέθοδος κύριων συνιστωσών για την μείωση διαστάσεων του προβλήματος και αντικατάσταση των αρχικών μεταβλητών με ένα άλλο μικρότερο σύνολο ασυσχέτιστων μεταβλητών τις συνιστώσες, οι οποίες προκύπτουν ως ένας γραμμικός συνδυασμός των αρχικών μεταβλητών. Αφού έγινε αρχικά τυποποίηση των τιμών των μεταβλητών και εφαρμόσθηκε η μέθοδος PCA, παρατηρήθηκε ότι οι τρεις κύριες συνιστώσες

εξηγούσαν μαζί το 79,26% της συνολικής μεταβλητότητας των αρχικών δεδομένων οπότε αυτές χρησιμοποιήθηκαν για την συσταδοποίηση. Εντοπίστηκαν υψηλά φορτία μεταξύ των μεταβλητών που αφορούσαν τις διαστάσεις του πλοίου και της πρώτης κύρια συνιστώσας, ενώ εμφανίστηκαν υψηλά φορτία μεταξύ περιβαλλοντικής απόδοσης των πλοίων και της δεύτερης κύριας συνιστώσας. Τέλος η τρίτη κύρια συνιστώσα είχε υψηλές τιμές φορτίων με μεταβλητές που αφορούσαν την περιβαλλοντική απόδοση του πλοίου ανά περιοχή. Έπειτα εφαρμόστηκε ιεραρχική συσταδοποίηση με βάση τις τρεις κύριες συνιστώσες όμως οι ακραίες παρατηρήσεις που δεν είχαν απομακρυνθεί από το σύνολο δεδομένων φαίνεται να επηρέαζαν τον αριθμό των συστάδων και έτσι δεν υπήρχε ξεκάθαρα κάποιος κατάλληλος αριθμός συστάδων. Οπότε εφαρμόστηκε πάλι η ανάλυση αφού είχαν απομακρυνθεί οι ακραίες παρατηρήσεις. Οι τρεις πρώτες κύριες συνιστώσες και πάλι εξηγούσαν το 80% της συνολικής μεταβλητότητας των αρχικών δεδομένων και εμφάνιζαν υψηλά φορτία με τις ίδιες μεταβλητές όπως και στην περίπτωση της ανάλυσης πριν την απομάκρυνση των ακραίων παρατηρήσεων. Οι δείκτες «Silhouette coefficient», «Davies – Bouldin» και «Calinski – Harabasz» έδειξαν ότι ο κατάλληλος αριθμός συστάδων ήταν οι τέσσερις συστάδες. Εφαρμόζοντας λοιπόν συσταδοποίηση για αριθμό συστάδων ίσο με 4, προέκυψε το συμπέρασμα ότι στην δημιουργία συστάδων συνείσφεραν περισσότερο οι πρώτες δύο κύριες συνιστώσες που αφορούσαν τις διαστάσεις των πλοίων και την περιβαλλοντική τους απόδοση αντίστοιχα. Τελικά στην πρώτη συστάδα εντοπίστηκαν 716 πλοία, στην δεύτερη συστάδα βρέθηκαν 280 πλοία, στην τρίτη συστάδα υπήρχαν 624 πλοία ενώ στην τέταρτη συστάδα υπήρχαν 169 πλοία. Βλέποντας τα περιγραφικά μέτρα των μεταβλητών ανά συστάδα, αποδόθηκαν κάποιες ετικέτες ονομάτων στις συστάδες. Η πρώτη συστάδα αποτελούταν από μικρών διαστάσεων πλοία τα οποία ήταν φιλικά προς το περιβάλλον, η δεύτερη συστάδα αποτελούταν από μεγάλων διαστάσεων πλοία που δεν ήταν φιλικά προς το περιβάλλον, η τρίτη συστάδα περιείχε μεγάλων διαστάσεων πλοία που ήταν φιλικά προς το περιβάλλον ενώ η τέταρτη συστάδα περιείχε μικρών διαστάσεων πλοία που δεν ήταν φιλικά προς το περιβάλλον.

Με γνώμονα τις παραπάνω μελέτες, φαίνεται η σημαντικότητα της ανάλυσης δεδομένων και της στατιστικής μηχανικής μάθησης στον κλάδο της ναυτιλίας και της πολύτιμης γνώσης που προκύπτει από την επεξεργασία και ανάλυση πολυπληθών δεδομένων. Πλέον η εφαρμογή μεθόδων στατιστικής μηχανικής μάθησης έχει βοηθήσει δραματικά τις ναυτιλιακές εταιρίες στην λήψη καλύτερων, αξιόπιστων και πιο αποδοτικών αποφάσεων σχετικά με την διαχείριση των πλοίων και φυσικά έχει επιφέρει πολλά κέρδη και οφέλη στον χώρο της ναυτιλίας.

7. Βιβλιογραφία

Ξένη

(2014). Στο M. Zaki, & W. Meira, *Data Mining and Analysis* (σσ. 577-602).

Abebe, M., Shin, Y., Noh, Y., Lee, S., & Lee, I. (2020). Machine Learning Approaches for Ship Speed Prediction towards Energy Efficient Shipping.

All about shipping. (χ.χ.). Ανάκτηση από https://allaboutshipping.co.uk/2022/03/24/greek-controlled-shipping-2022/

Analytics Vidhya. (χ.χ.). Ανάκτηση από https://www.analyticsvidhya.com/blog/2021/08/decision-tree-algorithm/

Analytics Vidhya. (χ.χ.). Ανάκτηση από https://www.analyticsvidhya.com/blog/2021/06/understanding-random-forest/

Analytics Vidhya. (χ.χ.). Ανάκτηση από https://www.analyticsvidhya.com/blog/2021/09/gradient-boosting-algorithm-a-complete-guide-for-beginners/

Analytics Vidhya. (χ.χ.). Ανάκτηση από https://www.analyticsvidhya.com/blog/2020/10/feature-selection-techniques-in-machine-learning/

Aznar, P. (2020). *QuantDare*. Ανάκτηση από https://quantdare.com/what-is-the-difference-between-extra-trees-and-random-forest/

Bassam, A., Phillips, A., Turnock, S., & Wilson, P. (2022). Ship speed prediction based on machine learning for efficient shipping operation.

Budu, E. (2023). *Baeldung*. Ανάκτηση από https://www.baeldung.com/cs/random-forest-vs-extremely-randomized-trees

Cruise Activities in MedCruise Ports 2017 statistics. (2017). Ανάκτηση από cruise_activities_in_medcruise_ports-statistics_2017_final%20(1).pdf

DataCamp. (2022). Ανάκτηση από https://www.datacamp.com/tutorial/tutorial-lasso-ridge-regression

Discussion: Subset Selection, Ridge Regression and the Lasso. (2001). Στο T. Hastie, R. Tibshinari, & J. Friedman, *The Elements of Statistical Learning* (σσ. 69-73).

DNV. (χ.χ.). Ανάκτηση από https://www.dnv.com/maritime/hub/decarbonize-shipping/key-drivers/regulations/imo-regulations/ghg-vision.html

E nautilia. (χ.χ.). Ανάκτηση από https://e-nautilia.gr/katigories-kai-eidi-ploiwn/.

E nautilia. (2022). Ανάκτηση από https://e-nautilia.gr/oi-7-shmantikoteroi-thalassioi-diadromoi-sto-kosmo/

Hu, Z., Jin, Y., Hu, Q., Sen, S., Zhou, T., & Osman, M. (2019). Prediction of Fuel Consumption for Enroute Ship Based on Machine Learning.

Hwang, T., & Youn, I.-H. (2021). Navigation Situation Clustering Model of Human-Operated Ships for Maritime Autonomous Surface Ship Collision Avoidance Tests.

Ibna Zaman, K. P. (2017). Challenges and Opportunities of Big Data Analytics for Upcoming Regulations and Future Transformation of the Shipping Industry.

Jaadi, Z. (2023). *built in*. Ανάκτηση από https://builtin.com/data-science/step-step-explanation-principal-component-analysis

K means algorithm. (2014). Στο M. Zaki, & W. Meira, *Data Mining and Analysis* (σσ. 357-365).

King, A. (2022). *European Commission*. Ανάκτηση από https://ec.europa.eu/research-and-innovation/en/horizon-magazine/emissions-free-sailing-full-steam-ahead-ocean-going-shipping

Lloyd's List. (χ.χ.). Ανάκτηση από https://lloydslist.maritimeintelligence.informa.com/one-hundred-container-ports-2020/port-data

Lyman Ott, M. T. (2015). *An Introduction to Statistical Methods and Data Analysis.*

Marine Digital. (χ.χ.). Ανάκτηση από https://marine-digital.com/article_top_9_sensors_on_the_ship

Masui, T. (2022). *TowardsDatascience*. Ανάκτηση από https://towardsdatascience.com/all-you-need-to-know-about-gradient-boosting-algorithm-part-1-regression-2520a34a502

Michael W. Berry, A. M. (2019). *Supervised and Unsupervised Learning for Data Science.*

Mohammad Hossein Moradi, M. B. (2022). *Marine route optimization using reinforcement learning approach to reduce fuel consumption and consequently minimize CO2 emissions.*

Mohammed J. Zaki, W. M. (2014). *Data Mining and Analysis.*

Niu, H., Ozanich, E., & Gerstoft, P. (2017). Ship localization in Santa Barbara Channel using machine learning classifiers.

Olivier Chapelle, B. S. (2006). *Semi - Supervised Learning.*

Pierre Geurts, D. E. (2006). Extremely randomized trees.

Rawson, A., Brito, M., Sabeur, Z., & Tran-Thanh, L. (2021). A machine learning approach for monitoring ship safety in extreme weather events.

Richard A. Johnson, D. W. (2007). *Applied Multivariate Statistical Analysis.*

scikit - kearn. (χ.χ.). Ανάκτηση από https://scikit-learn.org/stable/

scikit - learn. (χ.χ.). Ανάκτηση από https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.ExtraTreesRegressor.html

scikit - learn. (χ.χ.). Ανάκτηση από https://scikit-learn.org/stable/modules/naive_bayes.html

simplilearn. (χ.χ.). Ανάκτηση από https://www.simplilearn.com/normalization-vs-standardization-article

Skarlatos, K., Fousteris, A., Bersimis, S., Georgakellos, D., & Economou, P. (2023). Assessing Ships' Environmental Performance Using Machine Learning.

StatLect. (2021). Ανάκτηση από https://www.statlect.com/fundamentals-of-statistics/ridge-regression

Su, Z., Wu, C., & He, H. (2022). Study on the prediction model of accidents and incidents of cruise ship operation based on machine learning.

Subramaniam, G., & Mahmoud, M. (2021). Fraud Detection in Shipping Industry using K-NN Algorithm.

Support Vector Machines. (2014). Στο M. Zaki, & W. Meira, *Data Mining and Analysis* (σσ. 543-563).

The geography of trasnport system. (χ.χ.). Ανάκτηση από https://transportgeography.org/contents/chapter5/maritime-transportation/vessel-size-groups/

TowardsDatascience. (2018). Ανάκτηση από https://towardsdatascience.com/linear-regression-using-gradient-descent-97a6c8700931

Trevor Hastie, R. T. (2001). *The Elements of Statistical Learning.*

Tsaganos, G., Papachristos, D., Nikitakos, N., Dalaklis, D., & Olcer, A. (2018). Fault Detection and Diagnosis of Two-Stroke Low-Speed Marine Engine with Machine Learning Algorithms.

Turing. (χ.χ.). Ανάκτηση από https://www.turing.com/kb/guide-to-principal-component-analysis

tutorials point. (χ.χ.). Ανάκτηση από https://www.tutorialspoint.com/what-is-attribute-selection-measures

UNCTAD. (2021). Ανάκτηση από https://unctad.org/system/files/official-document/rmt2021_en_0.pdf

Uyanik, T., Karatug, C., & Arslanoglu, Y. (2020). Machine learning approach to ship fuel consumption: A case of container vessel.

Viellechner, A., & Spinler, S. (2020). Novel Data Analytics Meets Conventional Container Shipping: Predicting Delays by Comparing Various Machine Learning Algorithms.

World Shipping Council. (χ.χ.). Ανάκτηση από https://www.worldshipping.org/top-50-ports

Zou, H., & Hastie, T. (2005). Regularization and Variable Selection via the Elastic Net pp. 301-320.

Ελληνική

Γεώργιος Δεμερούτης, Δημήτριος Μυλωνόπουλος. (2010). *Ναυτιλιακές γνώσεις.*

Δημήτριος Μυλωνόπουλος, Γ. (χ.χ.). Κατάταξη πλοίων. Στο *Ναυτιλιακές γνώσεις* (σσ. 14-29).

Εισαγωγή στην Ridge Regression. (2021). Στο Σ. Μπερσίμης, Α. Σαχλάς, Γ. Μπάρτζης, & Γ. Παπαδάκης, *Εφαρμοσμένη στατιστική και στατιστική μηχανική μάθηση* (σσ. 435-447).

Εισαγωγή στην μέθοδο Nearest Neighbor. (2021). Στο Σ. Μπερσίμης, Α. Σαχλάς, Γ. Μπάρτζης, & Γ. Παπαδάκης, *Εφαρμοσμένη στατιστική και στατιστική μηχανική μάθηση* (σ. 571).

Ένωση Εφοπλιστών Ελλάδος. (2022). Ανάκτηση από https://www.ugs.gr/gr/greek-shipping-and-economy/greek-shipping-and-economy-2022/characteristics-of-the-greek-owned-fleet/

Ζυγομαλάς, Ν. (2021). *Μεταφορά Φορτίων.* Αθήνα.

Ζυγομαλάς, Ν. (χ.χ.). Πλοία μεταφοράς χύδην ξηρών φορτίων. Στο *Μεταφορά φορτίων* (σσ. 59-60).

Κούτρας, Μ., & Ευαγγελάρας, Χ. (2010). *Ανάλυση Παλινδρόμησης.*

Λυκούδης, Π. (1988). Τα ναυτιλιακά έγγραφα του πλοίου. Στο *Στοιχεία ναυτικού δικαίου* (σσ. 27-31).

Ναυτικό μουσείο Αιγαίου. (χ.χ.). Ανάκτηση από Ναυτικό μουσείο Αιγαίου: https://aegean-maritime-museum.gr/el/ekthemata-kai-istoria/synthiki-tou-kioutsoyk-kainartzi/127-synthiki-kioutsouk-kainartzi-2,

ΟΛΠ. (χ.χ.). Ανάκτηση από https://www.olp.gr/en/

Παναγιώτης, Λ. (1988). *Στοιχεία ναυτικού δικαίου.*

Πολέμης, Σ. (χ.χ.). The history of greek shipping.

Σάγος, Γ. (2019). *Εισαγωγή στην Υδροακουστική & στην Τεχνολογία Sonar.*

Στατιστική μηχανική μάθηση. (2021). Στο Σ. Μπερσίμης, Α. Σαχλάς, Γ. Μπάρτζης, & Γ. Παπαδάκης, *Εφαρμοσμένη στατιστική και στατιστική μηχανική μάθηση* (σσ. 19-20).

Σωτήρης Μπερσίμης, Αθανάσιος Σαχλάς (2016). *Εφαρμοσμένη στατιστική με έμφαση στις επιστήμες υγείας.*

Σωτήρης Μπερσίμης, Αθανάσιος Σαχλάς (2021). *Εφαρμοσμένη στατιστική και στατιστική μηχανική μάθηση.*

Παραρτήματα

Πηγαίος κώδικας σε Python

```
# importing our libraries

import pandas as pd
import numpy as np
import re #for regular expression
import matplotlib.pyplot as plt #Visualization
import seaborn as sns # Better visualization
import warnings # Ignore warnings
warnings.filterwarnings('ignore')

# importing our data
data = pd.read_csv(r'C:\Users\Administrator\Desktop\Thesis Data\Ship Index\Ship Index\FINAL_THESIS_DATA_2021.csv')
print(data.info())



## Exploratory data analysis -- Visualization ##


# Looking for duplicates indices and print the total sum of duplicates
duplicates = (data.index[data.duplicated()])
print(data.duplicated().sum())

# Searching for missing values
print(data.isna().sum())

# Dropping rows missing values of column 'Distance'
data.dropna(subset=['Distance'],inplace=True)
print(data.info())


# sort by column 'CO2 emissions' our dataframe to see the values if there are any zeros
data_sorted = data.sort_values(by='Total CO2 emissions [m tonnes]')



# Descriptive statistics for all our variables
print(data.describe())
```

```
# Calculating descriptive measures for our 4 variables
data_descr = data[['Total fuel consumption [m tonnes]','Total CO2 emissions [m tonnes]','Distance','ves_dwt']]
stats = data_descr.describe()
from scipy.stats import trim_mean
#trimmed mean, range, IQR
stats.loc['IQR'] = stats.loc["75%"]-stats.loc["25%"] #Interquartile range
stats.loc["trimmed"] = trim_mean(data_descr,0.1) #trimmed mean
stats.loc['range'] = stats.loc['max']-stats.loc['min'] #range
stats = stats.append(data_descr.agg(['skew','kurt','sem'])) #by using 'append' function we add to stats the measures 'skewness',
# 'kurtosis' and standard error of mean



## Visualization - Normality Tests ##

# A condition to separate ship based on their deadweight and save them in a new column - - visualization purposes only
ship_dwt_category = []

for dwt in data['ves_dwt']:
  if (dwt <= 10000):
    ship_dwt_category.append('MINI HANDY')
  elif (dwt >= 10000.001) and (dwt <= 35000):
    ship_dwt_category.append('HANDYSIZE')
  elif (dwt >= 35000.001) and (dwt <= 50000 ):
    ship_dwt_category.append('HANDYMAX')
  elif (dwt >= 50000.001) and (dwt <= 60000):
    ship_dwt_category.append('SUPRAMAX')
  elif (dwt >= 60000.001) and (dwt <=80000):
    ship_dwt_category.append('PANAMAX')
  elif (dwt >= 80000.001 ) and (dwt <= 100000):
    ship_dwt_category.append('POST PANAMAX')
  elif (dwt >= 100000.001 ) and (dwt <= 200000):
    ship_dwt_category.append('CAPESIZE')
  else:
    ship_dwt_category.append('VLBC')

data['ves_dwt_category'] = ship_dwt_category



# Barplot of value counts of column ves_dwt_category
data.ves_dwt_category.value_counts().plot(kind='bar',color=['blue','orange','red','green','yellow','pink','purple','brown'])
```

```
plt.title("Barplot of vessels' categories")
plt.xlabel('Categories of vessels')
plt.ylabel('Deadweight')

# value counts
print(data.ves_dwt_category.value_counts())


#Density plots for some of our "ves_dwt" variable

# Create density plot
sns.kdeplot(data['ves_dwt'], shade=True, color='red')

# Add axis labels and plot title
sns.set_style("darkgrid")
sns.set_palette("icefire")
sns.set(font_scale=1.2)
plt.xlabel("Deadweight")
plt.ylabel("Density")
plt.title("Density Plot of deadweight")
plt.show()



#histogram subplots for our continuous variables

data_con = data.drop(columns=['Ship type','Reporting Period','ves_dwt_category'])

# Using regex to rename columns - - crossing out '[....]' just for visualizing purposes
data_con = data_con.rename(columns=lambda x: re.sub('(.*)(?:\s\[).*',r'\1',x))



# Choosing two subests for a better and clear view of subplots
data_con_1 = data_con.loc[:,'Total fuel consumption':'CO2 emissions from all voyages which departed from ports under a MS jurisdiction']

data_con_2 = data_con.loc[:,'Annual average Fuel consumption per transport work (mass)':'ves_main_engine_kw']

data_con_3 = data_con.loc[:,'CO2 emissions from all voyages to ports under a MS jurisdiction':'Annual average Fuel consumption per distance']


histogram_1 = data_con_1.hist(figsize=(22,16),color='orange')
histogram_2 = data_con_2.hist(figsize=(22,16),color='green')
histogram_3 = data_con_3.hist(figsize=(22,16),color='purple')
```

```
# Kolmogorov smirnov test for Normality
from scipy.stats import kstest



#perform Kolmogorov-Smirnov test for all continuous variables

for i in data_con:
 print('Kolmogorov - Smirnov Test for:',{i})
 print(kstest(data_con[i], 'norm'))
 print("-----------------------------")



# Outliers detection -- through IQR Range

df_outlier = data.drop(columns=['Ship type', 'Reporting Period','Annual average Fuel consumption per transport work (mass) [g / m tonnes Â· n miles]',
                    'Annual average CO2 emissions per transport work (mass) [g CO2 / m tonnes Â· n miles]'])

Q1 = df_outlier.quantile(0.25)
Q3 = df_outlier.quantile(0.75)
IQR = Q3 - Q1


outliers_per_variable = ((df_outlier < (Q1 - 1.5 * IQR)) | (df_outlier > (Q3 + 1.5 * IQR))).sum()
print(outliers_per_variable)


#using the library matplotlib.pyplot for data visualization

# boxplot for variable 'ves_dwt'
dwt_box = data.boxplot(column = 'ves_dwt',figsize=(8,4),color='Red')
dwt_box.plot()
plt.title('Boxplot for Deadweight')
plt.show()


# boxplot for variable 'ves_depth'
depth_box = data.boxplot(column = 'ves_depth',figsize=(8,4),color='green')
depth_box.plot()
plt.title('Boxplot for depth of the ship')
plt.show()
```

```
# boxplot for variable 'ves_loa'

loa_box = data.boxplot(column = 'ves_loa', figsize=(8,4), color='orange')
loa_box.plot()
plt.title('Boxplot for the Length of the ship')
plt.show()


# boxplot for variable 'ves_draft'

draft_box = data.boxplot(column = 'ves_draft', figsize=(8,4), color='blue')
draft_box.plot()
plt.title('Boxplot for the draft of the ship')
plt.show()


# boxplot for variable 'ves_beam'

beam_box = data.boxplot(column = 'ves_beam', figsize=(8,4), color='brown')
beam_box.plot()
plt.title('Boxplot for the beam of the ship')
plt.show()

# boxplot for variable 'Distance'

distance_box = data.boxplot(column = 'Distance', figsize=(8,4), color='blue')
distance_box.plot()
plt.title('Boxplot for the distance that ship has travelled')
plt.show()


# boxplot for variable 'ves_capacity_grain'

grain_box = data.boxplot(column = 'ves_capacity_grain', figsize=(8,4), color='pink')
grain_box.plot()
plt.title('Boxplot for grain capacity')
plt.show()



# boxplot for variable 'ves_main_engine_kw'

engine_box = data.boxplot(column = 'ves_main_engine_kw', figsize=(8,4), color='purple')
engine_box.plot()
plt.title('Boxplot for main engine kw')
plt.show()
```

```
# Visualization of outliers for the other variables of dataset

data_box = data_con.drop(columns=['ves_main_engine_kw','ves_capacity_grain','Distance','ves_beam','ves_draft','ves_loa','ves_depth','ves_dwt'])
data_box.plot(figsize=(24,18),kind='box')
plt.show()



# Correlation matrix with spearman
from scipy.stats import spearmanr

cormat=data_con.corr(method='spearman')
f,ax=plt.subplots(figsize=(18,14))
sns.heatmap(cormat,square=True,annot=True,cmap="PuBu")
plt.show()


#Save each pair with correlations in a list named 'high_cor' and


# Saving the pairs with correlation > 0.7 to find if their correlations are significantly important
c = data_con.corr().abs()
s = c.unstack()
high_cor = s.sort_values(kind="quicksort")
high_cor = pd.DataFrame(high_cor)
high_cor.rename(columns={0:'cor_value'}, inplace=True)
highest_corr = high_cor[(high_cor['cor_value'] >0.7) & (high_cor['cor_value']<1)]


# Scatterplot for Total consumption and Total CO2 emissions
data.plot.scatter(x='Total fuel consumption [m tonnes]',y= 'Total CO2 emissions [m tonnes]')


# A function that calculates spearman coefficient p-value for each pair of variables
def spear_pvalues(data_con):
   cols = pd.DataFrame(columns=data_con.columns)
   p = cols.transpose().join(cols, how='outer')
   for i in data_con.columns:
      for j in data_con.columns:
         tmp = data_con[data_con[i].notnull() & data_con[j].notnull()]
         p[i][j] = round(spearmanr(tmp[i], tmp[j])[1], 4)

   return p
```

```
p_val = spear_pvalues(data_con)
print(p_val)



# VIF calculating to solve a possible multicollinearity problem

#dropping two uneccassary columns
data_con = data_con.drop(columns=['Annual average Fuel consumption per transport work (mass)','Annual average CO2 emissions per transport work (mass)'])

# using scikit-learn - Performing a linear model for predicting CO2 emissions to see the coefficients of independent variables for
# their relationship with CO2 emissions

from sklearn import linear_model
reg = linear_model.LinearRegression()


y = data_con['Total CO2 emissions']
X = data_con.drop(columns=['Total CO2 emissions'])


# full regression this time using statsmodels
import statsmodels.api as sm

model = sm.OLS(y,X)
results = model.fit()
print(results.summary())


print('----------First Regression model VIF values----------')

# VIF values for the above model
from statsmodels.stats.outliers_influence import variance_inflation_factor

VIF = pd.DataFrame()
VIF['feature']  = X.columns
VIF['VIF'] = [variance_inflation_factor(X.values, i) for i in range(X.shape[1])]

print(VIF)
print('----------End of trial 1----------')


# dropping high correlated variables
```

```
data_con_clear = data_con.drop(columns=['CO2 emissions from all voyages between ports under a MS jurisdiction','ves_main_engine_kw','Annual Total time spent at sea','Annual average Fuel consumption per distance'])

#Performing again multiple linear regression model to see the coefficients of independent variables

from sklearn import linear_model
reg = linear_model.LinearRegression()

# dropping two columns that have nan values and are not important in our analysis
#data_con = data_con.drop(columns=['Annual average Fuel consumption per transport work (mass)','Annual average CO2 emissions per transport work (mass)'])


y = data_con_clear['Total CO2 emissions']
X = data_con_clear.drop(columns=['Total CO2 emissions'])


# full regression this time using statsmodels
import statsmodels.api as sm

model_2  = sm.OLS(y,X)
results = model_2.fit()
print(results.summary())

print('----------Second regression model VIF values----------')
# See the VIF values again

VIF = pd.DataFrame()
VIF['feature']  = X.columns
VIF['VIF'] = [variance_inflation_factor(X.values, i) for i in range(X.shape[1])]

print(VIF)
print('----------End of trial 2----------')


# Dropping again more columns
data_con_clear.drop(columns=['CO2 emissions from all voyages which departed from ports under a MS jurisdiction','Annual average CO2 emissions per distance',
                             'ves_draft','ves_beam','ves_depth','ves_loa','Total time spent at sea','ves_capacity_grain','Total fuel consumption'],inplace=True)


# Performing again multiple linear regression model

reg_3 = linear_model.LinearRegression()
```

```
y = data_con_clear['Total CO2 emissions']
X = data_con_clear.drop(columns=['Total CO2 emissions'])


# full regression this time using statsmodels
import statsmodels.api as sm

model_3   = sm.OLS(y,X)
results = model_3.fit()
print(results.summary())


print('----------VIF values for third time!!----------')



VIF = pd.DataFrame()
VIF['feature']  = X.columns
VIF['VIF'] = [variance_inflation_factor(X.values, i) for i in range(X.shape[1])]

print(VIF)
print('----------End of trial 3----------')



print('----------Time to fit the final model according to low VIF variables----------')

# Our final model's significance through anova F test
import statsmodels.formula.api as smf

# keep the columns for the final model and rename them to avoid 'errors'
data_model = data_con[['Total CO2 emissions','Technical efficiency','CO2 emissions which occurred within ports under a MS jurisdiction at berth','Distance','ves_dwt','CO2 emissions from all voyages to ports under a MS jurisdiction']]
data_model.rename(columns = {'Total CO2 emissions':'CO2_emissions','Technical efficiency':'Tech_eff','CO2 emissions which occurred within ports under a MS jurisdiction at berth':'CO2_emis_MS_berth','CO2 emissions from all voyages to ports under a MS jurisdiction':'CO2_emis_from_MS_to'},inplace=True)


model_final = smf.ols(formula ='CO2_emissions ~ Tech_eff + CO2_emis_from_MS_to + CO2_emis_MS_berth + Distance + ves_dwt', data = data_model)
results = model_final.fit()
print(results.summary())

#Anova F test
```

```
table = sm.stats.anova_lm(results)
print(table)



######## Calculations #########

### Calculate our new variables in order create a target variable named "ves_pollution"

#We are using all of our initial dataset for this purpose to insert in it our newly created variabes

# Setting up our parameters for later use

Z_2023 = 5  # 2023 reduction factor
Z_2024 = 7  # 2024 reduction factor
Z_2025 = 9  # 2025 reduction factor
Z_2026 = 11

# Two parameters used to calculate CII reference year
a = 4745
c = 0.622


# 4 vectors that categorize our vessels in categories A,B,C,D,E depending on their CII values

d1 = 0.86
d2 = 0.94
d3 = 1.06
d4 = 1.18

#  Calculating new variable "Attained_CII" which is the attained CII for year 2021.

# Equation: Attained CII = [Total CO2 emissions /(Deadweight * Distance)] * 10^6

data['Attained_CII'] = (data['Total CO2 emissions [m tonnes]'] / (data['ves_dwt'] * data['Distance']))
* (10**6)
print(data['Attained_CII'].head(10))


# Calculating new variable "CII_ref"

# Equation: a * Deadweight ^ -c

data['CII_ref'] = a * data['ves_dwt'] ** -c
print(data['CII_ref'].head(10))
```

```
# Calculating new variable "Required_CII" which is the Required CII for years 2023,2024,2025,2026.

# Equation: Required CII_2023 = CII ref * ((100 - Z)/100)

# 2023 Required CII
data['Required_CII_2023'] = data['CII_ref'] * ((100 - Z_2023)/100)

# 2024 Required CII
data['Required_CII_2024'] = data['CII_ref'] * ((100 - Z_2024)/100)

# 2025 Required CII
data['Required_CII_2025'] = data['CII_ref'] * ((100 - Z_2025)/100)

# 2026 Required CII
data['Required_CII_2026'] = data['CII_ref'] * ((100 - Z_2026)/100)


# Calculating Attained CII / Required CII for each year of 2023, 2024, 2025, 2026 to identify through 4 vectors
# the category of ships' CO2 emissions A,B,C,D or E

# CII Rating for 2023
data['CII_Rate_score_2023'] = data['Attained_CII'] / data['Required_CII_2023']

# CII Rating for 2024
data['CII_Rate_score_2024'] = data['Attained_CII'] / data['Required_CII_2024']

# CII Rating for 2025
data['CII_Rate_score_2025'] = data['Attained_CII'] / data['Required_CII_2025']

# CII Rating for 2026
data['CII_Rate_score_2026'] = data['Attained_CII'] / data['Required_CII_2026']



# According to values and the vectors' range, we categorize the ships in 5 categories of pollution for each years (2023,2024,2025,2026)

# For 2023

ship_CII_category_2023 = []

for val in data['CII_Rate_score_2023']:
  if (val < d1):
    ship_CII_category_2023.append('A')
  elif (val >= d1) and (val < d2):
```

```
        ship_CII_category_2023.append('B')
    elif (val >= d2 ) and (val < d3 ):
        ship_CII_category_2023.append('C')
    elif (val >= d3 ) and (val < d4 ):
        ship_CII_category_2023.append('D')
    else:
        ship_CII_category_2023.append('E')

data['CII_Rate_Class_2023'] = ship_CII_category_2023


# For 2024

ship_CII_category_2024 = []

for val in data['CII_Rate_score_2024']:
    if (val < d1):
        ship_CII_category_2024.append('A')
    elif (val >= d1) and (val < d2):
        ship_CII_category_2024.append('B')
    elif (val >= d2 ) and (val < d3 ):
        ship_CII_category_2024.append('C')
    elif (val >= d3 ) and (val < d4 ):
        ship_CII_category_2024.append('D')
    else:
        ship_CII_category_2024.append('E')

data['CII_Rate_Class_2024'] = ship_CII_category_2024


# For 2025

ship_CII_category_2025 = []

for val in data['CII_Rate_score_2025']:
    if (val < d1):
        ship_CII_category_2025.append('A')
    elif (val >= d1) and (val < d2):
        ship_CII_category_2025.append('B')
    elif (val >= d2 ) and (val < d3 ):
        ship_CII_category_2025.append('C')
    elif (val >= d3 ) and (val < d4 ):
        ship_CII_category_2025.append('D')
    else:
        ship_CII_category_2025.append('E')

data['CII_Rate_Class_2025'] = ship_CII_category_2025
```

```
# For 2026

ship_CII_category_2026 = []

for val in data['CII_Rate_score_2026']:
   if (val < d1):
      ship_CII_category_2026.append('A')
   elif (val >= d1) and (val < d2):
      ship_CII_category_2026.append('B')
   elif (val >= d2 ) and (val < d3 ):
      ship_CII_category_2026.append('C')
   elif (val >= d3 ) and (val < d4 ):
      ship_CII_category_2026.append('D')
   else:
      ship_CII_category_2026.append('E')

data['CII_Rate_Class_2026'] = ship_CII_category_2026
```

```
# Creation of our discrete target variable 'ves_pollution'


# A function that finds if there is a 'D' rating in three consecutive years or an 'E' rating in a single year
```

```
def probabilities(con):
  if (con['CII_Rate_Class_2023'] == 'D') and (con['CII_Rate_Class_2024'] == 'D') and (con['CII_Rate_Class_2025'] == 'D'):
    return 1
  elif (con['CII_Rate_Class_2024'] == 'D') and (con['CII_Rate_Class_2025'] == 'D') and (con['CII_Rate_Class_2026'] == 'D'):
    return 1
  elif (con['CII_Rate_Class_2023'] == 'D') and (con['CII_Rate_Class_2025'] == 'D') and (con['CII_Rate_Class_2026'] == 'D'):
    return 1
  elif (con['CII_Rate_Class_2023'] == 'D') and (con['CII_Rate_Class_2024'] == 'D') and (con['CII_Rate_Class_2026'] == 'D'):
    return 1
  elif (con['CII_Rate_Class_2023'] == 'E') or (con['CII_Rate_Class_2024'] == 'E') or (con['CII_Rate_Class_2025'] == 'E') or (con['CII_Rate_Class_2026'] == 'E'):
    return 1
  else:
    return 0


# Perform the function above in our dataset
data['ves_pollution'] = data.apply(probabilities, axis=1)


# value counts for each year's CII Ratings
print(data['CII_Rate_Class_2026'].value_counts())


# Donut plots for CII Ratings 2023 - 2024

# Try this code on jupyter notebook
import plotly.graph_objects as go
from plotly.subplots import make_subplots

labels = ["A", "B", "C", "D", "E"]

colors = ['blue','green','yellow','orange','red']

# Create subplots
specs = [[{'type':'domain'}, {'type':'domain'}], [{'type':'domain'}, {'type':'domain'}]]
fig = make_subplots(rows=2, cols=2, specs=specs)


fig.add_trace(go.Pie(labels=labels, values=[181, 314,592 , 520, 335], name='CII Ratings for 2023',
              marker_colors=colors), 1, 1)
fig.add_trace(go.Pie(labels=labels, values=[137,267,587 ,536 , 415], name='CII Ratings for 2024',
              marker_colors=colors), 1, 2)
fig.add_trace(go.Pie(labels=labels, values=[105,214 ,565 ,544 ,514 ], name='CII Ratings for 2025',
```

```
            marker_colors=colors), 2, 1)
fig.add_trace(go.Pie(labels=labels, values=[80,175,505,554,628 ], name='CII Ratings for 2026',
            marker_colors=colors), 2, 2)


fig.update_traces(hoverinfo='label+percent+name', textinfo='none',hole=.3)
fig.update(layout_title_text='CII Ratings for years 2023-2026 with reference the year 2021',
      layout_showlegend=False)

fig = go.Figure(fig)
fig.show()


# Percentages of ships that are in danger for gas emissions and not

# Create a labeled variable just for visualization purpose ---  991 in 'Non polluting' and 951 in 'Pollutant'
data['ves_pollution_labeled'] = data['ves_pollution'].map({0:'Non polluting', 1:'Pollutant'})
print(data['ves_pollution'].value_counts())


# This part of code for donuts plots -- jupyter
labels = ['Non Pollutant','Pollutant']
values = [991, 951]
colors = ['green','red']
# Use `hole` to create a donut-like pie chart
fig = go.Figure(data=[go.Pie(labels=labels,marker_colors = colors, values=values, hole=.3)])
fig.show()



# Test to check if there is diffience in Total CO2 emissions between
data_test = data[['Total CO2 emissions [m tonnes]','ves_pollution_labeled']]

# Calculating the mean and std for two categories
print(data_test.groupby('ves_pollution_labeled').agg(['mean','std']).round(2))

# We will perform Shapiro Wilk normality test for 2 populations
from scipy.stats import shapiro
r=data_test.groupby('ves_pollution_labeled').agg(shapiro) #fit the shapiro wilk check of normality (1st column "W', 2ND COLUMN= P-VALUE
r=pd.DataFrame(r['Total CO2 emissions [m tonnes]'].tolist(),index=r.index,columns=['W','P']) #it creates a data frame with two columns, one for
# W statistic and one for the p-value
r=r.drop(columns=['W']).round(3) #we drop from our new created data frame the "W statistic of shapiro wilk"
print(r)
```

```
# Non parametric test for two independent samples -- Mann Whitney
from scipy.stats import mannwhitneyu
pollutant = data_test[data_test['ves_pollution_labeled']=='Pollutant'] #go to data where parents are ill
non_pol = data_test[data_test['ves_pollution_labeled']=='Non polluting'] #go to data fram named data, where parents column is "healthy"
print(mannwhitneyu(pollutant['Total CO2 emissions [m tonnes]'],non_pol['Total CO2 emissions [m tonnes]'],alternative='two-sided'))




# 2nd Part - Classification of data in two categories (Pollutant or Non-polluting) ships


data_class = data.drop(columns=['Ship type','Reporting Period','ves_dwt_category','Attained_CII','CII_ref','Required_CII_2023',

'Required_CII_2024','Required_CII_2025','Required_CII_2026','CII_Rate_score_2023','CII_Rate_score_2024',

'CII_Rate_score_2025','CII_Rate_score_2026','CII_Rate_Class_2023','CII_Rate_Class_2024',
                    'CII_Rate_Class_2025','CII_Rate_Class_2026','ves_pollution_labeled',
                    'Annual average Fuel consumption per transport work (mass) [g / m tonnes Â· n miles]',
                    'Annual average CO2 emissions per transport work (mass) [g CO2 / m tonnes Â· n miles]'])


print(data_class.info())

# Feature selection -- Extra trees classifier

print('--------------------FIRST FEATURE SELECTION METHOD --- EXTRA TREES CLASSIFIER---------- ')
# input variables
X = data_class.drop(columns=['ves_pollution'])  #independent columns

# output variable
y = data_class.loc[:,"ves_pollution"]    #target column NSP with 2 categories
#importing Extra Trees Classifier
from sklearn.ensemble import ExtraTreesClassifier
import matplotlib.pyplot as plt

model = ExtraTreesClassifier()
model.fit(X,y)
print(model.feature_importances_) #use inbuilt class feature_importances of tree based classifiers
```

```
#plot graph of feature importances for better visualization
feat_importances = pd.Series(model.feature_importances_, index=X.columns)
feat_importances.nlargest(11).plot(kind='barh')
plt.show()



print('--------------------SECOND FEATURE SELECTION METHOD --- SELECT K BEST----------------------
-------- ')

# Not so reliable results cause the variables should be categorical to perfrom chi^2 test

# Feature Selection - - Sekect K best based on chi^2 scores

#importing our neccessary libraries
from sklearn.feature_selection import SelectKBest
from sklearn.feature_selection import chi2


X = data_class.drop(columns=['ves_pollution','Total fuel consumption [m tonnes]','CO2 emissions
from all voyages which departed from ports under a MS jurisdiction [m tonnes]',
                    'CO2 emissions from all voyages between ports under a MS jurisdiction [m
tonnes]','Annual Total time spent at sea [hours]',
                    'Annual average Fuel consumption per distance [kg / n mile]','Total time spent at sea
[hours]',
                    'ves_capacity_grain'])  #independent columns
y = data_class.loc[:,'ves_pollution'] #dependentcolumn
#apply SelectKBest class to extract top 10 best features
bestfeatures = SelectKBest(score_func=chi2, k=11)
fit = bestfeatures.fit(X,y)
dfscores = pd.DataFrame(fit.scores_)
dfcolumns = pd.DataFrame(X.columns)
#concat two dataframes for better visualization
featureScores = pd.concat([dfcolumns,dfscores],axis=1)
featureScores.columns = ['Variables','Score']  #naming the dataframe columns
print(featureScores.nlargest(11,'Score'))  #print 11 best features



print('------------THIRD FEATURE SELECTION METHOD --- LASSO REGRESSION')

# Lasso regression feature selection
import sklearn
from sklearn.linear_model import LinearRegression, Lasso #importing Lasso
from mlxtend.feature_selection import SequentialFeatureSelector
from sklearn.model_selection import train_test_split
```

```
#select input variables
X = data_class.drop(columns=['ves_pollution','Total fuel consumption [m tonnes]','CO2 emissions from all voyages which departed from ports under a MS jurisdiction [m tonnes]',
                    'CO2 emissions from all voyages between ports under a MS jurisdiction [m tonnes]','Annual Total time spent at sea [hours]',
                    'Annual average Fuel consumption per distance [kg / n mile]','Total time spent at sea [hours]',
                    'ves_capacity_grain'])
#select target variable
y = data_class.ves_pollution


sfs_selector = SequentialFeatureSelector(Lasso(), k_features = 9, cv =0)
sfs_selector2 = SequentialFeatureSelector(Lasso(), k_features = 9, cv =5)

sfs1 = sfs_selector.fit(X, y)
sfs1pdf = pd.DataFrame(sfs1.subsets_)
sfs1pdf = sfs1pdf.T
sfs1pdf = sfs1pdf.drop(columns = ['feature_idx'])
cv_scores = [np.round(num, 2) for num in sfs1pdf.cv_scores]
avg_score = [np.round(num, 2) for num in sfs1pdf.avg_score]
sfs1pdf.cv_scores = cv_scores
sfs1pdf.avg_score = avg_score
print('\nLasso Regression:\n',sfs1pdf)



print('------Fourth feature selection method----------Logistic regression with lasso reguralization')

from sklearn.preprocessing import StandardScaler
from sklearn.feature_selection import SelectFromModel
from sklearn.linear_model import Lasso, LogisticRegression
from sklearn.model_selection import train_test_split
# input and output space
X = data_class.drop(columns=['ves_pollution'])
y = data_class.ves_pollution

# Split train and test set
X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=34)

#scaling data
sc = StandardScaler()
sc.fit(X_train)
```

```
# select features utilizing logistic regression as a classifier, with the Lasso regularization
sel_ = SelectFromModel(
   LogisticRegression(C=0.5, penalty='l1', solver='liblinear', random_state=34))
sel_.fit(sc.transform(X_train), y_train)

# Features that need to be removed
removed_feats = X_train.columns[(sel_.estimator_.coef_ == 0).ravel().tolist()]
print(removed_feats)

# the dataset with only the selected features from this method
data_lassr = data_class.drop(columns=['Total fuel consumption [m tonnes]', 'Total CO2 emissions [m tonnes]',
     'CO2 emissions from all voyages which departed from ports under a MS jurisdiction [m tonnes]',
     'Annual Total time spent at sea [hours]',
     'Total time spent at sea [hours]', 'Distance', 'ves_capacity_grain','ves_pollution'])

print(data_lassr.columns)


print('------Final feature selection method----------------Random forest classifier feature selection----------')

from sklearn.ensemble import RandomForestClassifier

model = RandomForestClassifier(n_estimators=100)

# input and output space
X = data_class.drop(columns=['ves_pollution'])
y = data_class.ves_pollution

# fitting model
model.fit(X,y)


# the importance of resultiung features
importances = model.feature_importances_

# a dataframe for visualization
final_df = pd.DataFrame({'Features':pd.DataFrame(X).columns, 'Importances':importances})
final_df.set_index('Importances')

# sort in ascending order for better visualization
final_df = final_df.sort_values('Importances')

#plot in bars the importances
final_df.plot.bar(color = 'purple')
```

```
print(final_df)

# Top features are 'Annual average CO2 emissions per distance', 'Annual average fuel cons per distnce','ves_dwt','ves_capacity_grain',
# 'Technical Efficiency', 'ves_depth', 'ves_main_engine_kw', 'ves_loa', 'ves_draft', 'distance', 'ves_beam'

print('----------CLASSIFICATION PROBLEM----------')


# FINAL DATAFRAME WITH ALL THE SELECTED VARIABLES FOR CLASSIFICATION

data_class = data[['Annual average CO2 emissions per distance [kg CO2 / n mile]','Annual average Fuel consumption per distance [kg / n mile]','CO2 emissions from all voyages between ports under a MS jurisdiction [m tonnes]',
        'Technical efficiency','Distance','ves_dwt','ves_loa','ves_draft','ves_depth',
        'ves_main_engine_kw','ves_capacity_grain','ves_pollution']]

# This dataset includes only the variables that had low VIF values to avoid multicollinearity of data
data_class_2 = data[['Technical efficiency','CO2 emissions from all voyages between ports under a MS jurisdiction [m tonnes]',
            'ves_dwt','Distance','CO2 emissions which occurred within ports under a MS jurisdiction at berth [m tonnes]',
            'ves_pollution']]

print('----------Fitting classification models without model selection----------')

#importing packages
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

from sklearn.preprocessing import StandardScaler #standard sclaing of data
from sklearn.naive_bayes import GaussianNB #Gaussian Naive Bayes
from sklearn.linear_model import LogisticRegression #Logistic Regression
from sklearn.ensemble import RandomForestClassifier #Random Forrest Classification
from sklearn.neighbors import KNeighborsClassifier
from sklearn.svm import LinearSVC #Linear Support Vector Classifier
from sklearn.tree import DecisionTreeClassifier # Import Decision Tree Classifier
from sklearn.model_selection import train_test_split #for model training/test
from sklearn.metrics import accuracy_score  #evaluate classification with Accuracy
from sklearn.metrics import confusion_matrix #to construct the confusion matrix
from sklearn.metrics import precision_recall_fscore_support #other evaluation metrics (recall, F score, precision)

import seaborn as sns #better visualization
```

```
# GAUSSIAN NAIVE BAYES

print('---------------------- 1st --- GAUSSIAN NAIVE BAYES-------------------------')
df = data_class.drop(columns=['ves_pollution']) #change data_class to data_class_2 to view the results for reduced VIF model


import time #to estimate the time needed to execute the algorithm

start = time.time()

target = data_class['ves_pollution'] #target variable --change data_class to data_class_2 for the other dataset features
y = pd.DataFrame(target).to_numpy().ravel() #is used to change a 2-dimensional array or a multi-dimensional array into a contiguous flattened array

X = pd.DataFrame(df).to_numpy() #common numpy dtype for all variables
X = StandardScaler().fit_transform(X) #standard scaling

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state = 34) #seperating train/test set

gaus = GaussianNB()


gaus.fit(X_train, y_train)
y_pred = gaus.predict(X_test)

#creaing the confusion matrix
cm = confusion_matrix(y_test, y_pred)
print(cm)

print('Accuracy = ' + str(gaus.score(X_test,y_test))) #print us the 'Accuracy'

#evaluation with other metrics
gaus_metrics = precision_recall_fscore_support(y_test, y_pred, average='macro')
gaus_metrics_labels = ['Precision','Recall', 'F-score','Support']

for i in range(0,len(gaus_metrics)):
   print(gaus_metrics_labels[i] + ' = ' + str(gaus_metrics[i]))


ax= plt.subplot()
sns.heatmap(cm, annot=True, fmt='g', ax = ax); #annot=True to annotate cells

# labels, title and ticks
ax.set_xlabel('Predicted labels');ax.set_ylabel('True labels');
ax.set_title('Confusion Matrix');
```

```
ax.xaxis.set_ticklabels(['Non polluting', 'Pollutant']);
ax.yaxis.set_ticklabels(['Non polluting', 'Pollutant']);

plt.show()

#end time
end = time.time()

# print the difference between start
# and end time in secs
# get the execution time
elapsed_time = end - start
print('Execution time:', elapsed_time, 'seconds')



# Logistic regression classification

print('---------------------- 2nd --- Logistic regression------------------------')



import time #to estimate the time needed to execute the algorithm

start = time.time()


target = data_class['ves_pollution'] #target variable --change data_class to data_class_2 for the other dataset features
y = pd.DataFrame(target).to_numpy().ravel() #is used to change a 2-dimensional array or a multi-dimensional array into a contiguous flattened array

X = pd.DataFrame(df).to_numpy() #common numpy dtype for all variables
X = StandardScaler().fit_transform(X) #standard scaling

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state = 34) #seperating train/test set

log_r = LogisticRegression()


log_r.fit(X_train, y_train)
y_pred = log_r.predict(X_test)

#creaing the confusion matrix
cm = confusion_matrix(y_test, y_pred)
print(cm)
```

```
print('Accuracy = ' + str(log_r.score(X_test,y_test))) #print us the 'Accuracy'

#evaluation with other metrics
log_r_metrics = precision_recall_fscore_support(y_test, y_pred, average='macro')
log_r_metrics_labels = ['Precision','Recall', 'F-score', 'Support']

for i in range(0,len(gaus_metrics)):
   print(log_r_metrics_labels[i] + ' = ' + str(log_r_metrics[i]))


ax= plt.subplot()
sns.heatmap(cm, annot=True, fmt='g', ax = ax); #annot=True to annotate cells

# labels, title and ticks
ax.set_xlabel('Predicted labels');ax.set_ylabel('True labels');
ax.set_title('Confusion Matrix');
ax.xaxis.set_ticklabels(['Non polluting','Pollutant']);
ax.yaxis.set_ticklabels(['Non polluting', 'Pollutant']);

plt.show()


#end time
end = time.time()

# print the difference between start
# and end time in secs
# get the execution time
elapsed_time = end - start
print('Execution time:', elapsed_time, 'seconds')

# better results with standard scaler ->accuracy = 0,97
# min max -> accuracy = 0,94


# Support vector machines classification

print('---------------------- 3rd --- Support vector Machines-------------------------')


import time

start = time.time()

target = data_class['ves_pollution'] #target variable --change data_class to data_class_2 for the other dataset features
y = pd.DataFrame(target).to_numpy().ravel() #is used to change a 2-dimensional array or a multi-dimensional array into a contiguous flattened array
```

```
X = pd.DataFrame(df).to_numpy() #common numpy dtype for all variables
X = StandardScaler().fit_transform(X) #standard scaling

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state = 34) #seperating train/test set

lin_svc = LinearSVC(C=1.0, max_iter=1000) #its default number of itterations,  C = reguralization parameter default = 1.0

lin_svc.fit(X_train, y_train)
y_pred = lin_svc.predict(X_test)

#creaing the confusion matrix
cm = confusion_matrix(y_test, y_pred)
print(cm)

print('Accuracy = ' + str(lin_svc.score(X_test,y_test))) #print us the 'Accuracy' for predicted CLASS (we want the number as string)

#evaluation with other metrics
lin_svc_metrics = precision_recall_fscore_support(y_test, y_pred, average='macro')
lin_svc_metrics_labels = ['Precision','Recall', 'F-score', 'Support']

for i in range(0,len(lin_svc_metrics)):
   print(lin_svc_metrics_labels[i] + ' = ' + str(lin_svc_metrics[i]))


ax= plt.subplot()
sns.heatmap(cm, annot=True, fmt='g', ax = ax); #annot=True to annotate cells

# labels, title and ticks
ax.set_xlabel('Predicted labels');ax.set_ylabel('True labels');
ax.set_title('Confusion Matrix');
ax.xaxis.set_ticklabels(['Non polluting','Pollutant']);
ax.yaxis.set_ticklabels(['Non polluting','Pollutant']);

plt.show()


#end time
end = time.time()

# print the difference between start
# and end time in secs
# get the execution time
elapsed_time = end - start
print('Execution time:', elapsed_time, 'seconds')
```

```
# Random forest classification

print('---------------------- 4th --- Random Forest-------------------------')


import time

start = time.time()

target = data_class['ves_pollution'] #target variable --change data_class to data_class_2 for the other dataset features
y = pd.DataFrame(target).to_numpy().ravel() #is used to change a 2-dimensional array or a multi-dimensional array into a contiguous flattened array

X = pd.DataFrame(df).to_numpy() #common numpy dtype for all variables
X = StandardScaler().fit_transform(X) #standard scaling

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state = 34) #seperating train/test set


clf = RandomForestClassifier()

clf.fit(X_train, y_train)
y_pred = clf.predict(X_test)

#creaing the confusion matrix
cm = confusion_matrix(y_test, y_pred)
print(cm)

print('Accuracy = ' + str(clf.score(X_test,y_test))) #print us the 'Accuracy' for predicted CLASS (we want the number as string)

#evaluation with other metrics
clf_metrics = precision_recall_fscore_support(y_test, y_pred, average='macro')
clf_metrics_labels = ['Precision','Recall', 'F-score', 'Support']

for i in range(0,len(clf_metrics)):
   print(clf_metrics_labels[i] + ' = ' + str(clf_metrics[i]))

#plot the heatmap for confusion matrix
ax= plt.subplot()
sns.heatmap(cm, annot=True, fmt='g', ax = ax); #annot=True to annotate cells

# labels, title and ticks
```

```
ax.set_xlabel('Predicted labels');ax.set_ylabel('True labels');
ax.set_title('Confusion Matrix');
ax.xaxis.set_ticklabels(['Non polluting','Pollutant']);
ax.yaxis.set_ticklabels(['Non polluting','Pollutant']);

plt.show()

#end time
end = time.time()

# print the difference between start
# and end time in secs
# get the execution time
elapsed_time = end - start
print('Execution time:', elapsed_time, 'seconds')


# Decision tree classification

print('---------------------- 5th --- Decision trees-------------------------')


import time

start = time.time()

target = data_class['ves_pollution'] #target variable --change data_class to data_class_2 for the other dataset features
y =  pd.DataFrame(target).to_numpy().ravel() #is used to change a 2-dimensional array or a multi-dimensional array into a contiguous flattened array

X = pd.DataFrame(df).to_numpy() #common numpy dtype for all variables
X = StandardScaler().fit_transform(X) #standard scaling

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state = 34) #seperating train/test set

# Create Decision Tree classifer object
trees =  DecisionTreeClassifier()

# Train Decision Tree Classifer
trees.fit(X_train,y_train)

#Predict the response for test dataset
y_pred = trees.predict(X_test)

#creaing the confusion matrix
cm = confusion_matrix(y_test, y_pred)
```

```
print(cm)

print('Accuracy = ' + str(trees.score(X_test,y_test))) #print us the 'Accuracy' for predicted CLASS (we want the number as string)

#evaluation with other metrics
trees_metrics = precision_recall_fscore_support(y_test, y_pred, average='macro')
trees_metrics_labels = ['Precision','Recall', 'F-score', 'Support']

for i in range(0,len(trees_metrics)):
   print(trees_metrics_labels[i] + ' = ' + str(trees_metrics[i]))

#plot the heatmap for confusion matrix
ax= plt.subplot()
sns.heatmap(cm, annot=True, fmt='g', ax = ax); #annot=True to annotate cells

# labels, title and ticks
ax.set_xlabel('Predicted labels');ax.set_ylabel('True labels');
ax.set_title('Confusion Matrix');
ax.xaxis.set_ticklabels(['Non polluting','Pollutant']);
ax.yaxis.set_ticklabels(['Non polluting','Pollutant']);

plt.show()

#end time
end = time.time()

# print the difference between start
# and end time in secs
# get the execution time
elapsed_time = end - start
print('Execution time:', elapsed_time, 'seconds')

#
# KNN classification

print('---------------------- 6h --- K Nearest neighbors-------------------------')


import time

start = time.time()

target = data_class['ves_pollution'] #target variable --change data_class to data_class_2 for the other dataset features
y =  pd.DataFrame(target).to_numpy().ravel() #is used to change a 2-dimensional array or a multi-dimensional array into a contiguous flattened array
```

```
X = pd.DataFrame(df).to_numpy() #common numpy dtype for all variables
X = StandardScaler().fit_transform(X) #standard scaling

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state = 34) #seperating train/test set

# Create Decision Tree classifer object
knear =  KNeighborsClassifier(n_neighbors=2) #tuning this parameter

# Train Decision Tree Classifer
knear.fit(X_train,y_train)

#Predict the response for test dataset
y_pred = knear.predict(X_test)

#creaing the confusion matrix
cm = confusion_matrix(y_test, y_pred)
print(cm)

print('Accuracy = ' + str(knear.score(X_test,y_test))) #print us the 'Accuracy' for predicted CLASS (we want the number as string)

#evaluation with other metrics
knear_metrics = precision_recall_fscore_support(y_test, y_pred, average='macro')
knear_metrics_labels = ['Precision','Recall', 'F-score', 'Support']

for i in range(0,len(knear_metrics)):
   print(knear_metrics_labels[i] + ' = ' + str(knear_metrics[i]))

#plot the heatmap for confusion matrix
ax= plt.subplot()
sns.heatmap(cm, annot=True, fmt='g', ax = ax); #annot=True to annotate cells

# labels, title and ticks
ax.set_xlabel('Predicted labels');ax.set_ylabel('True labels');
ax.set_title('Confusion Matrix');
ax.xaxis.set_ticklabels(['Non polluting','Pollutant']);
ax.yaxis.set_ticklabels(['Non polluting','Pollutant']);

plt.show()

#end time
end = time.time()

# print the difference between start
# and end time in secs
# get the execution time
elapsed_time = end - start
```

```
print('Execution time:', elapsed_time, 'seconds')



print('----------PCA & CLUSTERING----------')


df = data.drop(columns=['Ship type','Reporting Period','ves_dwt_category','Attained_CII','CII_ref','Required_CII_2023',

'Required_CII_2024','Required_CII_2025','Required_CII_2026','CII_Rate_score_2023','CII_Rate_score_2024',

'CII_Rate_score_2025','CII_Rate_score_2026','CII_Rate_Class_2023','CII_Rate_Class_2024',
                'CII_Rate_Class_2025','CII_Rate_Class_2026','ves_pollution_labeled',
                'Annual average Fuel consumption per transport work (mass) [g / m tonnes Â· n miles]',
                'Annual average CO2 emissions per transport work (mass) [g CO2 / m tonnes Â· n miles]','ves_pollution'])

# Using regex to rename columns - - crossing out '[....]' just for visualizing purposes
df = df.rename(columns=lambda x: re.sub('(.*)(?:\s\[).*',r'\1',x))

print(df.info())


# Standardization of dataset and PCA

#Standardization
from sklearn.preprocessing import StandardScaler
# define standard scaler
X_std = StandardScaler()
# transform data
X_std = X_std.fit_transform(df)
print(X_std)


#Principal Component analysis for reduction of dataset -1st method for Min Max scaled DATA
from sklearn.decomposition import PCA

pca = PCA()#PCA model
Xstd_pca = pca.fit_transform(X_std)#transform and fit the data to pca model

#print(pca.components_.T)
pcs = pd.DataFrame(pca.components_.T)#traspose pca components and insert to dataframe - we rotate the components ->more variance explained
columns = []
```

```
#auto create columns based the number of PC's
for i in range(len(pca.components_)):
   columns.append('PC'+str(i+1))
pcs.columns =  columns
pcs.index = df.columns#use the data columns as index in the new dataframe pcs
print(pcs)


# Plot scree plot and see eigenvalues to conclude in how many pcs we should keep

explained_variance = pd.DataFrame()
index = []
#create an index for each PC in dataframe
for i in range(len(pca.explained_variance_)):
   index.append('PC'+str(i+1))
explained_variance['Eigenvalues'] = pca.explained_variance_#import explained variance to
dataframe column  #explained_variance = diagonal elements of PC's Covariance matrix
explained_variance.index = index
print(explained_variance)
#plot the explained variance
plt.plot(pca.explained_variance_)
plt.xlabel('number of components')
plt.ylabel('Eigenvalues');
plt.xticks(np.arange(len(df.columns)), ('1', '2', '3', '4', '5',
'6','7','8','9','10','11','12','13','14','15','16','17','18','19'))
plt.show()
#plot the cumulative explained variance
plt.plot(np.cumsum(pca.explained_variance_ratio_))
plt.xlabel('number of components')
plt.ylabel('cumulative explained variance ratio')
plt.xticks(np.arange(len(df.columns)), ('1', '2', '3', '4', '5',
'6','7','8','9','10','11','12','13','14','15','16','17','18','19'))
plt.show()


# Calculating the eigenvalues again, with the total percentage of variance explained of each PC
cov_mat = np.cov(X_std.T)

# From this covariance matrix, caluclate the Eigenvalues and the Eigenvectors
eigen_vals, eigen_vecs = np.linalg.eig(cov_mat)

# print the Eigenvalues
print("Raw Eigenvalues: \n", eigen_vals)
# the sum of the Eigenvalues
print("Percentage of Variance Explained by Each Component: \n", eigen_vals/sum(eigen_vals))
```

```
#Loadings and biplot


# Scatter points in biplot
pca = PCA(n_components=2)

PC_scores = pd.DataFrame(pca.fit_transform(X_std),
          columns = ['PC1', 'PC2'])
print(PC_scores.head(10))

#Loadings in a dataframe for first 2 PCs

loadings = pd.DataFrame(pca.components_.T, columns=['PC1', 'PC2'],
                index=df.columns)
print(loadings)

# Data of biplot

PC1 = pca.fit_transform(X_std)[:,0]
PC2 = pca.fit_transform(X_std)[:,1]
ldngs = pca.components_

#We should also define the scaling factors scalePC1 and scalePC2 to fit the data of PC1, PC2 and ldngs on the same plot.
# Additionally, we can define the target group names to label the target groups in our biplots, see features below

#renaming columns names for better visualization plots
df.columns                                                                                                    =
['Z1','Z2','Z3','Z4','Z5','Z6','Z7','Z8','Z9','Z10','Z11','Z12','Z13','Z14','Z15','Z16','Z17','Z18','Z19']


scalePC1 = 1.0/(PC1.max() - PC1.min())
scalePC2 = 1.0/(PC2.max() - PC2.min())
features = df.columns


#Biplot

fig, ax = plt.subplots(figsize=(14, 9))

for i, feature in enumerate(features):
   ax.arrow(0, 0, ldngs[0, i],
         ldngs[1, i],
         head_width=0.03,
         head_length=0.03,
         color="red")
```

```
    ax.text(ldngs[0, i] * 1.15,
            ldngs[1, i] * 1.15,
            feature,color="purple", fontsize=18)

    ax.scatter(PC1 * scalePC1,
                PC2 * scalePC2, s=5)

for i, label in enumerate(PC_scores.index):
    ax.text(PC1[i] * scalePC1,
            PC2[i] * scalePC2, str(label),
            fontsize=10)

ax.set_xlabel('PC1', fontsize=20)
ax.set_ylabel('PC2', fontsize=20)
ax.set_title('Figure 2', fontsize=20)


# LOADINGS TABLE OF PCA for the first 3 PCs

pca = PCA(n_components=3)

PC_scores = pd.DataFrame(pca.fit_transform(X_std),
             columns = ['PC1', 'PC2','PC3'])
print(PC_scores.head(10))

#Loadings in a dataframe for first 2 PCs

loadings = pd.DataFrame(pca.components_.T, columns=['PC1', 'PC2','PC3'],
                    index=df.columns)
print(loadings)

# Data of biplot

PC1 = pca.fit_transform(X_std)[:,0]
PC2 = pca.fit_transform(X_std)[:,1]
PC3 = pca.fit_transform(X_std)[:,2]
ldngs = pca.components_



print('----------HIERARCHICAL CLUSTERING----------')

#importing our neccessary libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.decomposition import PCA #dimensionality reduction - better view of our plots
from sklearn.cluster import AgglomerativeClustering #for agglomerative clustering
```

```
from sklearn.preprocessing import StandardScaler #standard scaling
from sklearn.metrics import silhouette_score  #our scoring coefficient
import scipy.cluster.hierarchy as shc

#our full dataset

#Standardization
from sklearn.preprocessing import StandardScaler
# define standard scaler
X_std = StandardScaler()
# transform data
X_std = X_std.fit_transform(df)
print(X_std)

pca = PCA(n_components = 3) #We will keep the first 3 principal components
X_principal = pca.fit_transform(X_std)
X_principal = pd.DataFrame(X_principal)
X_principal.columns = ['PC1','PC2','PC3']


# Ward dendrogram
plt.figure(figsize =(8, 8))
plt.title('Visualising the full dataset')
Dendrogram = shc.dendrogram((shc.linkage(X_principal, method ='ward')))


#furthest neighbor dendrogram

plt.figure(figsize =(8, 8))
plt.title('Visualising the full dataset')
Dendrogram = shc.dendrogram((shc.linkage(X_principal, method ='complete')))


print('------------------CLUSTERING FOR PC1 & PC2-------------------------')
# CLUSTERING FOR PC1 AND PC2

ac2 = AgglomerativeClustering(linkage="ward", affinity="euclidean", n_clusters = 2) #for 2
clusters

# Visualizing the clustering
plt.figure(figsize =(6, 6))
plt.scatter(X_principal['PC1'], X_principal['PC2'],
        c = ac2.fit_predict(X_principal), cmap ='flare')
plt.ylabel('Operating enviromental performance over time and distance')
plt.xlabel('Dimensions of the ship')
plt.show()
```

```
ac3 = AgglomerativeClustering(linkage="ward", affinity="euclidean", n_clusters = 3)

plt.figure(figsize =(6, 6))
plt.scatter(X_principal['PC1'], X_principal['PC2'],
        c = ac3.fit_predict(X_principal), cmap ='flare')
plt.ylabel('Operating enviromental performance over time and distance')
plt.xlabel('Dimensions of the ship')
plt.show()



ac4 = AgglomerativeClustering(linkage="ward", affinity="euclidean", n_clusters = 4)

plt.figure(figsize =(6, 6))
plt.scatter(X_principal['PC1'], X_principal['PC2'],
        c = ac4.fit_predict(X_principal), cmap ='flare')
plt.ylabel('Operating enviromental performance over time and distance')
plt.xlabel('Dimensions of the ship')
plt.show()


ac55 = AgglomerativeClustering(linkage="ward", affinity="euclidean", n_clusters = 5)

plt.figure(figsize =(6, 6))
plt.scatter(X_principal['PC1'], X_principal['PC2'],
        c = ac55.fit_predict(X_principal), cmap ='flare')
plt.ylabel('Operating enviromental performance over time and distance')
plt.xlabel('Dimensions of the ship')
plt.show()



ac66 = AgglomerativeClustering(linkage="ward", affinity="euclidean", n_clusters = 6)

plt.figure(figsize =(6, 6))
plt.scatter(X_principal['PC1'], X_principal['PC2'],
        c = ac66.fit_predict(X_principal), cmap ='flare')
plt.ylabel('Operating enviromental performance over time and distance')
plt.xlabel('Dimensions of the ship')
plt.show()


# Davies bouldind and silhouette coefficient plots for the appropriate number of clusters for PC1 and PC2

#silhouette scores for number of clusters
k = [2, 3, 4, 5, 6]
```

```
# Appending the silhouette scores of the different models to the list
silhouette_scores = []
silhouette_scores.append(
    silhouette_score(X_principal, ac2.fit_predict(X_principal)))
silhouette_scores.append(
    silhouette_score(X_principal, ac3.fit_predict(X_principal)))
silhouette_scores.append(
    silhouette_score(X_principal, ac4.fit_predict(X_principal)))
silhouette_scores.append(
    silhouette_score(X_principal, ac55.fit_predict(X_principal)))
silhouette_scores.append(
    silhouette_score(X_principal, ac66.fit_predict(X_principal)))

# Plotting a bar graph to compare the results
plt.bar(k, silhouette_scores)
plt.xlabel('Number of clusters', fontsize = 20)
plt.ylabel('Silhouette Coef.', fontsize = 20)



from sklearn.metrics import davies_bouldin_score
#davies bouldin scores for number of clusters
k = [2, 3, 4, 5, 6]

# Appending the silhouette scores of the different models to the list
davies_bouldin_scores = []
davies_bouldin_scores.append(
    davies_bouldin_score(X_principal, ac2.fit_predict(X_principal)))
davies_bouldin_scores.append(
    davies_bouldin_score(X_principal, ac3.fit_predict(X_principal)))
davies_bouldin_scores.append(
    davies_bouldin_score(X_principal, ac4.fit_predict(X_principal)))
davies_bouldin_scores.append(
    davies_bouldin_score(X_principal, ac55.fit_predict(X_principal)))
davies_bouldin_scores.append(
    davies_bouldin_score(X_principal, ac66.fit_predict(X_principal)))

# Plotting a bar graph to compare the results
plt.bar(k, davies_bouldin_scores,color='orange')
plt.xlabel('Number of clusters', fontsize = 20)
plt.ylabel('Davies - Bouldin', fontsize = 20)
plt.show()

from sklearn import metrics
print('---------the exact values of evaluation indexes for each pair and cluster----------')

# exact calculation of silhouette coefficient for the number of clusters we chose
db2 = AgglomerativeClustering(linkage= 'ward', affinity='euclidean',n_clusters=2).fit(X_principal)
```

```
labels = db2.labels_
print("Silhouette Coefficient for 2 clusters: %0.3f" % metrics.silhouette_score(X_principal, labels))

db3 = AgglomerativeClustering(linkage= 'ward', affinity='euclidean',n_clusters=3).fit(X_principal)
labels = db3.labels_
print("Silhouette Coefficient for 3 clusters: %0.3f" % metrics.silhouette_score(X_principal, labels))

db4 = AgglomerativeClustering(linkage= 'ward', affinity='euclidean',n_clusters=4).fit(X_principal)
labels = db4.labels_
print("Silhouette Coefficient for 4 clusters: %0.3f" % metrics.silhouette_score(X_principal, labels))

db55 = AgglomerativeClustering(linkage= 'ward',
affinity='euclidean',n_clusters=5).fit(X_principal)
labels = db55.labels_
print("Silhouette Coefficient for 5 clusters: %0.3f" % metrics.silhouette_score(X_principal, labels))

db66 = AgglomerativeClustering(linkage= 'ward',
affinity='euclidean',n_clusters=6).fit(X_principal)
labels = db66.labels_
print("Silhouette Coefficient for 6 clusters: %0.3f" % metrics.silhouette_score(X_principal, labels))



# Davies bouldin

db22 = AgglomerativeClustering(linkage= 'ward',
affinity='euclidean',n_clusters=2).fit(X_principal)
labels = db22.labels_
print("Davies Bouldin for 2 clusters: %0.3f" % metrics.davies_bouldin_score(X_principal, labels))

db32 = AgglomerativeClustering(linkage= 'ward',
affinity='euclidean',n_clusters=3).fit(X_principal)
labels = db32.labels_
print("Davies Bouldin for 3 clusters: %0.3f" % metrics.davies_bouldin_score(X_principal, labels))

db42 = AgglomerativeClustering(linkage= 'ward',
affinity='euclidean',n_clusters=4).fit(X_principal)
labels = db42.labels_
print("Davies Bouldin for 4 clusters: %0.3f" % metrics.davies_bouldin_score(X_principal, labels))

db52 = AgglomerativeClustering(linkage= 'ward',
affinity='euclidean',n_clusters=5).fit(X_principal)
labels = db52.labels_
print("Davies Bouldin for 5 clusters: %0.3f" % metrics.davies_bouldin_score(X_principal, labels))

db62 = AgglomerativeClustering(linkage= 'ward',
affinity='euclidean',n_clusters=6).fit(X_principal)
labels = db62.labels_
```

```
print("Davies Bouldin for 6 clusters: %0.3f" % metrics.davies_bouldin_score(X_principal, labels))



# Calinski Harabasz Score

db23 = AgglomerativeClustering(linkage= 'ward', affinity='euclidean',n_clusters=2).fit(X_principal)
labels = db23.labels_
print("Calinski Harabasz for 2 clusters: %0.3f" % metrics.calinski_harabasz_score(X_principal, labels))

db33 = AgglomerativeClustering(linkage= 'ward', affinity='euclidean',n_clusters=3).fit(X_principal)
labels = db33.labels_
print("Calinski Harabasz for 3 clusters: %0.3f" % metrics.calinski_harabasz_score(X_principal, labels))

db43 = AgglomerativeClustering(linkage= 'ward', affinity='euclidean',n_clusters=4).fit(X_principal)
labels = db43.labels_
print("Calinski Harabasz for 4 clusters: %0.3f" % metrics.calinski_harabasz_score(X_principal, labels))

db53 = AgglomerativeClustering(linkage= 'ward', affinity='euclidean',n_clusters=5).fit(X_principal)
labels = db53.labels_
print("Calinski Harabasz for 5 clusters: %0.3f" % metrics.calinski_harabasz_score(X_principal, labels))

db63 = AgglomerativeClustering(linkage= 'ward', affinity='euclidean',n_clusters=6).fit(X_principal)
labels = db63.labels_
print("Calinski Harabasz for 6 clusters: %0.3f" % metrics.calinski_harabasz_score(X_principal, labels))



print('----------------------SECOND PAIR OF PCs   PC1 & PC3--------------------')
# Clustering for PC1 and PC3

ac5 = AgglomerativeClustering(linkage="ward", affinity="euclidean", n_clusters = 2) #for 2 clusters

# Visualizing the clustering
plt.figure(figsize =(6, 6))
plt.scatter(X_principal['PC1'], X_principal['PC3'],
```

```
        c = ac5.fit_predict(X_principal), cmap ='rainbow')
plt.ylabel('Operating enviromental performance over region')
plt.xlabel('Dimensions of the ship')
plt.show()


ac6 = AgglomerativeClustering(linkage="ward", affinity="euclidean", n_clusters = 3)

plt.figure(figsize =(6, 6))
plt.scatter(X_principal['PC1'], X_principal['PC3'],
        c = ac6.fit_predict(X_principal), cmap ='rainbow')
plt.ylabel('Operating enviromental performance over region')
plt.xlabel('Dimensions of the ship')
plt.show()



ac7 = AgglomerativeClustering(linkage="ward", affinity="euclidean", n_clusters = 4)

plt.figure(figsize =(6, 6))
plt.scatter(X_principal['PC1'], X_principal['PC3'],
        c = ac7.fit_predict(X_principal), cmap ='rainbow')
plt.ylabel('Operating enviromental performance over region')
plt.xlabel('Dimensions of the ship')
plt.show()


ac77 = AgglomerativeClustering(linkage="ward", affinity="euclidean", n_clusters = 5)

plt.figure(figsize =(6, 6))
plt.scatter(X_principal['PC1'], X_principal['PC3'],
        c = ac77.fit_predict(X_principal), cmap ='rainbow')
plt.ylabel('Operating enviromental performance over region')
plt.xlabel('Dimensions of the ship')
plt.show()



ac78 = AgglomerativeClustering(linkage="ward", affinity="euclidean", n_clusters = 6)

plt.figure(figsize =(6, 6))
plt.scatter(X_principal['PC1'], X_principal['PC3'],
        c = ac78.fit_predict(X_principal), cmap ='rainbow')
plt.ylabel('Operating enviromental performance over region')
plt.xlabel('Dimensions of the ship')
plt.show()
```

```
print('-----CLUSTERING FOR PC2 & PC3----------')
# Clustering for PC2 and PC3

ac8 = AgglomerativeClustering(linkage="ward", affinity="euclidean", n_clusters = 2) #for 2 clusters

# Visualizing the clustering
plt.figure(figsize =(6, 6))
plt.scatter(X_principal['PC2'], X_principal['PC3'],
        c = ac8.fit_predict(X_principal), cmap ='rainbow')
plt.ylabel('Operating enviromental performance over time and distance')
plt.xlabel('Operating enviromental performance over region')
plt.show()


ac9 = AgglomerativeClustering(linkage="ward", affinity="euclidean", n_clusters = 3)

plt.figure(figsize =(6, 6))
plt.scatter(X_principal['PC2'], X_principal['PC3'],
        c = ac9.fit_predict(X_principal), cmap ='rainbow')
plt.ylabel('Operating enviromental performance over time and distance')
plt.xlabel('Operating enviromental performance over region')
plt.show()



ac10 = AgglomerativeClustering(linkage="ward", affinity="euclidean", n_clusters = 4)

plt.figure(figsize =(6, 6))
plt.scatter(X_principal['PC2'], X_principal['PC3'],
        c = ac10.fit_predict(X_principal), cmap ='rainbow')
plt.ylabel('Operating enviromental performance over time and distance')
plt.xlabel('Operating enviromental performance over region')
plt.show()



ac11 = AgglomerativeClustering(linkage="ward", affinity="euclidean", n_clusters = 5)

plt.figure(figsize =(6, 6))
plt.scatter(X_principal['PC2'], X_principal['PC3'],
        c = ac11.fit_predict(X_principal), cmap ='rainbow')
plt.ylabel('Operating enviromental performance over time and distance')
plt.xlabel('Operating enviromental performance over region')
plt.show()
```

```
ac12 = AgglomerativeClustering(linkage="ward", affinity="euclidean", n_clusters = 6)

plt.figure(figsize =(6, 6))
plt.scatter(X_principal['PC2'], X_principal['PC3'],
        c = ac12.fit_predict(X_principal), cmap ='rainbow')
plt.ylabel('Operating enviromental performance over time and distance')
plt.xlabel('Operating enviromental performance over region')
plt.show()




# Perform clustering for first 3 PCs

print('-----CLUSTERING FOR PC1, PC2 & PC3----------------')
# Clustering for PC2 and PC3

ac100 = AgglomerativeClustering(linkage="ward", affinity="euclidean", n_clusters = 2) #for 2
clusters

# Visualizing the clustering
plt.figure(figsize =(6, 6))
plt.scatter(X_principal['PC1'], X_principal['PC2'], X_principal['PC3'],
        c = ac100.fit_predict(X_principal), cmap ='rainbow')
plt.ylabel('Operating enviromental performance over time and distance')
plt.xlabel('Operating enviromental performance over region')
plt.show()


ac101 = AgglomerativeClustering(linkage="ward", affinity="euclidean", n_clusters = 3)

plt.figure(figsize =(6, 6))
plt.scatter(X_principal['PC1'], X_principal['PC2'], X_principal['PC3'],
        c = ac101.fit_predict(X_principal), cmap ='rainbow')
plt.ylabel('Operating enviromental performance over time and distance')
plt.xlabel('Operating enviromental performance over region')
plt.show()



ac102 = AgglomerativeClustering(linkage="ward", affinity="euclidean", n_clusters = 4)

plt.figure(figsize =(6, 6))
plt.scatter(X_principal['PC1'], X_principal['PC2'], X_principal['PC3'],
        c = ac102.fit_predict(X_principal), cmap ='rainbow')
plt.ylabel('Operating enviromental performance over time and distance')
```

```
plt.xlabel('Operating enviromental performance over region')
plt.show()



ac103 = AgglomerativeClustering(linkage="ward", affinity="euclidean", n_clusters = 5)

plt.figure(figsize =(6, 6))
plt.scatter(X_principal['PC1'], X_principal['PC2'], X_principal['PC3'],
        c = ac103.fit_predict(X_principal), cmap ='rainbow')
plt.ylabel('Operating enviromental performance over time and distance')
plt.xlabel('Operating enviromental performance over region')
plt.show()



ac104 = AgglomerativeClustering(linkage="ward", affinity="euclidean", n_clusters = 6)

plt.figure(figsize =(6, 6))
plt.scatter(X_principal['PC1'], X_principal['PC2'], X_principal['PC3'],
        c = ac12.fit_predict(X_principal), cmap ='rainbow')
plt.ylabel('Operating enviromental performance over time and distance')
plt.xlabel('Operating enviromental performance over region')
plt.show()

db_final = AgglomerativeClustering(linkage= 'ward',
affinity='euclidean',n_clusters=4).fit(X_principal)
labels = db_final.labels_
data['label'] = labels

# View how many ships there are in each cluster
ships_mean = pd.DataFrame(labels)
ships_mean.rename(columns={0:'Clusters'},inplace=True)
print(ships_mean.Clusters.value_counts())



# How many ships there are in each cluster and what cluster is it
print(data.info())


clust1 = data[data['label']==1]
clust0 = data[data['label']==0]
clust2 = data[data['label']==2]
clust3 = data[data['label']==3]
```

```
clust1= clust1[['ves_dwt','ves_loa','ves_beam','ves_capacity_grain','ves_draft','ves_depth','Total CO2 emissions [m tonnes]',
        'Distance','Annual Total time spent at sea [hours]','Total time spent at sea [hours]','Annual average CO2 emissions per distance [kg CO2 / n mile]',
        'Annual average Fuel consumption per distance [kg / n mile]','CO2 emissions which occurred within ports under a MS jurisdiction at berth [m tonnes]',
        'CO2 emissions from all voyages between ports under a MS jurisdiction [m tonnes]']]

clust0 = clust0[['ves_dwt','ves_loa','ves_beam','ves_capacity_grain','ves_draft','ves_depth','Total CO2 emissions [m tonnes]',
        'Distance','Annual Total time spent at sea [hours]','Total time spent at sea [hours]','Annual average CO2 emissions per distance [kg CO2 / n mile]',
        'Annual average Fuel consumption per distance [kg / n mile]','CO2 emissions which occurred within ports under a MS jurisdiction at berth [m tonnes]',
        'CO2 emissions from all voyages between ports under a MS jurisdiction [m tonnes]']]


clust2 = clust2[['ves_dwt','ves_loa','ves_beam','ves_capacity_grain','ves_draft','ves_depth','Total CO2 emissions [m tonnes]',
        'Distance','Annual Total time spent at sea [hours]','Total time spent at sea [hours]','Annual average CO2 emissions per distance [kg CO2 / n mile]',
        'Annual average Fuel consumption per distance [kg / n mile]','CO2 emissions which occurred within ports under a MS jurisdiction at berth [m tonnes]',
        'CO2 emissions from all voyages between ports under a MS jurisdiction [m tonnes]']]

clust3 = clust3[['ves_dwt','ves_loa','ves_beam','ves_capacity_grain','ves_draft','ves_depth','Total CO2 emissions [m tonnes]',
        'Distance','Annual Total time spent at sea [hours]','Total time spent at sea [hours]','Annual average CO2 emissions per distance [kg CO2 / n mile]',
        'Annual average Fuel consumption per distance [kg / n mile]','CO2 emissions which occurred within ports under a MS jurisdiction at berth [m tonnes]',
        'CO2 emissions from all voyages between ports under a MS jurisdiction [m tonnes]']]

desc_clust0 =clust0.describe()

desc_clust1 = clust1.describe()

desc_clust2 = clust2.describe()

desc_clust3 = clust3.describe()



print('---------CLUSTERING & PCA WITHOUT OUTLIERS----------')

# importing our data
```

```
data = pd.read_csv(r'C:\Users\Administrator\Desktop\Thesis Data\Ship Index\Ship Index\FINAL_THESIS_DATA_2021.csv')
print(data.info())



## Exploratory data analysis -- Visualization ##


# Looking for duplicates indices and print the total sum of duplicates
duplicates = (data.index[data.duplicated()])
print(data.duplicated().sum())

# Searching for missing values
print(data.isna().sum())

# Dropping rows missing values of column 'Distance'
data.dropna(subset=['Distance'],inplace=True)
print(data.info())


print('----------PCA & CLUSTERING----------')


df = data.drop(columns=['Ship type','Reporting Period',
                        'Annual average Fuel consumption per transport work (mass) [g / m tonnes Â· n miles]',
                        'Annual average CO2 emissions per transport work (mass) [g CO2 / m tonnes Â· n miles]'])

# Using regex to rename columns - - crossing out '[....]' just for visualizing purposes
df = df.rename(columns=lambda x: re.sub('(.*)(?:\s\[).*',r'\1',x))

print(df.info())


# DROP OUTLIERS
import pandas as pd
import numpy as np
from scipy import stats


# Removing outliers using Z score
df_new = df[(np.abs(stats.zscore(df)) < 3).all(axis=1)]
df = df_new
```

```
# Standardization of dataset and PCA

#Standardization
from sklearn.preprocessing import StandardScaler
# define standard scaler
X_std = StandardScaler()
# transform data
X_std = X_std.fit_transform(df)
print(X_std)


#Principal Component analysis for reduction of dataset -1st method for Min Max scaled DATA
from sklearn.decomposition import PCA

pca = PCA()#PCA model
Xstd_pca = pca.fit_transform(X_std)#transform and fit the data to pca model

#print(pca.components_.T)
pcs = pd.DataFrame(pca.components_.T)#traspose pca components and insert to dataframe - we rotate the components ->more variance explained
columns = []

#auto create columns based the number of PC's
for i in range(len(pca.components_)):
   columns.append('PC'+str(i+1))
pcs.columns =  columns
pcs.index = df.columns#use the data columns as index in the new dataframe pcs
print(pcs)



# Plot scree plot and see eigenvalues to conclude in how many pcs we should keep

explained_variance = pd.DataFrame()
index = []
#create an index for each PC in dataframe
for i in range(len(pca.explained_variance_)):
   index.append('PC'+str(i+1))
explained_variance['Eigenvalues'] = pca.explained_variance_#import explained variance to dataframe column  #explained_variance = diagonal elements of PC's Covariance matrix
explained_variance.index = index
print(explained_variance)
#plot the explained variance
plt.plot(pca.explained_variance_)
plt.xlabel('number of components')
plt.ylabel('Eigenvalues');
```

```
plt.xticks(np.arange(len(df.columns)), ('1', '2', '3', '4', '5', '6','7','8','9','10','11','12','13','14','15','16','17','18','19'))
plt.show()
#plot the cumulative explained variance
plt.plot(np.cumsum(pca.explained_variance_ratio_))
plt.xlabel('number of components')
plt.ylabel('cumulative explained variance ratio')
plt.xticks(np.arange(len(df.columns)), ('1', '2', '3', '4', '5', '6','7','8','9','10','11','12','13','14','15','16','17','18','19'))
plt.show()


# Calculating the eigenvalues again, with the total percentage of variance explained of each PC
cov_mat = np.cov(X_std.T)

# From this covariance matrix, caluclate the Eigenvalues and the Eigenvectors
eigen_vals, eigen_vecs = np.linalg.eig(cov_mat)

# print the Eigenvalues
print("Raw Eigenvalues: \n", eigen_vals)
# the sum of the Eigenvalues
print("Percentage of Variance Explained by Each Component: \n", eigen_vals/sum(eigen_vals))



#Loadings and biplot


# Scatter points in biplot
pca = PCA(n_components=2)

PC_scores = pd.DataFrame(pca.fit_transform(X_std),
        columns = ['PC1', 'PC2'])
print(PC_scores.head(10))

#Loadings in a dataframe for first 2 PCs

loadings = pd.DataFrame(pca.components_.T, columns=['PC1', 'PC2'],
                index=df.columns)
print(loadings)

# Data of biplot

PC1 = pca.fit_transform(X_std)[:,0]
PC2 = pca.fit_transform(X_std)[:,1]
ldngs = pca.components_
```

```
#We should also define the scaling factors scalePC1 and scalePC2 to fit the data of PC1, PC2 and ldngs on the same plot.
# Additionally, we can define the target group names to label the target groups in our biplots, see features below

#renaming columns names for better visualization plots
#df.columns =
['Z1','Z2','Z3','Z4','Z5','Z6','Z7','Z8','Z9','Z10','Z11','Z12','Z13','Z14','Z15','Z16','Z17','Z18','Z19']


scalePC1 = 1.0/(PC1.max() - PC1.min())
scalePC2 = 1.0/(PC2.max() - PC2.min())
features = df.columns


#Biplot

fig, ax = plt.subplots(figsize=(14, 9))

for i, feature in enumerate(features):
   ax.arrow(0, 0, ldngs[0, i],
         ldngs[1, i],
         head_width=0.03,
         head_length=0.03,
         color="red")
   ax.text(ldngs[0, i] * 1.15,
        ldngs[1, i] * 1.15,
        feature,color="purple", fontsize=18)

   ax.scatter(PC1 * scalePC1,
              PC2 * scalePC2, s=5)

for i, label in enumerate(PC_scores.index):
   ax.text(PC1[i] * scalePC1,
        PC2[i] * scalePC2, str(label),
        fontsize=10)

ax.set_xlabel('PC1', fontsize=20)
ax.set_ylabel('PC2', fontsize=20)
ax.set_title('Figure 2', fontsize=20)


# LOADINGS TABLE OF PCA for the first 3 PCs

pca = PCA(n_components=3)

PC_scores = pd.DataFrame(pca.fit_transform(X_std),
          columns = ['PC1', 'PC2','PC3'])
```

```
print(PC_scores.head(10))

#Loadings in a dataframe for first 2 PCs

loadings = pd.DataFrame(pca.components_.T, columns=['PC1', 'PC2','PC3'],
                        index=df.columns)
print(loadings)

# Data of biplot

PC1 = pca.fit_transform(X_std)[:,0]
PC2 = pca.fit_transform(X_std)[:,1]
PC3 = pca.fit_transform(X_std)[:,2]
ldngs = pca.components_



print('----------HIERARCHICAL CLUSTERING----------')

#importing our neccessary libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.decomposition import PCA #dimensionality reduction - better view of our plots
from sklearn.cluster import AgglomerativeClustering #for agglomerative clustering
from sklearn.preprocessing import StandardScaler #standard scaling
from sklearn.metrics import silhouette_score  #our scoring coefficient
import scipy.cluster.hierarchy as shc

#our full dataset

#Standardization
from sklearn.preprocessing import StandardScaler
# define standard scaler
X_std = StandardScaler()
# transform data
X_std = X_std.fit_transform(df)
print(X_std)

pca = PCA(n_components = 3) #We will keep the first 3 principal components
X_principal = pca.fit_transform(X_std)
X_principal = pd.DataFrame(X_principal)
X_principal.columns = ['PC1','PC2','PC3']


# Ward dendrogram
plt.figure(figsize =(8, 8))
plt.title('Visualising the full dataset')
```

```
Dendrogram = shc.dendrogram((shc.linkage(X_principal, method ='ward')))


#furthest neighbor dendrogram

plt.figure(figsize =(8, 8))
plt.title('Visualising the full dataset')
Dendrogram = shc.dendrogram((shc.linkage(X_principal, method ='complete')))


print('------------------CLUSTERING FOR PC1 & PC2-------------------------')
# CLUSTERING FOR PC1 AND PC2

ac2 = AgglomerativeClustering(linkage="ward", affinity="euclidean", n_clusters = 2) #for 2 clusters

# Visualizing the clustering
plt.figure(figsize =(6, 6))
plt.scatter(X_principal['PC1'], X_principal['PC2'],
        c = ac2.fit_predict(X_principal), cmap ='rainbow')
plt.ylabel('Operating enviromental performance over time and distance')
plt.xlabel('Dimensions of the ship')
plt.show()


ac3 = AgglomerativeClustering(linkage="ward", affinity="euclidean", n_clusters = 3)

plt.figure(figsize =(6, 6))
plt.scatter(X_principal['PC1'], X_principal['PC2'],
        c = ac3.fit_predict(X_principal), cmap ='rainbow')
plt.ylabel('Operating enviromental performance over time and distance')
plt.xlabel('Dimensions of the ship')
plt.show()



ac4 = AgglomerativeClustering(linkage="ward", affinity="euclidean", n_clusters = 4)

plt.figure(figsize =(6, 6))
plt.scatter(X_principal['PC1'], X_principal['PC2'],
        c = ac4.fit_predict(X_principal), cmap ='rainbow')
plt.ylabel('Operating enviromental performance over time and distance')
plt.xlabel('Dimensions of the ship')
plt.show()


ac55 = AgglomerativeClustering(linkage="ward", affinity="euclidean", n_clusters = 5)
```

```
plt.figure(figsize =(6, 6))
plt.scatter(X_principal['PC1'], X_principal['PC2'],
        c = ac55.fit_predict(X_principal), cmap ='rainbow')
plt.ylabel('Operating enviromental performance over time and distance')
plt.xlabel('Dimensions of the ship')
plt.show()



ac66 = AgglomerativeClustering(linkage="ward", affinity="euclidean", n_clusters = 6)

plt.figure(figsize =(6, 6))
plt.scatter(X_principal['PC1'], X_principal['PC2'],
        c = ac66.fit_predict(X_principal), cmap ='rainbow')
plt.ylabel('Operating enviromental performance over time and distance')
plt.xlabel('Dimensions of the ship')
plt.show()


# Davies bouldind and silhouette coefficient plots for the appropriate number of clusters for PC1 and PC2

#silhouette scores for number of clusters
k = [2, 3, 4, 5, 6]

# Appending the silhouette scores of the different models to the list
silhouette_scores = []
silhouette_scores.append(
      silhouette_score(X_principal, ac2.fit_predict(X_principal)))
silhouette_scores.append(
      silhouette_score(X_principal, ac3.fit_predict(X_principal)))
silhouette_scores.append(
      silhouette_score(X_principal, ac4.fit_predict(X_principal)))
silhouette_scores.append(
      silhouette_score(X_principal, ac55.fit_predict(X_principal)))
silhouette_scores.append(
      silhouette_score(X_principal, ac66.fit_predict(X_principal)))

# Plotting a bar graph to compare the results
plt.bar(k, silhouette_scores)
plt.xlabel('Number of clusters', fontsize = 20)
plt.ylabel('Silhouette Coef.', fontsize = 20)



from sklearn.metrics import davies_bouldin_score
#davies bouldin scores for number of clusters
k = [2, 3, 4, 5, 6]
```

```
# Appending the silhouette scores of the different models to the list
davies_bouldin_scores = []
davies_bouldin_scores.append(
    davies_bouldin_score(X_principal, ac2.fit_predict(X_principal)))
davies_bouldin_scores.append(
    davies_bouldin_score(X_principal, ac3.fit_predict(X_principal)))
davies_bouldin_scores.append(
    davies_bouldin_score(X_principal, ac4.fit_predict(X_principal)))
davies_bouldin_scores.append(
    davies_bouldin_score(X_principal, ac55.fit_predict(X_principal)))
davies_bouldin_scores.append(
    davies_bouldin_score(X_principal, ac66.fit_predict(X_principal)))

# Plotting a bar graph to compare the results
plt.bar(k, davies_bouldin_scores,color='orange')
plt.xlabel('Number of clusters', fontsize = 20)
plt.ylabel('Davies - Bouldin', fontsize = 20)
plt.show()

from sklearn import metrics
print('---------the exact values of evaluation indexes for each  cluster----------')

# exact calculation of silhouette coefficient for the number of clusters we chose
db2 = AgglomerativeClustering(linkage= 'ward', affinity='euclidean',n_clusters=2).fit(X_principal)
labels = db2.labels_
print("Silhouette Coefficient for 2 clusters: %0.3f" % metrics.silhouette_score(X_principal, labels))

db3 = AgglomerativeClustering(linkage= 'ward', affinity='euclidean',n_clusters=3).fit(X_principal)
labels = db3.labels_
print("Silhouette Coefficient for 3 clusters: %0.3f" % metrics.silhouette_score(X_principal, labels))

db4 = AgglomerativeClustering(linkage= 'ward', affinity='euclidean',n_clusters=4).fit(X_principal)
labels = db4.labels_
print("Silhouette Coefficient for 4 clusters: %0.3f" % metrics.silhouette_score(X_principal, labels))

db55 = AgglomerativeClustering(linkage= 'ward',
affinity='euclidean',n_clusters=5).fit(X_principal)
labels = db55.labels_
print("Silhouette Coefficient for 5 clusters: %0.3f" % metrics.silhouette_score(X_principal, labels))

db66 = AgglomerativeClustering(linkage= 'ward',
affinity='euclidean',n_clusters=6).fit(X_principal)
labels = db66.labels_
print("Silhouette Coefficient for 6 clusters: %0.3f" % metrics.silhouette_score(X_principal, labels))
```

```
# Davies bouldin

db22 = AgglomerativeClustering(linkage= 'ward', affinity='euclidean',n_clusters=2).fit(X_principal)
labels = db22.labels_
print("Davies Bouldin for 2 clusters: %0.3f" % metrics.davies_bouldin_score(X_principal, labels))

db32 = AgglomerativeClustering(linkage= 'ward', affinity='euclidean',n_clusters=3).fit(X_principal)
labels = db32.labels_
print("Davies Bouldin for 3 clusters: %0.3f" % metrics.davies_bouldin_score(X_principal, labels))

db42 = AgglomerativeClustering(linkage= 'ward', affinity='euclidean',n_clusters=4).fit(X_principal)
labels = db42.labels_
print("Davies Bouldin for 4 clusters: %0.3f" % metrics.davies_bouldin_score(X_principal, labels))

db52 = AgglomerativeClustering(linkage= 'ward', affinity='euclidean',n_clusters=5).fit(X_principal)
labels = db52.labels_
print("Davies Bouldin for 5 clusters: %0.3f" % metrics.davies_bouldin_score(X_principal, labels))

db62 = AgglomerativeClustering(linkage= 'ward', affinity='euclidean',n_clusters=6).fit(X_principal)
labels = db62.labels_
print("Davies Bouldin for 6 clusters: %0.3f" % metrics.davies_bouldin_score(X_principal, labels))



# Calinski Harabasz Score

db23 = AgglomerativeClustering(linkage= 'ward', affinity='euclidean',n_clusters=2).fit(X_principal)
labels = db23.labels_
print("Calinski Harabasz for 2 clusters: %0.3f" % metrics.calinski_harabasz_score(X_principal, labels))

db33 = AgglomerativeClustering(linkage= 'ward', affinity='euclidean',n_clusters=3).fit(X_principal)
labels = db33.labels_
print("Calinski Harabasz for 3 clusters: %0.3f" % metrics.calinski_harabasz_score(X_principal, labels))

db43 = AgglomerativeClustering(linkage= 'ward', affinity='euclidean',n_clusters=4).fit(X_principal)
labels = db43.labels_
```

```
print("Calinski Harabasz for 4 clusters: %0.3f" % metrics.calinski_harabasz_score(X_principal, labels))

db53 = AgglomerativeClustering(linkage= 'ward', affinity='euclidean',n_clusters=5).fit(X_principal)
labels = db53.labels_
print("Calinski Harabasz for 5 clusters: %0.3f" % metrics.calinski_harabasz_score(X_principal, labels))

db63 = AgglomerativeClustering(linkage= 'ward', affinity='euclidean',n_clusters=6).fit(X_principal)
labels = db63.labels_
print("Calinski Harabasz for 6 clusters: %0.3f" % metrics.calinski_harabasz_score(X_principal, labels))



print('----------------------SECOND PAIR OF PCs   PC1 & PC3--------------------')
# Clustering for PC1 and PC3

ac5 = AgglomerativeClustering(linkage="ward", affinity="euclidean", n_clusters = 2) #for 2 clusters

# Visualizing the clustering
plt.figure(figsize =(6, 6))
plt.scatter(X_principal['PC1'], X_principal['PC3'],
        c = ac5.fit_predict(X_principal), cmap ='rainbow')
plt.ylabel('Operating enviromental performance over region')
plt.xlabel('Dimensions of the ship')
plt.show()


ac6 = AgglomerativeClustering(linkage="ward", affinity="euclidean", n_clusters = 3)

plt.figure(figsize =(6, 6))
plt.scatter(X_principal['PC1'], X_principal['PC3'],
        c = ac6.fit_predict(X_principal), cmap ='rainbow')
plt.ylabel('Operating enviromental performance over region')
plt.xlabel('Dimensions of the ship')
plt.show()



ac7 = AgglomerativeClustering(linkage="ward", affinity="euclidean", n_clusters = 4)

plt.figure(figsize =(6, 6))
plt.scatter(X_principal['PC1'], X_principal['PC3'],
        c = ac7.fit_predict(X_principal), cmap ='rainbow')
```

```
plt.ylabel('Operating enviromental performance over region')
plt.xlabel('Dimensions of the ship')
plt.show()


ac77 = AgglomerativeClustering(linkage="ward", affinity="euclidean", n_clusters = 5)

plt.figure(figsize =(6, 6))
plt.scatter(X_principal['PC1'], X_principal['PC3'],
        c = ac77.fit_predict(X_principal), cmap ='rainbow')
plt.ylabel('Operating enviromental performance over region')
plt.xlabel('Dimensions of the ship')
plt.show()



ac78 = AgglomerativeClustering(linkage="ward", affinity="euclidean", n_clusters = 6)

plt.figure(figsize =(6, 6))
plt.scatter(X_principal['PC1'], X_principal['PC3'],
        c = ac78.fit_predict(X_principal), cmap ='rainbow')
plt.ylabel('Operating enviromental performance over region')
plt.xlabel('Dimensions of the ship')
plt.show()


print('-----CLUSTERING FOR PC2 & PC3----------')
# Clustering for PC2 and PC3

ac8 = AgglomerativeClustering(linkage="ward", affinity="euclidean", n_clusters = 2) #for 2
clusters

# Visualizing the clustering
plt.figure(figsize =(6, 6))
plt.scatter(X_principal['PC2'], X_principal['PC3'],
        c = ac8.fit_predict(X_principal), cmap ='rainbow')
plt.ylabel('Operating enviromental performance over time and distance')
plt.xlabel('Operating enviromental performance over region')
plt.show()


ac9 = AgglomerativeClustering(linkage="ward", affinity="euclidean", n_clusters = 3)

plt.figure(figsize =(6, 6))
plt.scatter(X_principal['PC2'], X_principal['PC3'],
        c = ac9.fit_predict(X_principal), cmap ='rainbow')
plt.ylabel('Operating enviromental performance over time and distance')
plt.xlabel('Operating enviromental performance over region')
```

```
plt.show()



ac10 = AgglomerativeClustering(linkage="ward", affinity="euclidean", n_clusters = 4)

plt.figure(figsize =(6, 6))
plt.scatter(X_principal['PC2'], X_principal['PC3'],
        c = ac10.fit_predict(X_principal), cmap ='rainbow')
plt.ylabel('Operating enviromental performance over time and distance')
plt.xlabel('Operating enviromental performance over region')
plt.show()



ac11 = AgglomerativeClustering(linkage="ward", affinity="euclidean", n_clusters = 5)

plt.figure(figsize =(6, 6))
plt.scatter(X_principal['PC2'], X_principal['PC3'],
        c = ac11.fit_predict(X_principal), cmap ='rainbow')
plt.ylabel('Operating enviromental performance over time and distance')
plt.xlabel('Operating enviromental performance over region')
plt.show()



ac12 = AgglomerativeClustering(linkage="ward", affinity="euclidean", n_clusters = 6)

plt.figure(figsize =(6, 6))
plt.scatter(X_principal['PC2'], X_principal['PC3'],
        c = ac12.fit_predict(X_principal), cmap ='rainbow')
plt.ylabel('Operating enviromental performance over time and distance')
plt.xlabel('Operating enviromental performance over region')
plt.show()




db_final=AgglomerativeClustering(linkage='ward',
affinity='euclidean',n_clusters=4).fit(X_principal)
labels = db_final.labels_
df_new['label'] = labels



# View how many ships there are in each cluster
ships_mean = pd.DataFrame(labels)
```

```
ships_mean.rename(columns={0:'Clusters'},inplace=True)
print(ships_mean.Clusters.value_counts())



# How many ships there are in each cluster and what cluster is it
print(df_new.info())


clust1 = df_new[df_new['label']==1]
clust0 = df_new[df_new['label']==0]
clust2 = df_new[df_new['label']==2]
clust3 = df_new[df_new['label']==3]

clust1= clust1[['ves_dwt','ves_loa','ves_beam','ves_capacity_grain','ves_draft','ves_depth','Total CO2 emissions','Total fuel consumption',
        'Distance','Annual Total time spent at sea','Total time spent at sea','Annual average CO2 emissions per distance',
        'Annual average Fuel consumption per distance','CO2 emissions which occurred within ports under a MS jurisdiction at berth',
        'CO2 emissions from all voyages between ports under a MS jurisdiction']]

clust0 = clust0[['ves_dwt','ves_loa','ves_beam','ves_capacity_grain','ves_draft','ves_depth','Total CO2 emissions','Total fuel consumption',
        'Distance','Annual Total time spent at sea','Total time spent at sea','Annual average CO2 emissions per distance',
        'Annual average Fuel consumption per distance','CO2 emissions which occurred within ports under a MS jurisdiction at berth',
        'CO2 emissions from all voyages between ports under a MS jurisdiction']]


clust2 = clust2[['ves_dwt','ves_loa','ves_beam','ves_capacity_grain','ves_draft','ves_depth','Total CO2 emissions','Total fuel consumption',
        'Distance','Annual Total time spent at sea','Total time spent at sea','Annual average CO2 emissions per distance',
        'Annual average Fuel consumption per distance','CO2 emissions which occurred within ports under a MS jurisdiction at berth',
        'CO2 emissions from all voyages between ports under a MS jurisdiction']]

clust3 = clust3[['ves_dwt','ves_loa','ves_beam','ves_capacity_grain','ves_draft','ves_depth','Total CO2 emissions','Total fuel consumption',
        'Distance','Annual Total time spent at sea','Total time spent at sea','Annual average CO2 emissions per distance',
        'Annual average Fuel consumption per distance','CO2 emissions which occurred within ports under a MS jurisdiction at berth',
        'CO2 emissions from all voyages between ports under a MS jurisdiction']]
```

```
# get the descriptive statistics for each cluster
desc_clust0 =clust0.describe()

desc_clust1 = clust1.describe()

desc_clust2 = clust2.describe()

desc_clust3 = clust3.describe()
```