



«ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ ΚΑΙ ΤΕΧΝΟΛΟΓΙΑΣ»

«Μεταπτυχιακή Εξειδίκευση στα Πληροφοριακά
Συστήματα MSc (ΠΛΣ)»

Διπλωματική Εργασία

«Σχεδιασμός και ανάπτυξη εφαρμογής πρόβλεψης καθυστερήσεων
σε διαδρομές πλοίων (με τεχνικές μηχανικής μάθησης)»

ΠΑΡΟΙΚΑΚΗΣ ΓΕΩΡΓΙΟΣ

Επιτροπή Επίβλεψης Διπλωματικής Εργασίας

Επιβλέπων καθηγητής:

Συν-Επιβλέπων Καθηγητής

Παξιμάδης Κωνσταντίνος

Μπεληγιάννης Γρηγόριος

Ελληνικό Ανοικτό Πανεπιστήμιο

Ελληνικό Ανοικτό Πανεπιστήμιο

Μέλος ΣΕΠ

Μέλος ΣΕΠ

Πάτρα, «Μάιος» «2025»

Η παρούσα εργασία αποτελεί πνευματική ιδιοκτησία του/της φοιτητή/φοιτήτριας («συγγραφέας/δημιουργός») που την εκπόνησε. Στο πλαίσιο της πολιτικής ανοικτής πρόσβασης ο συγγραφέας/δημιουργός εκχωρεί στο ΕΑΠ, μη αποκλειστική άδεια χρήσης του δικαιώματος αναπαραγωγής, προσαρμογής, δημόσιου δανεισμού, παρουσίασης στο κοινό και ψηφιακής διάχυσής τους διεθνώς, σε ηλεκτρονική μορφή και σε οποιοδήποτε μέσο, για διδακτικούς και ερευνητικούς σκοπούς, άνευ ανταλλάγματος και για όλο το χρόνο διάρκειας των δικαιωμάτων πνευματικής ιδιοκτησίας. Η ανοικτή πρόσβαση στο πλήρες κείμενο για μελέτη και ανάγνωση δεν σημαίνει καθ' οιονδήποτε τρόπο παραχώρηση δικαιωμάτων διανοητικής ιδιοκτησίας του συγγραφέα/δημιουργού ούτε επιτρέπει την αναπαραγωγή, αναδημοσίευση, αντιγραφή, αποθήκευση, πώληση, εμπορική χρήση, μετάδοση, διανομή, έκδοση, εκτέλεση, «μεταφόρτωση» (downloading), «ανάρτηση» (uploading), μετάφραση, τροποποίηση με οποιονδήποτε τρόπο, τμηματικά ή περιληπτικά της εργασίας, χωρίς τη ρητή προηγούμενη έγγραφη συναίνεση του συγγραφέα/δημιουργού. Ο συγγραφέας/δημιουργός διατηρεί το σύνολο των ηθικών και περιουσιακών του δικαιωμάτων.



«Σχεδιασμός και ανάπτυξη εφαρμογής πρόβλεψης καθυστερήσεων
σε διαδρομές πλοίων (με τεχνικές μηχανικής μάθησης)»

«ΠΑΡΟΙΚΑΚΗΣ ΓΕΩΡΓΙΟΣ»

Επιτροπή Επίβλεψης Διπλωματικής Εργασίας

Επιβλέπων καθηγητής:

Συν-Επιβλέπων Καθηγητής

Παξιμάδης Κωνσταντίνος

Μπεληγιάννης Γρηγόριος

Ελληνικό Ανοικτό Πανεπιστήμιο

Ελληνικό Ανοικτό Πανεπιστήμιο

Μέλος ΣΕΠ

Μέλος ΣΕΠ

Πάτρα, «Μάιος» «2025»

«Στην Αγγελική για την υπομονή και την κατανόησή της...»

Περίληψη

Η ναυσιπλοΐα αποτελεί τον παλαιότερο τρόπο μεταφοράς προϊόντων και επιβατών. Τα πρώτα μέσα ναυσιπλοΐας ήταν βάρκες φτιαγμένες από κορμούς δέντρων. Σημαντικές ανακαλύψεις της παγκόσμιας ιστορίας θα ήταν ανέφικτες χωρίς τη δυνατότητα μετακίνησης μέσω της θάλασσας. Ο χώρος της ναυτιλίας επηρεάζει σημαντικά και σήμερα εν γένει την παγκόσμια οικονομία ενώ η τεχνολογική πρόοδος στην κατασκευή των πλοίων συμβαδίζει με την ανάπτυξη του ανθρώπινου πολιτισμού. Τα πρώτα ίχνη ναυτιλιακής δραστηριότητας χρονολογούνται ήδη περίπου 10000 χρόνια πριν. Το 3000 π.Χ. οι Αιγύπτιοι ήταν οι πρώτοι που χρησιμοποίησαν βάρκες από ξύλινες σανίδες για μεταφορές μεγάλης κλίμακας στον Νείλο, τον Ευφράτη και τον Τίγρη.

Η ελληνική ακτοπλοΐα αποτελεί έναν από τους σημαντικότερους παράγοντες της ελληνικής οικονομίας για δύο κατά κυρίως λόγους: Καταρχάς η δομή του ελλαδικού χώρου με τα δεκάδες νησιά, καθιστούν την ακτοπλοΐα σε αρκετές περιπτώσεις αν όχι τον μοναδικό αλλά σίγουρα τον δημοφιλέστερο τρόπο προσέγγισης αυτών των νησιών από την υπόλοιπη ηπειρωτική χώρα. Επιπλέον η «βαριά» βιομηχανία της Ελλάδας, δηλαδή ο τουρισμός, εξυπηρετείται κατά το μεγαλύτερο ποσοστό κατά τους θερινούς μήνες μέσω θαλάσσης, τόσο λόγω του αριθμού των επιβατών αλλά και κόστους. Σκοπός της παρούσας εργασίας είναι η ανάπτυξη ενός πληροφοριακού συστήματος ικανού να εντοπίσει με σημαντική πιθανότητα επιτυχίας, το ενδεχόμενο καθυστέρησης προσέλευσης ή απόπλου των επιβατικών πλοίων.

Η διπλωματική αποτελείται από θεωρητικό και τεχνικό σκέλος. Στο πρώτο σκέλος πραγματοποιείται η βιβλιογραφική επισκόπηση του χώρου της ακτοπλοΐας και των σημαντικότερων λόγων καθυστέρησης καθώς και του επιστημονικού χώρου της μηχανικής μάθησης πάνω στην οποία θα βασιστεί η ανάπτυξη του συστήματος. Το τεχνικό μέρος ασχολείται με το σχεδιασμό και την ανάπτυξη του συστήματος πρόβλεψης ακτοπλοϊκών καθυστερήσεων.

Λέξεις – Κλειδιά

Ακτοπλοΐα, καθυστερήσεις πλοίων, μηχανική μάθηση

*«Design and development of application for forecasting
delays in ship routes (with machine learning techniques)»*

«George Parikakis»

Abstract

Shipping is the oldest way of transporting goods and passengers. The first means of navigation were boats made from tree trunks. Major discoveries in world history would have been impossible without the ability to travel by sea. The shipping sector has a significant impact on the world economy in general today. Technological progress in shipbuilding has kept pace with the development of human civilization. The first traces of maritime activity date back to around 10,000 years ago. In 3000 BC the Egyptians were the first to use wooden plank boats for large-scale transport on the Nile, Euphrates and Tigris.

Greek shipping is one of the most important factors of the Greek economy for two main reasons: First of all, the structure of the Greek area with dozens of islands, make shipping in several cases, if not the only, but certainly the most popular way of approaching these islands from the rest of the mainland. In addition, Greece's "heavy" industry, i.e. tourism, is mainly served by sea during the summer months, both due to the number of passengers and costs. The purpose of this work is the development of an information system capable of identifying with a significant probability of success, the possibility of delayed arrival or departure of passenger ships.

Diplomacy consists of a theoretical and a technical part. In the first part, the bibliographic overview of the field of coastal shipping and the most important reasons for delay is carried out, as well as the field of machine learning. The technical part deals with the design and development of a system for predicting ferry delays.

Keywords

Coastal shipping, ship delays, machine learning

Περιεχόμενα

Περίληψη.....	v
Περιεχόμενα	vii
Κατάλογος Εικόνων / Σχημάτων	viii
Κατάλογος Πινάκων	viii
Συντομογραφίες & Ακρωνύμια.....	x
1. ΠΡΟΛΟΓΟΣ	13
2. Θαλάσσιες μεταφορές – Δρομολόγια	15
2.1. Ιστορία.....	16
2.2. Η ελληνική πραγματικότητα	21
2.3. Καθυστερήσεις δρομολογίων. Ανίχνευση αιτιών μέσω βιβλιογραφικής ανασκόπησης.....	27
3. Μηχανική μάθηση.....	29
3.1. Δέντρα απόφασης.....	30
3.2. Νευρωνικά δίκτυα	35
3.3. Κανόνες συσχέτισης.....	42
4. Σχεδιασμός Συστήματος	45
4.1. Ανάκτηση Δεδομένων.....	46
4.2. Ανάλυση Δεδομένων.....	49
4.3. Ανάπτυξη νευρωνικού δικτύου	59
4.4. Παρουσίαση λειτουργίας συστήματος	64
5. Συμπεράσματα - Αξιολόγηση	67
Βιβλιογραφία,.....	71
Παράρτημα Α: «Εκτέλεση J48».....	75
Παράρτημα Β: «Κώδικας Python»	97
Παράρτημα Γ: «Συντομογραφίες Λιμανιών»	109
Παράρτημα Δ: «Αυτοματοποιημένο σενάριο (automation script) σε γλώσσα προγραμματισμού Python».....	118

Κατάλογος Εικόνων / Σχημάτων

Εικόνα 1 Η αθηναϊκή Τριήρης.....	18
Εικόνα 2. Κατηγοριοποίηση εταιρειών ελληνικής ακτοπλοΐας (Πηγή: ΧRTC).....	25
Εικόνα 3. Παράδειγμα δέντρου απόφασης (Πηγή: Song et al, 2015)	34
Εικόνα 4. Η αρχιτεκτονική ενός ANN(Bre et al 2018).....	38
Εικόνα 5. Βάρος κάθε στοιχείου και είσοδος - έξοδος ενός τυπικού συστήματος ANN ...	39
Εικόνα 6. Η αρχιτεκτονική των συστημάτων τοπολογιών νευρωνικών δικτύων FeedForward (Mitra and Paul 2017).	40
Εικόνα 7. Η αρχιτεκτονική των συστημάτων FeedBack (Mitra and Paul 2017).	41
Εικόνα 8. Η αναγνώριση των γραμμάτων T και H με τη χρήση της προσαρμοστικότητας στα συστήματα ANN (Qasim 2013).....	41
Εικόνα 9. Οι μαθησιακές ικανότητες των συστημάτων ANN (Qasim 2013).	42
Εικόνα 10. Η εφαρμογή των ANN σε συστήματα επεξεργασίας σήματος (Cichocki, Unbehauen και Swiniarski 1993)	43
Εικόνα 11. Υποστήριξη και εμπιστοσύνη(Kumbhare et al., 2014)	45
Εικόνα 12. Στατιστικά J48	55
Εικόνα 13. Min-Max scaler	63
Εικόνα 14. Μετρικές νευρωνικού δικτύου	65
Εικόνα 15. Πίνακας συγχυσης.....	66
Εικόνα 16. Classification report	66
Εικόνα 17 Δείγμα στιγμιότυπου οθόνης από αυτοματοποιημένο σύστημα σε Linux.....	120

Κατάλογος Πινάκων

Πίνακας 1 Παγκόσμια μεταφορά επιβατών και οχημάτων στην ακτοπλοΐα (Πηγή ΟΗΕ) .	19
Πίνακας 2 Συνολικό μέγεθος παγκόσμιου στόλου από το 2008 έως το 2013(πηγή ΟΗΕ).	21
Πίνακας 3. Ο στόλος και η επιβατική κίνηση μεγάλων εταιρειών(Πηγή ΧRTC).....	26
Πίνακας 4. Μικρές εταιρείες ελληνικής ακτοπλοΐας(Πηγή ΣΕΕΝ)	27
Πίνακας 5. Κίνηση επιβατών ακτοπλοΐας (πηγή ΕΛΣΤΑΤ)	28
Πίνακας 6. Κίνηση οχημάτων στην ακτοπλοΐα (πηγή ΕΛΣΤΑΤ)	28
Πίνακας 7.Ναυτιλιακή ανά πλοίο	48

Συντομογραφίες & Ακρωνύμια

GRT	Gross Registered Tonnage
ANN	Artificial Neural Network

1. ΠΡΟΛΟΓΟΣ

Η παρούσα εργασία εκπονήθηκε κατά το ακαδημαϊκό έτος 2024-25 στα πλαίσια του μεταπτυχιακού προγράμματος «Μεταπτυχιακή Εξειδίκευση στα Πληροφοριακά Συστήματα». Η εργασία πραγματοποιήθηκε υπό την επίβλεψη του καθηγητή κου. Παξιμάδη Κωνσταντίνου.

Σκοπός της παρούσας εργασίας αποτελεί η συστηματική βιβλιογραφική επισκόπηση του κλάδου της ναυτιλίας τόσο σε τοπικό όσο και σε διεθνές επίπεδο. Ειδικότερα η εργασία εστιάζει στην ανίχνευση των αιτιών που προκαλούν καθυστερήσεις τόσο κατά την άφιξη όσο και κατά τον απόπλου επιβατηγών πλοίων. Επίσης καταγράφονται οι πλέον διαδεδομένοι μηχανισμοί υλοποίησης μηχανική μάθησης όπως τα νευρωνικά δίκτυα, τα δέντρα απόφασης κ.α. Τέλος με βάση τα ευρήματα της βιβλιογραφικής επισκόπησης τόσο του τομέα της ακτοπλοΐας αλλά και της μηχανικής μάθησης επιχειρείται η ανάπτυξη ενός πληροφοριακού συστήματος ικανού να προβλέψει με ικανοποιητικό βαθμό ακρίβειας δρομολόγια με ενδεχόμενη καθυστέρηση καθώς και ένα κατά προσέγγιση υπολογισμό του προβλεπόμενου χρόνου καθυστέρησης.

Στα προσδοκώμενα αποτελέσματα καταγράφεται κυρίως η συγκέντρωση της ήδη υπάρχουσας γνώσης στο τομέα της πρόβλεψης της καθυστέρησης στο χώρο των μεταφορών και ειδικότερα στο χώρο της ακτοπλοΐας, όπου δεν υπάρχει προς το παρόν κάποια συστηματική επιστημονική καταγραφή. Η πρωτοτυπία της εργασίας συνίσταται κυρίως στην ανάπτυξη του συστήματος ανίχνευσης και πρόβλεψης των ακτοπλοϊκών καθυστερήσεων, αφού κάτι αντίστοιχο δεν καταγράφεται στην διεθνή βιβλιογραφία.

Η εργασία χωρίζεται σε 4 διακριτά κεφάλαια πέρα του παρόντος. Στο κεφάλαιο 2 επιχειρείται μία ιστορική αναδρομή του κλάδου της ναυτιλίας, η παρούσα κατάσταση στον ελλαδικό χώρο ενώ τέλος ανιχνεύονται οι κυριότεροι παράγοντες πρόκλησης χρονικών καθυστερήσεων. Στο κεφάλαιο 3 παρουσιάζονται οι σημαντικότεροι τρόποι εφαρμογής του κλάδου της μηχανικής μάθησης (νευρωνικά δίκτυα, δέντρα απόφασης κ.α.) Στο κεφάλαιο 4 καταγράφονται τα βήματα σχεδιασμού, ανάπτυξης και αξιολόγησης του συστήματος πρόβλεψης καθυστερήσεων. Τέλος στο κεφάλαιο 5

καταγράφονται συνολικά συμπεράσματα από το σύνολο των προηγούμενων κεφαλαίων.

2. Θαλάσσιες μεταφορές – Δρομολόγια

Στο παρόν κεφάλαιο επιχειρείται μία κατά κύριο λόγο βιβλιογραφική επισκόπηση του τομέα της ακτοπλοΐας τόσο σε ελληνικό όσο και σε διεθνές επίπεδο. Ειδικότερα στο πρώτο κεφάλαιο εξετάζεται ιστορικά από την αρχαιότητα ως τους νεότερους χρόνους, η εξέλιξη των μεταφορών μέσω της θάλασσας. Στο δεύτερο κεφάλαιο καταγράφεται η σημερινή κατάσταση της ακτοπλοΐας στον ελλαδικό χώρο, με αναφορές στον υπάρχοντα στόλο και καταγραφή των εταιρειών που δραστηριοποιούνται στο σύνολο της επικράτειας. Τέλος στο κεφάλαιο τρία παρουσιάζονται οι πλέον σημαντικοί λόγοι που προκαλούν καθυστέρηση τόσο κατά την άφιξη, όσο και κατά τον απόπλου των επιβατηγών πλοίων.

2.1. Ιστορία

Οι θαλάσσιες μεταφορές χρονολογούνται χιλιάδες χρόνια πίσω. Στην ναυτική ιστορία, στοιχεία του θαλάσσιου εμπορίου μεταξύ πολιτισμών καταγράφονται τουλάχιστον πριν δύο χιλιετίες. Τα πρώτα προϊστορικά σκάφη εικάζεται ότι ήταν κανό που αναπτύχθηκαν από διάφορους πληθυσμούς της Εποχής του Λίθου. Μεσολιθικό ναυπηγείο έχει βρεθεί στο Isle of Wight της Βρετανίας. Οι αρχαιολόγοι που καταδύθηκαν στην περιοχή, λίγο έξω από τις νότιες ακτές της Αγγλίας, βρήκαν τα ερείπια μιας ξύλινης πλατφόρμας που χρονολογείται πριν από 8.000 χρόνια, αποτελώντας την αρχαιότερη καταγραφή τοποθεσίας κατασκευής πλοίων, αν και υπάρχουν αρκετές ενστάσεις (Collins, 2019).

Στην Ασία, οι αρχαιότερες αποδείξεις των θαλασσίων μεταφορών αποτελούν τα νεολιθικά εμπορικά δίκτυα των αυστρωνησιακών λαών και κυρίως η εμπορία νεφρίτη των Φιλιππίνων, της Ταϊβάν, του νότιου Βιετνάμ και της χερσονήσου Ταϊλάνδης. Περιλάμβανε επίσης τις υπεραστικές διαδρομές των Αυστρωνησίων εμπορών από την Ινδονησία και τη Μαλαισία που συνέδεαν την Κίνα με τη Νότια Ασία και τη Μέση Ανατολή από το 500 π.Χ. περίπου. Η ανάπτυξη των θαλασσίων μεταφορών ευνόησε τη διάδοση των μπαχαρικών της Νοτιοανατολικής Ασίας και των κινεζικών προϊόντων προς τα δυτικά, καθώς και τη διάδοση του Ινδουισμού και του Βουδισμού στα ανατολικά. Η συγκεκριμένη διαδρομή θα γίνει αργότερα γνωστή ως ο Θαλάσσιος Δρόμος του Μεταξιού, αν και εσφαλμένα, καθώς κατά μήκος της διαδρομής διακινούνταν μπαχαρικά και όχι μετάξι. Πολλές αυστρωνησιακές τεχνολογίες, όπως το καταμαράν, καθώς και αυστρωνησιακές ορολογίες πλοίων, εξακολουθούν να υπάρχουν ακόμη και σήμερα (Doran, 1974).

Την ίδια περίοδο, οι Ρωμαίοι ανέπτυξαν στόλους που μπορούσαν να διασχίσουν τη Μεσόγειο Θάλασσα. Παρόμοια με τις επίγειες μεταφορές, οι δυνατότητες και η οργάνωση των θαλασσίων μεταφορών άλλαξαν ουσιαστικά κατά τη διάρκεια της Ρωμαϊκής αυτοκρατορίας. Μαζί με το μέγεθος των σκαφών και των πλοίων, αυξήθηκε δραματικά και η ποσότητα των εμπορευμάτων που αποστέλλονταν μέσω πλωτών οδών. Μεγάλες φορτηγίδες με επίπεδο πυθμένα χρησιμοποιήθηκαν για ποτάμιες μεταφορές και ήταν σε θέση να μεταφέρουν περίπου 150 τόνους φορτίου. Τα θαλάσσια

πλοία είχαν την δυνατότητα μεταφοράς έως και 1.000 τόνους. Τα κανάλια που συνήθως κατασκευάζονταν για να παρακάμπτουν επικίνδυνα σημεία ποταμών ή να διευκολύνουν τη ναυσιπλοΐα μέσω των δέλτα συνέβαλαν επίσης στη μείωση του κόστους μεταφοράς νερού. Η πολιτική σταθερότητα σε όλη την αυτοκρατορία (pax Romana) τόνωσε περαιτέρω τη δημιουργία και την εμβάθυνση εμπορικών σχέσεων μεγάλων αποστάσεων. Η πειρατεία στη Μεσόγειο, για παράδειγμα, παλαιότερα σύνηθες και αποτρεπτικό πρόβλημα του εμπορίου, καταργήθηκε σε μεγάλο βαθμό μετά το 67 π.Χ. Η εισαγωγή ενός κοινού νομίσματος καθώς και οι βελτιώσεις στις τεχνολογίες ναυτιλίας και εμπορευματοκιβωτίων (αμφορείς και βαρέλια) διευκόλυνε περαιτέρω το εμπόριο μεγάλων αποστάσεων (Fluckiger et al., 2021).

Η ιδιαίτερη γεωμορφία του ελλαδικού χώρου, με τα δεκάδες χιλιάδες χιλιόμετρα ακτογραμμών έστρεψε τα πρώιμα ελληνικά φύλλα προς την θάλασσα και κυρίως προς το Αιγαίο. Διάχυτη είναι η πεποίθηση ότι ήδη από την Μεσολιθική εποχή οι κάτοικοι τόσο της ηπειρωτικής όσο και της νησιωτικής Ελλάδας, είχαν την ικανότητα πλεύσης και μεταφοράς εμπορευμάτων. Στην εποχή του χαλκού τα θαλάσσια σύνορα των Ελλήνων διευρύνονται φτάνοντας μέχρι την Σικελία και τις Δαλματικές ακτές. Κατά το 2000 π.Χ. κυριαρχεί ο Μινωικός πολιτισμός με κέντρο της θαλάσσιας δραστηριότητας τη νήσο της Κρήτης. Τον μινωικό πολιτισμό διαδέχεται ο μυκηναϊκός, με κύρια χαρακτηριστικά την βελτίωσή της τεχνολογίας των πλοίων και ως άμεσο επακόλουθο την διεύρυνση των θαλάσσιων δρόμων, με τους Μυκήνες να κυριαρχούν σε όλη τη Μεσόγειο θάλασσα. Τον 11ο αιώνα η θαλάσσια κυριαρχία περνάει στην Αθήνα. Κατά την περίοδο αυτή, τα πλοία διαχωρίζονται σε εμπορικά και πολεμικά, με προφανείς κατασκευαστικές διαφοροποιήσεις. Σήμα κατατεθέν την εποχής αποτελεί η Τριήρης, πολεμικό πλοίο το οποίο χρησιμοποιήθηκε εναντίον των Φοινίκων και των Περσών, καθιστώντας τους Έλληνες κυρίαρχους της Μεσογείου (εικόνα 1).



Εικόνα 1 Η αθηναϊκή Τριήρης

Από τον 15ο αιώνα, η ανάπτυξη εμπορικών οδών κατά μήκος και μεταξύ του Ατλαντικού και του Ειρηνικού Ωκεανού άλλαξε την εικόνα του σύγχρονου κόσμου. Η εφεύρεση του ατμόπλοιου στις αρχές του 19ου αιώνα προκάλεσε μια ταχεία αύξηση στο μέγεθος του παγκόσμιου εμπορίου και πρακτικά ξεκίνησε τη διαδικασία της «σύγχρονης παγκοσμιοποίησης». Οι απαιτήσεις του ναυτιλιακού εμπορίου δημιούργησαν νέες επιχειρηματικές ανάγκες από την ασφάλιση έως τη ανάγκη εγχώριας αλλά και διεθνούς χρηματοδότησης, οδήγησε σε σημαντικές μεταβολές και βελτιώσεις στη μηχανολογία ενώ επέφερε και την δημιουργία νέων επιστημών προκειμένου να καλυφθούν οι ανάγκες της ναυσιπλοΐας. Μεταξύ 1970 και 2012, οι θαλάσσιες μεταφορές πετρελαίου και φυσικού αερίου σχεδόν διπλασιάστηκαν (98 τοις εκατό), αυτές του γενικού φορτίου τετραπλασιάστηκαν (411 τοις εκατό) ενώ οι μεταφορές σιτηρών και ορυκτών σχεδόν πενταπλασιάστηκαν (495 τοις εκατό) . Συμβατικά, περίπου το 90 τοις εκατό του διεθνούς εμπορίου κατ' όγκο μεταφέρεται δια θαλάσσης. Τέλος καταγράφεται και αύξηση στη μέση διάρκεια των ταξιδιών: μεταξύ 2000 και 2013, η εκτιμώμενη ποσότητα διεθνών θαλάσσιων αποστολών μετρημένη σε τόνο-μίλια αυξήθηκε κατά 65 τοις εκατό, από 30.648 σε 50.506 δισεκατομμύρια τόνους- μίλια, ενώ το συνολικό ποσό του διεθνούς φορτίου αυξήθηκε μόνο κατά περίπου 50 τοις εκατό. Αυτή η αύξηση στη μέση διάρκεια των ταξιδιών οφείλεται σε μεγάλο βαθμό στη μεταφορά άνθρακα, σιτηρών και μεταλλευμάτων (Simcock et al, 2016).

Αναφορικά με τον τομέα της ακτοπλοΐας, μετά την εμφάνιση των μεγάλων αεροσκαφών, η θαλάσσια επιβατική κίνηση περιορίστηκε σε πλοία μικρών αποστάσεων και κρουαζιερόπλοια. Κάθε κράτος με κατοικημένα υπεράκτια νησιά και

με δεδομένη την αδυναμία των αεροσκαφών να καλύψουν τις ανάγκες αυτών των περιοχών, η ακτοπλοΐα αποτελεί ουσιαστικά την μοναδική λύση μετακίνησης. Επίσης τα κράτη με πελάγη ή αρχιπελάγη βασίζονται σε μεγάλο βαθμό σε πορθμεία για την εσωτερική μεταφορά επιβατών. Τα διεθνή επιβατηγά πορθμεία είναι ιδιαίτερα σημαντικά στη Βαλτική Θάλασσα, τη Βόρεια Θάλασσα και την Καραϊβική. Τα πλοία Roll-on/roll-off (όπου επιβατικά οχήματα και επιβάτες ταξιδεύουν ταυτόχρονα) έχουν βοηθήσει σημαντικά την ανάπτυξη των μεταφορών μικρών αποστάσεων. Τα πλοία Roll-on/roll-off είναι επίσης σημαντικά για τις τοπικές μεταφορές εμπορευμάτων, ειδικά στην Ευρώπη. Η ανάπτυξη των επιβατικών μεταφορών με πορθμεία διέπεται κυρίως από βελτιώσεις στις εγκαταστάσεις και γενικότερη οικονομική ανάπτυξη. Την τελευταία δεκαετία, για παράδειγμα, η κίνηση στα ελληνικά επιβατηγά πλοία παρέμεινε στάσιμη υπό το πρίσμα της ελληνικής οικονομικής κρίσης, ενώ η κίνηση με επιβατηγά πλοία στην Ινδονησία και τις Φιλιππίνες συνέχισε να αυξάνεται σημαντικά. Οι συνολικοί επιβάτες πορθμείων παγκοσμίως το 2008 και τα περιφερειακά στοιχεία παρουσιάζονται στον Πίνακα 1 (Simcock et al, 2016).

Πίνακας 1 Παγκόσμια μεταφορά επιβατών και οχημάτων στην ακτοπλοΐα (Πηγή ΟΗΕ)

	Passengers	Cars	Buses	Freight vehicles
World traffic volumes (millions of journeys)	2,052	252	677	32
<i>Percentages of world total in each region</i>				
America and Caribbean	14.6	29.7	11.9	2.7
Baltic	10.9	33.7	38.6	24.3
Mediterranean	21.2	14.3	14.9	26.7
North Sea	4.4	7.5	32.4	31.7
Pacific	1.5	0.4	<0.1	1.5

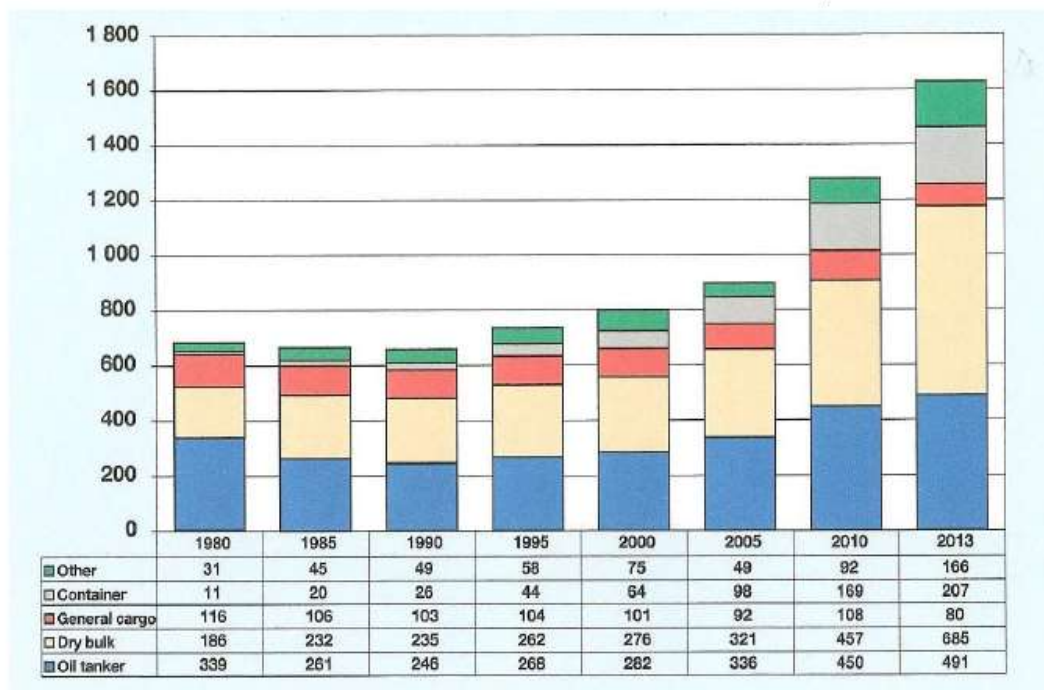
Ο άλλος σημαντικός τομέας θαλάσσιων μεταφορών επιβατών είναι τα κρουαζιερόπλοια. Αν και τα θαλάσσια τουριστικά ταξίδια μπορούν να ανιχνευθούν ήδη από το 1837, η σύγχρονη βιομηχανία κρουαζιερόπλοιων εμφανίστηκε στις δεκαετίες του 1960 και του 1970, σε μια εποχή που τα αεροπορικά ταξίδια επιβατών μεγάλων

αποστάσεων γνώριζαν ιδιαίτερη άνθηση και κυριαρχούσαν στην εν λόγω αγορά. Η απαίτηση της αγοράς για θαλάσσιες διακοπές, οδήγησε στην κατασκευή εξειδικευμένων πλοίων, με λιγότερη έμφαση στην ταχύτητα και περισσότερο στον χώρο της διασκέδασης και χαλάρωσης. Η αγορά από τότε έχει αναπτυχθεί σταθερά και γρήγορα με 3.774.000 ταξίδια το 1990 και 21.556.000 ταξίδια το 2013 (Εικόνα 4). Ο συνολικός κύκλος εργασιών της αγοράς κρουαζιέρας υπολογίζεται σε 37,1 δισεκατομμύρια δολάρια (CMW, 2014). Η ανάπτυξη επιβραδύνθηκε κάπως από το 2008, αλλά συνεχίστηκε στα επόμενα χρόνια.

Πάνω από το ήμισυ της ζήτησης της αγοράς το 2013 προήλθε από τις Ηνωμένες Πολιτείες (51,7%). Η εναπομένουσα ζήτηση καταγράφεται ως 26,6% από την Ευρώπη, 3,6% από την Αυστραλία και τη Νέα Ζηλανδία, 3,4% από τη Βραζιλία, 3,4% από τον Καναδά και 11,3% από τον υπόλοιπο κόσμο. Οι κύριες περιοχές-στόχοι όσον αφορά τα δρομολόγια και την ζήτηση πλοίων για κρουαζιέρες το 2013 αναφέρονται: Καραϊβική (34%), Μεσόγειος (22%), υπόλοιπη Ευρώπη (11%), Αυστραλία (5%)), Αλάσκα (5 %), Νότια Αμερική (4%) Ασία (3%) και άλλες περιοχές (16%) (CLIA, 2014).

Η γενικότερη παραδοχή είναι ότι φαίνεται να υπάρχει μια τάση προς μεγαλύτερα πλοία. Στη Βαλτική Θάλασσα, η Επιτροπή του Ελσίνκι υπολόγισε ότι ο μέσος αριθμός επιβατών στα κρουαζιερόπλοια που καταπλέουν στα λιμάνια της Βαλτικής αυξήθηκε μεταξύ 2006 και 2012 κατά 49%, από 1.099 σε 1.635 (HELCOM, 2014). Το μέγεθος του παγκόσμιου στόλου πλοίων αυξάνεται ραγδαία την περίοδο 2000 – 2013 (Πίνακας 2). Η διατήρηση ενός υψηλού ρυθμού ανάπτυξης μετά την οικονομική κρίση του 2008 οφείλεται στον σύντομο χρόνο παράδοσης μεταξύ της παραγγελίας και της παράδοσης πλοίων.

Πίνακας 2 Συνολικό μέγεθος παγκόσμιου στόλου από το 2008 έως το 2013(πηγή ΟΗΕ).



2.2. Η ελληνική πραγματικότητα

Η ελληνική ακτοπλοΐα περιλαμβάνει τη μετακίνηση επιβατών και εμπορευμάτων στο πλαίσιο ενός πολύπλοκου συστήματος συνδέσεων από ηπειρωτική χώρα σε νησί, νησί με νησί και ηπειρωτική χώρα με ηπειρωτική χώρα. Αν και ο πραγματικός αριθμός των νησιών είναι της τάξης των πολλών χιλιάδων, τα περισσότερα από αυτά τα νησιά είναι πολύ μικρά και ακατοίκητα, και μόνο 69 νησιά θεωρούνται σημαντικά από οικονομική άποψη. Τα νησιά και η ηπειρωτική χώρα εξυπηρετούνται από συνολικά 138 λιμάνια. Κατά σύμβαση, η «ηπειρωτική Ελλάδα» θεωρείται ότι περιλαμβάνει επίσης την Πελοπόννησο και την Εύβοια (συμπεριλαμβανομένων 20 νησιών με την αυστηρή έννοια του όρου, αλλά συνδέονται με την υπόλοιπη Ελλάδα με μικρές γέφυρες). Με βάση το παραπάνω ορισμό, η ηπειρωτική χώρα διαθέτει 42 λιμάνια. Τα υπόλοιπα

λιμάνια (96) θεωρούνται νησιωτικά λιμάνια. Η Κρήτη είναι το μεγαλύτερο νησί της Ελλάδας, με 8 λιμάνια (Psaraftis et al, 1992)

Μια σημαντική παράμετρος της ελληνική ναυτιλίας μικρών αποστάσεων είναι οι γνωστές ως «κύριες γραμμές». Κάθε κύρια γραμμή ορίζεται όχι ως μια προδιαγεγραμμένη ακολουθία επισκέψεων σε λιμάνι, αλλά ως ένα σύνολο γεωγραφικών συστάδων που συνδέονται εσωτερικά με ένα δίκτυο δρομολογίων πλοίων (π.χ. Πειραιάς-Κρήτη, Πειραιάς- Κυκλάδες-Δωδεκάνησα- Κρήτη κ.λπ.) . Υπάρχουν συνολικά 31 κύριες γραμμές, που καλύπτουν την ελληνική επικράτεια. Γενικότερα μπορεί να υπάρχουν περισσότερες από μία κύριες γραμμές που εξυπηρετούν μια συγκεκριμένη γεωγραφική περιοχή (π.χ. για τις Κυκλάδες υπάρχουν 10 διαφορετικές κύριες γραμμές). Εντός μίας συγκεκριμένης κύριας γραμμής, ποικιλίες μεμονωμένων διαδρομών δύναται να εξυπηρετήσουν τα λιμάνια που ανήκουν στην κύρια γραμμή (Psaraftis et al, 1992).

Η ελληνική ακτοπλοΐα μικρών αποστάσεων αποτελείται από πλοία διαφόρων κατηγοριών. Η πρώτη κατηγορία αφορά μόνο τη διακίνηση εμπορευμάτων και κατατάσσεται στη γενική κατηγορία «πλοία τροφοδοσίας». Αυτά μεταφέρουν φορτίο είτε χύμα είτε ως γενικό φορτίο μεταξύ των νησιών και της ηπειρωτικής χώρας. Τα πλοία φορτώνονται συνήθως από τον Πειραιά ή από άλλα μεγάλα λιμάνια της ηπειρωτικής χώρας και διανέμουν φορτίο στα νησιά (γίνεται και η αντίστροφη λειτουργία). Αυτή η κατηγορία πλοίων λειτουργεί κυρίως με ναυλώσεις και δεν περιορίζεται από συγκεκριμένες «κύριες γραμμές» ή δρομολόγια. Αποτελούσε τον κυρίαρχο τρόπο μεταφοράς εμπορευμάτων από και προς τα νησιά κατά το παρελθόν.

Μια δεύτερη μεγάλη κατηγορία πλοίων που αποκτά όλο και μεγαλύτερη σημασία τα τελευταία χρόνια είναι ο στόλος των πορθμείων για τη μεταφορά επιβατών, Ι.Χ., λεωφορείων, μοτοσικλετών και φορτηγών. Τα πλοία αυτά λειτουργούν εντός του συστήματος της «κύριας γραμμής» που περιγράφηκε προηγουμένως. Έχουν ουσιαστικά εκτοπίσει τα παραδοσιακά επιβατηγά ακτοπλοϊκά πλοία που παρείχαν υπηρεσίες στα νησιά τις δεκαετίες του '50 και του '60. Τέτοια πλοία εξακολουθούν να υπάρχουν αλλά ο αριθμός τους μειώνεται σταθερά. Έχοντας χάσει σημαντικό μερίδιο της αγοράς τους από πλοία σε διαδρομές μεγάλων αποστάσεων, φαίνεται να

εκτοπίζονται και σε μικρότερες διαδρομές από ταχύπλοα, όπως υδροπτέρυγα, καταμαράν, κ.λπ. (Ψαραύτης, 2006).

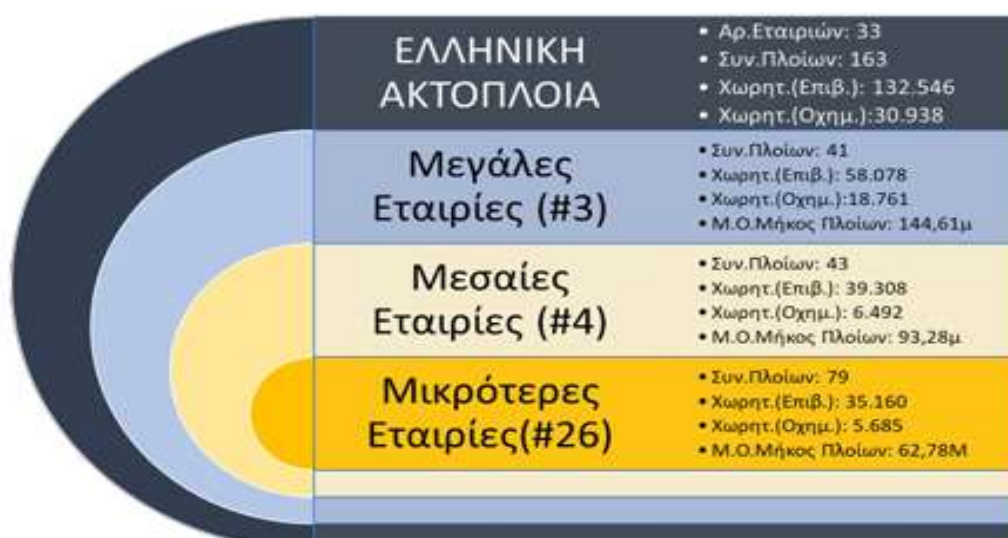
Αυτή η εξέλιξη από τα παραδοσιακά επιβατηγά πλοία σε μικτά επιβατηγά/οχηματαγωγά είναι η πρώτη σημαντική αλλαγή της φύσης των θαλάσσιων μεταφορών μικρών αποστάσεων στην Ελλάδα τα τελευταία χρόνια. Ο μετασχηματισμός πραγματοποιήθηκε κατά τη διάρκεια των τελευταίων 20-30 ετών και οφείλεται σε σημαντικό βαθμό στην ταχεία νησιωτική οικονομική ανάπτυξη και στις σημαντικές βελτιώσεις των υποδομών στα νησιωτικά λιμάνια στα τέλη της δεκαετίας του '60 και στις αρχές της δεκαετίας του '70. Το γεγονός αυτό επέτρεψε για πρώτη φορά και την εμφάνιση φέρι μποτ σε αυτά τα νησιά. Αξίζει να αναφερθεί ότι μέχρι τότε ήταν αδύνατο για επιβατηγά πλοία ακόμη και να ελλιμενιστούν στα περισσότερα νησιά και οι επιβάτες έπρεπε να επιβιβαστούν και να αποβιβαστούν χρησιμοποιώντας μικρά σκάφη που πλησίαζαν το πλοίο καθώς αυτό ήταν αγκυροβολημένο στη μέση του λιμανιού. Η κατασκευή προβλητών στα περισσότερα νησιά τερμάτισε αυτόν τον τρόπο λειτουργίας και άνοιξε την πόρτα στα δρομολόγια των πορθμείων. (Ψαραύτης, 2006).

Στη βάση δεδομένων Martedec καταγράφονται 106 πλοία κατά το έτος 1988. Από αυτά τα 83 είναι ferries. Τα υπόλοιπα 23 πλοία παρέχουν μόνο εξυπηρέτηση επιβατών αλλά δεν μεταφέρουν οχήματα. Δύο πλοία είναι για αμιγώς εμπορευματικές μεταφορές (ro/ro's), με την χωρητικότητα των επιβατών να ανέρχεται στους 12. Όλα τα άλλα είναι πλοία συνδυασμού επιβατών/εμπορευμάτων ή αμιγώς επιβατηγά πλοία. Η μέση χωρητικότητα των πορθμείων είναι 3.015 GRT, με το μεγαλύτερο πλοίο να είναι 11.546 GRT. Η μέση χωρητικότητα επιβατών ανέρχεται περίπου στους 700, με τη μέγιστη χωρητικότητα να είναι 2.300 επιβάτες (APOLLON EXPRESS). Η μέση χωρητικότητα αυτοκινήτων ανέρχεται στα 231, με τη μέγιστη χωρητικότητα να φτάνει στα 490 αυτοκίνητα (ΑΠΤΕΡΑ). Η μέση ταχύτητα των παραπάνω πλοίων υπολογίζεται περίπου στους 16 κόμβους, με τη μέγιστη ταχύτητα να είναι 23 κόμβοι (ΑΡΙΑΔΝΗ).

Η μέση ηλικία είναι σχετικά υψηλή: 19 έτη το 1988 (έναντι 16 ετών το 1985) (Psaraftis et al, 1992).

Κατά το έτος 2023, καταγράφονται 150 επιβατηγά πλοία τα οποία διαμοιράζονται στην εξυπηρέτηση 115 διαφορετικών προορισμών ανά την επικράτεια. Οι 33 διαφορετικές εταιρείες που δραστηριοποιούνται στο χώρο, μπορούν να κατηγοριοποιηθούν με βάση το μέγεθος του στόλου καθώς και άλλα χρηματοοικονομικά χαρακτηριστικά σε κατηγορίες σύμφωνα με την ετήσια έκθεση της XRTC Business Consultants Ltd(εικόνα 2) :

- Μεγάλες:
- Μεσαίες
- Μικρότερες



Εικόνα 2. Κατηγοριοποίηση εταιριών ελληνικής ακτοπλοΐας (Πηγή: XRTC)

Στις μεγάλες εταιρείες οι κυριότεροι εκπρόσωποι είναι οι εταιρείες Attica, ANEK και Μινωικές Γραμμές.

Ο στόλος και η επιβατική κίνηση κάθε εταιρείας καταγράφεται στον πίνακα 1.

Πίνακας 3. Ο στόλος και η επιβατική κίνηση μεγάλων εταιρειών(Πηγή ΧΡΤC)

ΕΤΑΙΡΕΙΕΣ	Αριθμός Πλοίων	Κίνηση Επιβατών	Κίνηση Αυτοκινήτων	Κίνηση Φορτηγών
Attica Group	33	6,100,000	1,000,000	420,000
Minoan Lines	4	906,000	161,000	84,000
ANEK	8	809,000	186,000	115,000
Σύνολο 2022	45	7,815,000	1,347,000	619,000
Σύνολο 2021	43	3,626,129	640,805	498,408
Ποσοστιαία Μεταβολή		116%	110%	24%

Στις μεσαίες εταιρείες κατατάσσονται οι SeaJets, Fast Ferries, Levante Ferries και Golden Star Ferries. Η εταιρεία SeaJets, που αποτελεί και την μεγαλύτερη ιδιωτική ελληνική εταιρεία, αποτελείται από 29 συνολικά πλοία, 24 ταχύπλοα και 5 συμβατικά. Στην FastFerries ανήκουν 5 πλοία, εκ των οποίων 4 ταχύπλοα και ένα συμβατικό. Η εταιρεία δραστηριοποιείται στις γραμμές Ραφήνα – Άνδρος- Τήνος-Μύκονος, Νάξος- Πάρος-Κουφονήσια και Πειραιά-Σύρο-Μύκονο-Νάξο. Η Levante Ferries, που αποτελείται από 6 πλοία, προσφέρει δρομολόγια στις γραμμές Πάτρας/Κυλλήνης – Ζακύνθου-Κεφαλλονιάς, Ιθάκης . Τέλος η Golden Star Ferries, με 2 πλοία προσφέρει υπηρεσίες μετακίνησης επιβατών στη γραμμή Ραφήνα-Κυκλάδες. Τέλος υπάρχουν συνολικά 26 μικρές εταιρείες(πίνακας 4).

Πίνακας 4. Μικρές εταιρείες ελληνικής ακτοπλοΐας(Πηγή ΣΕΕΝ)

	ΕΤΑΙΡΙΑ	ΣΥΜΒΑΤΙΚΑ	ΤΑΧΥΠΛΟΙΑ	ΣΥΝΟΛΟ	ΜΕΙΟΣ ΟΡΟΣ ΗΛΙΚΙΑΣ	ΓΡΑΜΜΕΣ 2023
1	AEGEAN FLYING DOLPHINS		3	3	35	Πειραιάς- Αίγινα- Αγκίστρι, Βόλος-Σποράδες
2	AEGEAN SEA LINES	1		1	43	Πειραιάς- Δυτικές & Νότιες Κυκλάδες
3	AGIA MARINI	1		1	43	Χίος- Ψαρά
4	AGIOREITIKES GRAMMES	7	2	9		Ουρανούπολη- Άγιο Όρος
5	ALPHA LINES		1	1	21	Πειραιάς - Πόρος- Ύδρα- Σπέτσες
6	ANE KALYMNOU	2		2	26	Κάλυμνος-Αστυπάλαια-Κως-Μισσηράκι- Ψέριμος
7	ANENDYK	6		6	36	Χώρα Σφακίων- Γαύδος
8	ANES FERRIES	4		4	40	Πειραιάς-Αίγινα, Βόλος- Εύβοια-Σποράδες
9	ANMEZ AE (ZANTE FERRIES)	2		2	35	Πειραιάς - Δυτικές & Νότιες Κυκλάδες, Αλεξανδρούπολη- Σαμοθράκη- Λήμνος
10	DODEKANISSOS SEAWAYS	1	2	3	26	Δωδεκάνησα/Ανατολικό Αιγαίο
11	GOUTOS LINES	1		1	51	Λαύριο- Κέα- Κύθνος
12	IONIAN SEA FERRIES	5		5		Κεφαλλονιά-Ζάκυνθος
13	IONION P.LINES	1	1	1	34	Αστακός-Κεφαλονιά-Ιθάκη-Λευκάδα-Ζάκυνθος
14	JOY CRUISES	1	1	2	32	Κέρκυρα-Παζοί
15	KARYSTIA	1		1	38	Λαύριο- Κέα- Κύθνος
16	KERKYRA LINES	9	1	10	30	Ηγουμενίτσα- Κέρκυρα-Παζοί- Διαπόντια
17	KERKYRA SEAWAYS	6		6	30	Ηγουμενίτσα- Κέρκυρα-Παζοί
18	LAFASI	2		2		Κάλυμνος- Κως- Λέρος- Μπουντρούμι
19	NOBEL MARITIME	3		3	39	Ηγουμενίτσα- Κέρκυρα- Ιταλία-Αλβανία
20	SAOS FERRIES	4		4	20	Ρόδος- Χάλκη-Σύμη- Τήλος-Νίσυρος-Κως-Κάλυμνος
21	SARONIC FERRIES	5		5	26	Πειραιάς- Αίγινα, Αγκίστρι, Μέθανα, Πόρος
22	SEBECO LINES	2		2	5	Ρόδος- Σύμη
23	SKYROS N.E.	1		1	36	Κύμη- Σκύρος
24	SMALL CYCLADES LINES	1		1	37	Αμοργός- Κουφονήσια- Νότος-Ιος-Θήρα
25	STAR GEM	1		1	19	Κως- Νίσυρος- Ρόδος- Τήλος
26	TRITON FERRIES	2		2	36	Λαύριο- Κέα- Κύθνος, Νεάπολη- Κύθηρα-Αντικύθηρα
		69	11	79	32	

Αναφορικά με τη ζήτηση των γραμμών ακτοπλοΐας που αναφέρθηκαν προηγουμένως, οι γραμμές Πειραιάς-Κυκλάδες, Αργοσαρωνικός και Ραφήνας-Εύβοιας-Άνδρου-Τήνου παρουσιάζουν τον μεγαλύτερο αριθμό επιβατών, ενώ η μεγαλύτερη ζήτηση αναφορικά με τη μετακίνηση οχημάτων καταγράφεται στη γραμμή Πειραιάς-Κρήτη(πίνακες 5,6). Τέλος αξίζει να αναφερθεί ότι το σύνολο της ζήτησης σχεδόν όλων των ακτοπλοϊκών γραμμών παρουσιάζει αυξητικές τάσεις, με μία μόνο ανεπαίσθητη μείωση στη ζήτηση οχημάτων της γραμμής Βόλου-Βόρειων Σποράδων-Κύμης.

Πίνακας 5. Κίνηση επιβατών ακτοπλοΐας (πηγή ΕΛΣΤΑΤ)

ΑΚΤΟΠΛΟΙΚΕΣ ΓΡΑΜΜΕΣ	2021	2022	Δ%
Αργοσαρωνικού	1,600,690	2,276,353	42%
Πειραιώς-Πελοποννήσου	52,914	66,645	26%
Πειραιώς-Κρήτης	1,191,120	1,711,462	44%
Πειραιώς-Κρήτης-Δωδεκανήσου	306,193	757,742	147%
Πειραιώς-Δωδεκανήσου	588,974	736,349	25%
Πειραιώς-Δυτικών Κυκλάδων	914,231	1,123,248	23%
Πειραιώς-Ανατολικών Κυκλάδων	1,825,150	2,635,580	44%
Πειραιώς-Μυκόνου-Τήνου-Σάμου	643,908	920,183	43%
Πειραιώς-Χίου-Μυτιλήνης	437,503	570,071	30%
Πατρών-Ακαρνανίας-Ιονίων Νήσων	415,491	574,962	38%
Ραφήνας-Ευβοίας-Άνδρου-Τήνου	1,527,804	2,195,265	44%
Βόλου-Βορείων Σποράδων-Κύμης	663,924	758,467	14%
Ζακύνθου-Κυλλήνης	671,027	900,838	34%
Κυλλήνη-Πόρου Κεφαλληνίας	360,195	447,743	24%
Λοιπές	1,647,247	1,850,766	12%
ΣΥΝΟΛΟ	12,846,371	17,525,674	36%

Πίνακας 6. Κίνηση οχημάτων στην ακτοπλοΐα (πηγή ΕΛΣΤΑΤ)

ΑΚΤΟΠΛΟΙΚΕΣ ΓΡΑΜΜΕΣ	2021	2022	Δ%
Αργοσαρωνικού	282,983	297,162	5%
Πειραιώς-Πελοποννήσου	18,819	21,116	12%
Πειραιώς-Κρήτης	514,239	560,531	9%
Πειραιώς-Κρήτης-Δωδεκανήσου	51,000	59,300	16%
Πειραιώς-Δωδεκανήσου	222,186	247,862	12%
Πειραιώς-Δυτικών Κυκλάδων	155,438	158,758	2%
Πειραιώς-Ανατολικών Κυκλάδων	347,132	367,130	6%
Πειραιώς-Μυκόνου-Τήνου-Σάμου	163,897	189,954	16%
Πειραιώς-Χίου-Μυτιλήνης	154,605	177,329	15%
Πατρών-Ακαρνανίας-Ιονίων Νήσων	153,504	190,063	24%
Ραφήνας-Ευβοίας-Άνδρου-Τήνου	377,116	453,718	20%
Βόλου-Βορείων Σποράδων-Κύμης	167,357	166,257	-1%
Ζακύνθου-Κυλλήνης	241,318	286,661	19%
Κυλλήνη-Πόρου Κεφαλληνίας	150,480	164,048	9%
Λοιπές	278,091	320,401	15%
ΣΥΝΟΛΟ	3,278,165	3,660,290	12%

2.3. Καθυστερήσεις δρομολογίων. Ανίχνευση αιτιών μέσω βιβλιογραφικής ανασκόπησης

Οι λόγοι καθυστέρησης στην ακτοπλοΐα δεν διαφέρουν σημαντικά από τους λόγους στα υπόλοιπα μέσα μεταφοράς. Ίσως για αυτό το λόγο δεν καταγράφεται σημαντική – συστηματική διεθνής ή εγχώρια βιβλιογραφική επισκόπηση.

Ο καιρός αποτελεί εν γένει το σημαντικότερο λόγο εκτροπής από τους προσχεδιασμένους χρόνους των δρομολογίων. Τα πλοία που πλέουν ειδικά σε ανοιχτές θάλασσα είναι εκτεθειμένα σε αντίξοες καιρικές συνθήκες κατά το μεγαλύτερο μέρος του ταξιδιού. Οποιοσδήποτε δυσμενείς καιρικές συνθήκες, όπως άνεμος, τυφώνες, καταιγίδες κ.λπ., δύναται να καθυστερήσουν το πλοίο και να προκαλέσουν χρονική επιβάρυνση. Ακόμη και κατά τον ελλιμενισμό, εάν οι κλιματικές συνθήκες είναι δύσκολες, οι δραστηριότητες φόρτωσης και εκφόρτωσης είναι ιδιαιτέρως χρονοβόρες, προκαλώντας καθυστέρηση στην έξοδο του πλοίου από το λιμάνι.

Υπάρχουν ορισμένες περίοδοι του χρόνου που η επιβατική κίνηση είναι πολύ μεγαλύτερη από το συνηθισμένο. Στον ελλαδικό χώρο, οι περίοδοι είναι αυτοί είναι κατά κύριο λόγο το καλοκαίρι λόγω διακοπών, αλλά και οι περίοδοι των Χριστουγέννων και του Πάσχα. Δεν υπάρχει τυπική ή ομοιόμορφη περίοδος αιχμής για τα λιμάνια παγκοσμίως. Η περίοδος αιχμής διαφέρει από λιμάνι σε λιμάνι, από χώρα σε χώρα και από περιοχή σε περιοχή.

Καθυστερήσεις επίσης καταγράφονται κυρίως κατά της περιόδους αιχμής, όχι τόσο λόγω της αυξημένης κίνησης αλλά κυρίως λόγω της υποδομής των λιμανιών, που αδυνατούν να υποστηρίξουν τον ελλιμενισμό αριθμού πλοίων πέρα από ένα συγκεκριμένο όριο, ιδιαίτερα κατά τους καλοκαιρινούς μήνες. Καθυστερήσεις επίσης καταγράφονται λόγω απεργίας λιμενεργατών ή ναυτικών αλλά και λόγω απροσδόκητων γεγονότων που μπορούν να συμβούν εντός ή εκτός πλοίου και δύναται να επηρεάσουν την ασφάλεια των επιβατών (π.χ. τηλεφώνημα για βόμβα).

Σημαντικές καθυστερήσεις προκύπτουν τέλος και από πιθανές μηχανικές βλάβες η προσκρούσεις πλοίων. Τα πλοία προτού επιστρέψουν στο προγραμματισμένο δρομολόγιο τους μετά από ένα τέτοιο γεγονός, υποχρεούνται να ελεγχθούν από τις λιμενικές αρχές, προκειμένου να πιστοποιήσουν την καταλληλότητα ναυσιπλοΐα τους.

3. Μηχανική μάθηση

Ο όρος μηχανική μάθηση αναφέρεται στον αυτοματοποιημένο εντοπισμό σημαντικών προτύπων στα δεδομένα. Τις τελευταίες δύο δεκαετίες έχει εξελιχθεί σε ιδιαίτερα διαδεδομένο εργαλείο σε πλήθος εργασιών που απαιτούν εξαγωγή πληροφοριών από μεγάλα σύνολα δεδομένων. Σημαντικό πλήθος νέων τεχνολογιών βασίζεται στη μηχανική μάθηση: οι μηχανές αναζήτησης μαθαίνουν να επιλέγουν τα καλύτερα αποτελέσματα με βάση την αναζήτηση του χρήστη, τα anti-spam λογισμικά μαθαίνουν να φιλτράρουν τα μηνύματα ηλεκτρονικού ταχυδρομείου ενώ οι συναλλαγές με πιστωτικές κάρτες προστατεύονται από λογισμικό που μαθαίνει να αναγνωρίζει προσπάθειες απάτης. Οι ψηφιακές φωτογραφικές μηχανές μαθαίνουν να ανιχνεύουν πρόσωπα και οι έξυπνες εφαρμογές προσωπικής βοήθειας στα smartphones μαθαίνουν να αναγνωρίζουν φωνητικές εντολές. Τα αυτοκίνητα είναι εξοπλισμένα με συστήματα πρόληψης ατυχημάτων χρησιμοποιώντας αλγόριθμους μηχανικής μάθησης. Η μηχανική μάθηση χρησιμοποιείται επίσης ευρέως σε επιστημονικές εφαρμογές όπως η βιοπληροφορική, η ιατρική και η αστρονομία.

3.1. Δέντρα απόφασης

Η εξόρυξη δεδομένων χρησιμοποιείται για την εξαγωγή χρήσιμων πληροφοριών από μεγάλα σύνολα δεδομένων καθώς και για την οπτικοποίηση αυτών παρέχοντας ευκολότερες ερμηνείες στις εξαγόμενες πληροφορίες. Τα δέντρα απόφασης που εισήχθησαν για πρώτη φορά στη δεκαετία του 1960 είναι μια από τις πιο αποτελεσματικές μεθόδους για την εξόρυξη δεδομένων (εικόνα 3). Έχουν χρησιμοποιηθεί ευρέως σε διάφορους κλάδους καθώς είναι εύχρηστα, χωρίς αμφισημίες (ambiguity) και ανθεκτικά ακόμη και σε απουσία τιμών ή ελλειπών δεδομένων. Τόσο διακριτές όσο και συνεχείς μεταβλητές μπορούν να χρησιμοποιηθούν είτε ως μεταβλητές στόχου είτε ως ανεξάρτητες μεταβλητές. Τα τελευταία χρόνια, η μεθοδολογία των δέντρων αποφάσεων έχει καταστεί δημοφιλής και στην ιατρική έρευνα. Ένα παράδειγμα ιατρικής χρήσης των δέντρων απόφασης είναι η διάγνωση μιας ιατρικής κατάστασης κάνοντας χρήση του μοτίβου των συμπτωμάτων (Fallon et al, 2013).

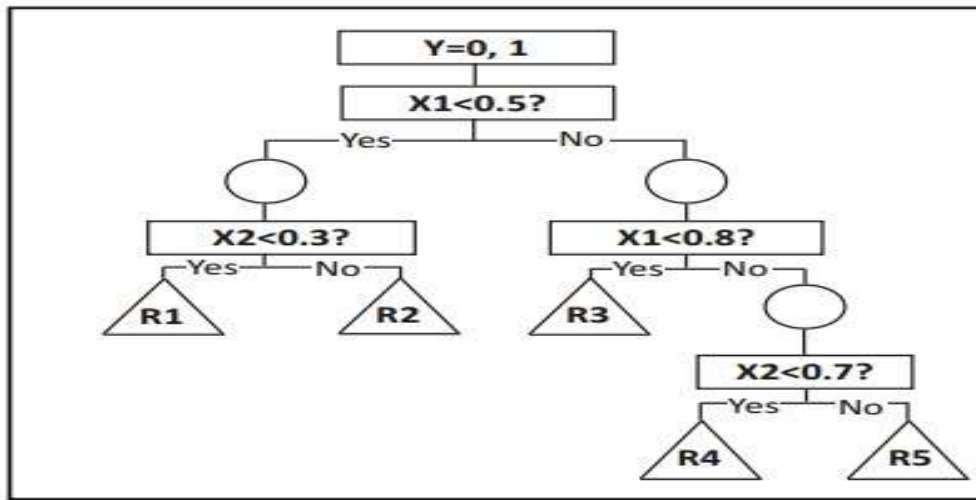
Συνήθεις χρήσεις των μοντέλων δέντρων απόφασης είναι η χρήση για:

- **Επιλογή μεταβλητών.** Ο αριθμός των μεταβλητών που παρακολουθούνται τακτικά σε ερευνητικά περιβάλλοντα έχει αυξηθεί δραματικά με την εισαγωγή της ηλεκτρονικής αποθήκευσης δεδομένων. Πολλές από αυτές τις μεταβλητές είναι οριακής συνάφειας με το πρόβλημα. Όπως η σταδιακή επιλογή μεταβλητών στην ανάλυση παλινδρόμησης, έτσι και τα δέντρα απόφασης μπορούν να χρησιμοποιηθούν για την επιλογή των πιο σχετικών μεταβλητών εισόδου που θα πρέπει να χρησιμοποιηθούν για να σχηματιστούν τα κατάλληλα μοντέλα δέντρων αποφάσεων, τα οποία μπορούν στη συνέχεια να χρησιμοποιηθούν για τη διατύπωση ερευνητικών υποθέσεων.

- **Εκτίμηση της σχετικής σημασίας των μεταβλητών.** Μόλις εντοπιστεί ένα σύνολο σχετικών μεταβλητών, η έρευνα εστιάζεται στην σημαντικότητα των μεταβλητών. Γενικά, η σημασία της μεταβλητής υπολογίζεται με βάση τη μείωση της ακρίβειας του μοντέλου (ή την καθαρότητα των κόμβων στο δέντρο) όταν αφαιρείται η μεταβλητή.

Στις περισσότερες περιπτώσεις, όσο περισσότερες εγγραφές επηρεάζει μια μεταβλητή, τόσο μεγαλύτερη καθίσταται η σημασία της.

- **Χειρισμός τιμών που λείπουν.** Μια συνηθισμένη - αλλά εσφαλμένη - μέθοδος χειρισμού δεδομένων που λείπουν είναι ο αποκλεισμός περιπτώσεων για αυτές τις τιμές, γεγονός που είναι ταυτόχρονα αναποτελεσματικό και εισάγει τον κίνδυνο να εισαχθεί μεροληψία στην ανάλυση. Η ανάλυση του δέντρου αποφάσεων μπορεί να αντιμετωπίσει δεδομένα που λείπουν με δύο τρόπους: είτε να ταξινομήσει τις τιμές που λείπουν ως ξεχωριστή κατηγορία ή να χρησιμοποιήσει ένα ενσωματωμένο μοντέλο δέντρου που ορίζει τη μεταβλητή τις ελλειπείς τιμές ως μεταβλητή στόχο, να προβλέψει την τιμή και να αντικαταστήσει τα δεδομένα που λείπουν με την προβλεπόμενη τιμή.
- **Πρόβλεψη.** Αυτή είναι μια από τις πιο σημαντικές χρήσεις των μοντέλων δέντρων απόφασης. Χρησιμοποιώντας το μοντέλο δέντρου που προέρχεται από ιστορικά δεδομένα, είναι εύκολο να προβλεφθούν αποτελέσματα για μελλοντικές εγγραφές.
- **Χειρισμός δεδομένων.** Υπερβολικά πολλές κατηγορίες μιας κατηγορικής μεταβλητής ή λοξά συνεχή δεδομένα είναι κάτι σύνηθες στην ιατρική έρευνα. Υπό αυτές τις συνθήκες, τα μοντέλα δέντρων απόφασης μπορούν να βοηθήσουν στην επιλογή του τρόπου με τον οποίο θα συμπτύσσονται καλύτερα οι κατηγορικές μεταβλητές σε έναν πιο διαχειρίσιμο αριθμό κατηγοριών ή πώς θα υποδιαιρούνται οι πολύ λοξές μεταβλητές σε εύρη.



Εικόνα 3. Παράδειγμα δέντρου απόφασης (Πηγή: Song et al, 2015)

Τα κύρια συστατικά ενός δέντρου απόφασης είναι:

Κόμβοι. Υπάρχουν τρεις τύποι κόμβων. (α) Ένας ριζικός κόμβος, που ονομάζεται επίσης κόμβος απόφασης που αντιπροσωπεύει μια επιλογή η οποία προκαλεί την υποδιαίρεση όλων των εγγραφών σε δύο ή περισσότερα αμοιβαία αποκλειόμενα υποσύνολα. (β) Οι εσωτερικοί κόμβοι που αντιπροσωπεύουν μία από τις πιθανές επιλογές που είναι διαθέσιμες σε εκείνο το σημείο της δενδρικής δομής. Το πάνω άκρο του κόμβου συνδέεται με τον γονικό του κόμβο και το κάτω άκρο συνδέεται με τους θυγατρικούς κόμβους ή τους κόμβους φύλλων του. (γ) Οι κόμβοι φύλλων, που ονομάζονται επίσης τερματικοί κόμβοι, αντιπροσωπεύουν το τελικό αποτέλεσμα ενός συνδυασμού αποφάσεων ή γεγονότων.

Κλάδοι. Οι κλάδοι αντιπροσωπεύουν τυχαία αποτελέσματα ή περιστατικά που προέρχονται από ριζικούς κόμβους και εσωτερικούς κόμβους. Ένα μοντέλο δέντρου αποφάσεων σχηματίζεται χρησιμοποιώντας μια ιεραρχία κλάδων. Κάθε διαδρομή από τον ριζικό κόμβο μέσω των εσωτερικών κόμβων σε έναν κόμβο φύλλου αντιπροσωπεύει έναν κανόνα απόφασης ταξινόμησης. Τα μονοπάτια του δέντρου αποφάσεων μπορούν επίσης να αναπαρασταθούν ως κανόνες «αν-τότε». Για

παράδειγμα, «αν προκύψουν συνθήκη 1 και συνθήκη 2 και συνθήκη ... και συνθήκη k, τότε προκύπτει το αποτέλεσμα j. »

Διαχωρισμός. Μόνο οι μεταβλητές εισόδου που σχετίζονται με τη μεταβλητή στόχο χρησιμοποιούνται για τον διαχωρισμό των γονικών κόμβων σε θυγατρικούς κόμβους της μεταβλητής στόχου. Μπορούν να χρησιμοποιηθούν τόσο διακριτές μεταβλητές όσο και μεταβλητές συνεχούς εισόδου (οι οποίες συμπίπτουν σε δύο ή περισσότερες κατηγορίες). Κατά την κατασκευή του μοντέλου πρέπει πρώτα να προσδιοριστούν οι πλέον σημαντικές μεταβλητές εισόδου και στη συνέχεια να χωριστούν οι εγγραφές στον ριζικό κόμβο και στους επόμενους εσωτερικούς κόμβους σε δύο ή περισσότερες κατηγορίες με βάση την τιμή των μεταβλητών. Χαρακτηριστικά που σχετίζονται με τον βαθμό σημαντικότητας των θυγατρικών κόμβων χρησιμοποιούνται για την επιλογή μεταξύ διαφορετικών δυνητικών μεταβλητών εισόδου (Patel et al 2012). Αυτά τα χαρακτηριστικά περιλαμβάνουν την εντροπία, τον δείκτη Gini, το σφάλμα ταξινόμησης, το κέρδος πληροφορίας, την αναλογία κέρδους. Η διαδικασία διαχωρισμού συνεχίζεται έως ότου ικανοποιηθούν προκαθορισμένα κριτήρια. Στις περισσότερες περιπτώσεις, δεν χρησιμοποιούνται όλες οι πιθανές μεταβλητές εισόδου για τη δημιουργία του μοντέλου και σε ορισμένες περιπτώσεις μια συγκεκριμένη μεταβλητή εισόδου μπορεί να χρησιμοποιηθεί πολλές φορές σε διαφορετικά επίπεδα του δέντρου αποφάσεων.

Διακοπή. Η πολυπλοκότητα και η ευρωστία είναι ανταγωνιστικά χαρακτηριστικά των μοντέλων που πρέπει να λαμβάνονται ταυτόχρονα υπόψη κατά την κατασκευή ενός στατιστικού μοντέλου. Όσο πιο περίπλοκο είναι ένα μοντέλο, τόσο λιγότερο αξιόπιστο θα είναι όταν χρησιμοποιείται για την πρόβλεψη μελλοντικών εγγραφών. Ένα τέτοιο δέντρο αποφάσεων θα ήταν υπερβολικά προσαρμοσμένο στις υπάρχουσες παρατηρήσεις έχοντας λίγες εγγραφές σε κάθε φύλλο, επομένως δεν θα είχε την ικανότητα να προβλέψει αξιόπιστα τις μελλοντικές περιπτώσεις και, ως εκ τούτου, θα είχε κακή γενίκευση (δηλαδή, έλλειψη ευρωστίας). Για να αποφευχθεί αυτό, πρέπει να εφαρμόζονται κανόνες διακοπής κατά την κατασκευή. Οι κοινές παράμετροι που χρησιμοποιούνται στους κανόνες διακοπής περιλαμβάνουν: (α) τον ελάχιστο αριθμό εγγραφών σε ένα φύλλο. (β) τον ελάχιστο αριθμό εγγραφών σε έναν κόμβο πριν από τη διαίρεση και (γ) το βάθος (δηλαδή αριθμός βημάτων) οποιουδήποτε φύλλου από τον

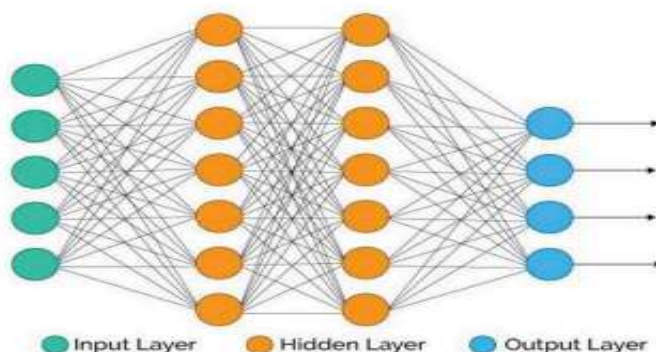
κόμβο ρίζας. Οι παράμετροι διακοπής πρέπει να επιλέγονται με βάση τον στόχο της ανάλυσης και τα χαρακτηριστικά του συνόλου δεδομένων που χρησιμοποιείται. Ως εμπειρικός κανόνας, οι Berry και Linoff (1999) συνιστούν την αποφυγή της υπερπροσαρμογής και της υποπροσαρμογής ορίζοντας την αναλογία στόχου των εγγραφών σε έναν κόμβο φύλλων μεταξύ 0.25 και 1.00% του πλήρους συνόλου δεδομένων εκπαίδευσης.

Κλάδεμα. Σε ορισμένες περιπτώσεις, οι κανόνες διακοπής δεν λειτουργούν καλά. Ένας εναλλακτικός τρόπος για τη δημιουργία ενός μοντέλου δέντρου αποφάσεων είναι να αναπτυχθεί πρώτα ένα μεγάλο δέντρο και μετά να κλαδευτεί στο βέλτιστο μέγεθος αφαιρώντας τους κόμβους που παρέχουν λιγότερες πρόσθετες πληροφορίες (Hastie et al, 2001). Μια κοινή μέθοδος επιλογής του καλύτερου δυνατού υποδέντρου από πολλούς υποψηφίους είναι να ληφθεί υπόψη η αναλογία των εγγραφών με πρόβλεψη σφαλμάτων (δηλαδή, η αναλογία στην οποία η προβλεπόμενη εμφάνιση του στόχου είναι εσφαλμένη). Άλλες μέθοδοι για την επιλογή της καλύτερης εναλλακτικής είναι η χρήση ενός συνόλου δεδομένων επικύρωσης (δηλαδή, διαίρεση του δείγματος στα δύο και δοκιμή του μοντέλου που αναπτύχθηκε στο σύνολο δεδομένων εκπαίδευσης στο σύνολο δεδομένων επικύρωσης) ή, για μικρά δείγματα, διασταυρούμενη επικύρωση (δηλαδή, διαίρεση του δείγματος σε 10 ομάδες δοκιμάζοντας το μοντέλο που αναπτύχθηκε από τις 9 ομάδες στη 10^η επαναλαμβανόμενη, υπολογίζοντας τον μέσο όρο των σωστών ή εσφαλμένων προβλέψεων). Υπάρχουν δύο είδη κλαδέματος, το προκλάδεμα (προς τα εμπρός) και το μετακλάδεμα (πίσω κλάδεμα). Το προ-κλάδεμα χρησιμοποιεί δοκιμές Chi-square (Zibran, 2012) ή μεθόδους προσαρμογής πολλαπλής σύγκρισης για την αποτροπή τη δημιουργίας μη σημαντικών κλάδων. Το μετα-κλάδεμα χρησιμοποιείται μετά τη δημιουργία ενός πλήρους δέντρου αποφάσεων για την αφαίρεση κλαδιών με τρόπο που βελτιώνει την ακρίβεια της συνολικής ταξινόμησης όταν εφαρμόζεται στο σύνολο δεδομένων επικύρωσης.

3.2. Νευρωνικά δίκτυα

Τα Τεχνητά Νευρωνικά Δίκτυα (ANN=Artificial Neural Networks), ή πιο απλά νευρωνικά δίκτυα, είναι νέα συστήματα και υπολογιστικές μέθοδοι για μηχανική μάθηση, επίδειξη γνώσης και, τέλος, εφαρμογή της γνώσης που αποκτάται για τη μεγιστοποίηση των αποκρίσεων εξόδου σύνθετων συστημάτων. Το τεχνητό νευρωνικό δίκτυο είναι ένα μοντέλο επεξεργασίας δεδομένων που βασίζεται στον τρόπο με τον οποίο τα βιολογικά νευρικά συστήματα, όπως ο εγκέφαλος, επεξεργάζονται δεδομένα. Επικεντρώνονται στη νευρωνική δομή του εγκεφαλικού φλοιού των θηλαστικών, αλλά σε πολύ μικρότερη κλίμακα. Πολλοί ειδικοί της τεχνητής νοημοσύνης πιστεύουν ότι τα τεχνητά νευρωνικά δίκτυα είναι η καλύτερη και ίσως η μόνη ελπίδα για το σχεδιασμό μιας ευφυούς μηχανής.

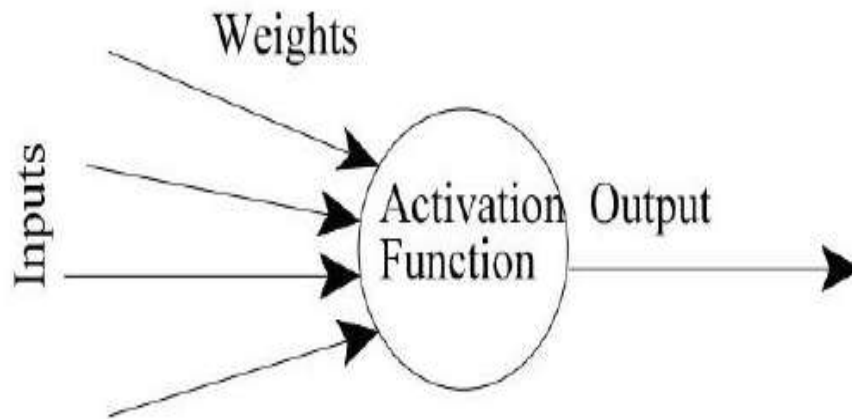
Τα τεχνητά νευρωνικά δίκτυα σχεδιάζονται με τον ίδιο τρόπο όπως ο ανθρώπινος εγκέφαλος, με νευρώνες κόμβους διασυνδεδεμένους με τρόπο που να μοιάζει με ιστό. Οι νευρώνες είναι δισεκατομμύρια κύτταρα που αποτελούν τον ανθρώπινο εγκέφαλο. Κάθε νευρώνας αποτελείται από ένα κυτταρικό σώμα που επεξεργάζεται τις πληροφορίες μεταφέροντας αυτές προς και από τον εγκέφαλο (εισροές και εκροές). Η κύρια ιδέα τέτοιων δικτύων είναι εμπνευσμένη από τον τρόπο που λειτουργεί το βιολογικό νευρωνικό σύστημα, για την επεξεργασία δεδομένων και πληροφοριών με σκοπό τη μάθηση και τη δημιουργία γνώσης. Το βασικό στοιχείο αυτής της ιδέας είναι η δημιουργία νέων δομών για το σύστημα επεξεργασίας πληροφοριών. Η αρχιτεκτονική ενός τεχνητού νευρωνικού δικτύου φαίνεται στην εικόνα 4 (Bre et al 2018).



Εικόνα 4. Η αρχιτεκτονική ενός ANN (Bre et al 2018)

Το σύστημα αποτελείται από έναν μεγάλο αριθμό διασυνδεδεμένων στοιχείων επεξεργασίας που ονομάζονται νευρώνες οι οποίοι συνεργάζονται για να λύσουν ένα πρόβλημα και να μεταδώσουν πληροφορίες μέσω συνάψεων (ηλεκτρομαγνητικές συνδέσεις). Οι νευρώνες είναι στενά συνδεδεμένοι και οργανωμένοι σε στρώματα. Το επίπεδο εισόδου λαμβάνει τα δεδομένα, ενώ το επίπεδο εξόδου δημιουργεί το τελικό αποτέλεσμα. Μεταξύ των δύο, συνήθως τοποθετούνται ένα ή περισσότερα μυστικά στρώματα. Αυτή η διάταξη καθιστά δύσκολη την πρόβλεψη ή τη γνώση της ακριβούς ροής των δεδομένων.

Κάθε σύνδεση έχει ένα βάρος σύνδεσης και κάθε νευρώνας έχει μια τιμή κατώφλιου και μια συνάρτηση ενεργοποίησης (Balakrishnan et al. 2019). Υπολογίζεται εάν κάθε είσοδος έχει θετικό ή αρνητικό βάρος με βάση το πρόσημο του βάρους της εισόδου. Το βάρος επηρεάζει την ένταση του σήματος σε μια σύνδεση (Liu et al. 2018). Οι νευρώνες έχουν ένα κατώφλι πάνω από το οποίο μεταδίδεται το σήμα. Η τιμή ενεργοποίησης είναι το σταθμισμένο άθροισμα κάθε μεμονωμένης εισόδου και η έξοδος παράγεται με βάση το σήμα από αυτήν την τιμή ενεργοποίησης. Η σχέση μεταξύ βάρους κάθε στοιχείου, εισόδου και εξόδου ενός τυπικού ANN φαίνεται στην εικόνα 5.

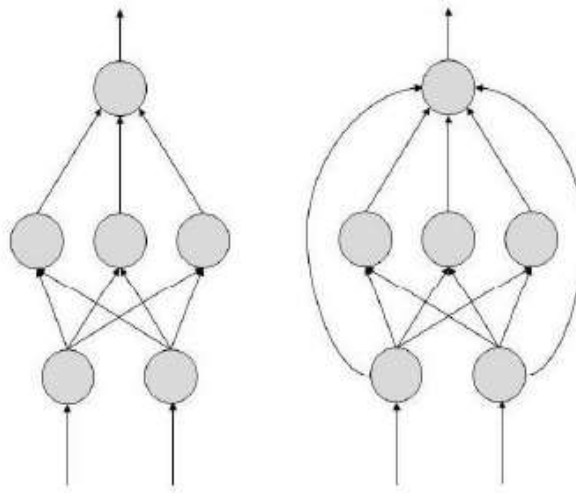


Εικόνα 5. Βάρος κάθε στοιχείου και είσοδος - έξοδος ενός τυπικού συστήματος ANN

Σε αυτά τα δίκτυα, εάν ένα κύτταρο καταστραφεί, άλλα κύτταρα μπορούν να αναπληρώσουν την απουσία του και να συμβάλουν στην αναδόμηση του. Αυτού του είδους τα δίκτυα διαθέτουν την ικανότητα της μάθησης. Βασικά, η ικανότητα μάθησης είναι το πιο σημαντικό χαρακτηριστικό ενός τέτοιου ευφυούς συστήματος. Ένα σύστημα μάθησης είναι πιο ευέλικτο και εύκολο στον προγραμματισμό, με αποτέλεσμα να μπορεί να ανταποκρίνεται καλύτερα σε νέα προβλήματα. Τα τεχνητά νευρωνικά δίκτυα, όπως και οι άνθρωποι, μαθαίνουν χρησιμοποιώντας διαφορετικά παραδείγματα και ένα νευρωνικό δίκτυο δημιουργείται για να εκτελεί συγκεκριμένες εργασίες, όπως ο εντοπισμός προτύπων και η κατηγοριοποίηση πληροφοριών, κατά τη διάρκεια μιας μαθησιακής διαδικασίας.

Τα τεχνητά νευρωνικά δίκτυα χρησιμοποιούνται όλο και περισσότερο στον έλεγχο ή τη μοντελοποίηση συστημάτων που έχουν άγνωστες ή πολύ περίπλοκες εσωτερικές δομές. Η μάθηση σε αυτά τα συστήματα είναι προσαρμοστική, δηλαδή, χρησιμοποιώντας παραβολές, το βάρος των συνάψεων αλλάζει με τέτοιο τρόπο ώστε το σύστημα να παράγει τη σωστή απόκριση εάν δοθούν νέες εισροές (Wu and Feng 2018). Σε ένα νευρωνικό δίκτυο δίνεται ένα σύνολο εισόδων και των αντίστοιχων εξόδων τους όταν εκπαιδεύεται (χρησιμοποιώντας μία από τις μεθόδους εκπαίδευσης).

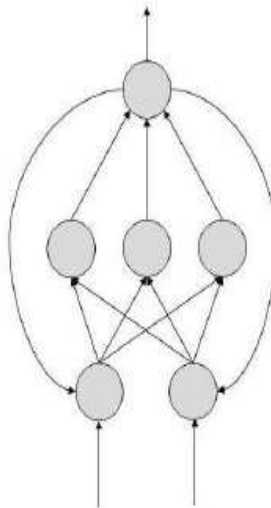
Υπάρχουν δύο βασικές τοπολογίες νευρωνικών δικτύων γνωστές ως **FeedForward** και **Feedback**. Στα συστήματα FeedForward, δεν υπάρχουν βρόχοι ανάδρασης. Μια μονάδα στέλνει πληροφορίες σε άλλη από την οποία όμως δεν λαμβάνει καμία αντίστοιχη πληροφορία. Οι είσοδοι και οι εξόδοι παραμένουν σταθερές. Κάθε μονάδα λαμβάνει πληροφορίες εισόδου από τις μονάδες στα αριστερά της και οι εξόδοι πολλαπλασιάζονται με το βάρος της κάθε σύνδεσης. Έτσι, λαμβάνονται τα αποτελέσματα εξόδου που σχετίζονται με το βάρος της κάθε σύνδεσης. Η δημιουργία προτύπων, η αναγνώριση και η ταξινόμηση αποτελούν ορισμένες εφαρμογές της μεθόδου. Τέτοια συστήματα χρησιμοποιούνται σε εφαρμογές όταν το αποτέλεσμα είναι γνωστό εκ των προτέρων. Πολλές εμπορικές εφαρμογές, όπως π.χ. computer vision, βασίζονται σε τέτοιες αρχιτεκτονικές. Η αρχιτεκτονική των συστημάτων τοπολογιών νευρωνικών δικτύων FeedForward παρουσιάζεται στην εικόνα 6 (Mitra and Paul 2017).



Εικόνα 6. Η αρχιτεκτονική των συστημάτων τοπολογιών νευρωνικών δικτύων FeedForward (Mitra and Paul 2017).

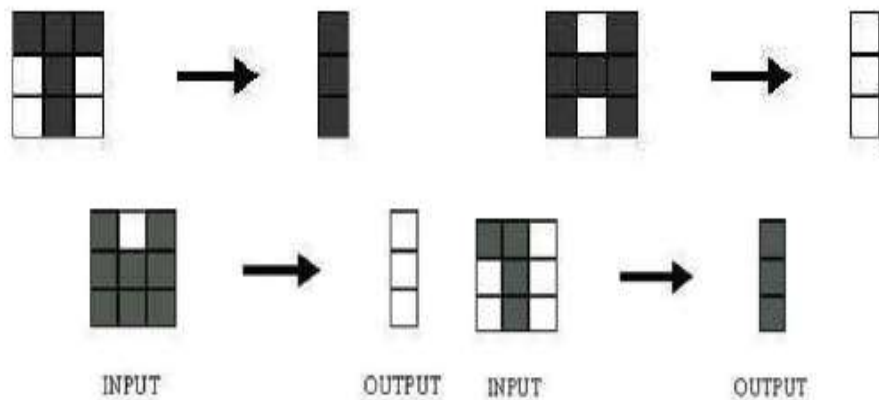
Στα συστήματα FeedBack, χρησιμοποιούνται μνήμες με διευθυνσιοδότηση περιεχομένου. Η διαδικασία μάθησης των νευρωνικών δικτύων χρησιμοποιώντας μια διαδικασία ανάδρασης γίνεται με τη σύγκριση της εξόδου ενός δικτύου με την έξοδο που είναι επιθυμητή και αναμενόμενη. Η διαφορά μεταξύ αυτών των δύο εξόδων χρησιμοποιείται για την αλλαγή και την τροποποίηση των βαρών των συνδέσεων

μεταξύ των μονάδων δικτύου. Η αρχιτεκτονική των συστημάτων ANN του FeedBack φαίνεται στην εικόνα 7 (Mitra and Paul 2017).



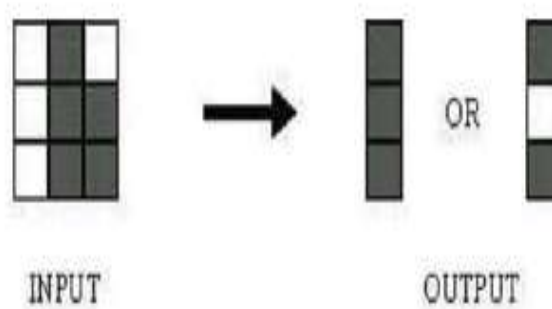
Εικόνα 7. Η αρχιτεκτονική των συστημάτων FeedBack (Mitra and Paul 2017).

Η προσαρμοστική μάθηση είναι το σημαντικότερο πλεονεκτήματα των συστημάτων ANN, δηλαδή η ικανότητα του συστήματος να “μάθει” πώς να εκτελεί εργασίες με βάση τις πληροφορίες που δίνονται. Π.χ., το σύστημα μπορεί να μάθει την αναγνώριση των γραμμάτων T και H χρησιμοποιώντας την προσαρμοστικότητα και τη δομή του σχεδίου. Η διαδικασία φαίνεται στην εικόνα 8 (Qasim 2013).



Εικόνα 8. Η αναγνώριση των γραμμάτων T και H με τη χρήση της προσαρμοστικότητας στα συστήματα ANN (Qasim 2013).

Το κύριο πλεονέκτημα της χρήσης ενός νευρωνικού δικτύου σε οποιοδήποτε από τα παραπάνω ζητήματα είναι η εξαιρετική ικανότητα του νευρωνικού δικτύου στη μάθηση καθώς και η σταθερότητα του απέναντι σε μικρές διαταραχές. Για παράδειγμα, εάν χρησιμοποιούνται κανονικές μέθοδοι για την ανίχνευση ενός ανθρώπινου χειρόγραφου, εσφαλμένες τιμές μπορούν να ληφθούν εξαιτίας μίας ελαφριάς δόνηση του χεριού, ενώ ένα σωστά εκπαιδευμένο νευρωνικό δίκτυο μπορεί να ανταποκριθεί ακόμη και σε περίπτωση τέτοιας διαταραχής.. Οι μαθησιακές ικανότητες των συστημάτων ANN παρουσιάζονται στην εικόνα 9 (Qasim 2013).



Εικόνα 9. Οι μαθησιακές ικανότητες των συστημάτων ANN (Qasim 2013).

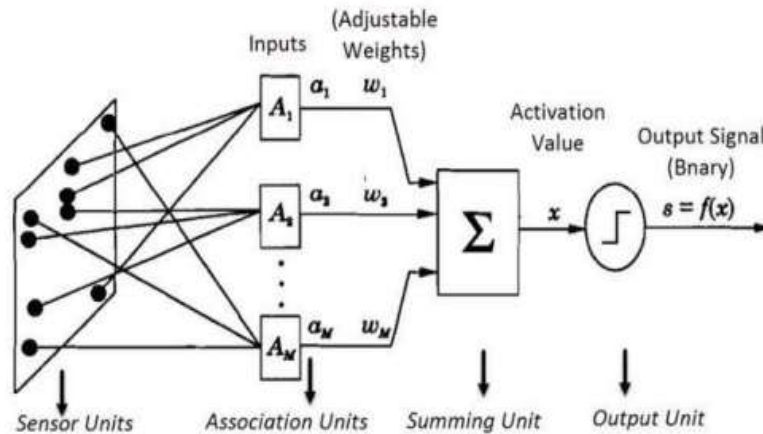
Οι υπολογισμοί δύναται να πραγματοποιηθούν με παράλληλο τρόπο ενώ χρησιμοποιείται και ειδικό υλικό για την παροχή των εφαρμογών απόδοσης σε πραγματικό χρόνο.

Μια μερική αστοχία του δικτύου οδηγεί σε αντίστοιχη υποβάθμιση της απόδοσής του, αν και ορισμένες δυνατότητες του δικτύου μπορεί να παραμείνουν ακόμη και σε περίπτωση βλάβης. Λόγω της υψηλής ταχύτητας των νευρωνικών δικτύων τόσο στην επεξεργασία όσο και στην ανάλυση δεδομένων, ο χρόνος που απαιτείται για την ανακάλυψη της βέλτιστης δομής μειώνεται (Wu and Feng 2018).

Παρά τα πλεονεκτήματα που έχουν τα νευρωνικά δίκτυα έναντι των συμβατικών συστημάτων, έχουν επίσης μειονεκτήματα που προσπαθούν να ελαχιστοποιήσουν οι ερευνητές σε αυτόν τον τομέα, όπως:

1. Δεν υπάρχουν συγκεκριμένοι κανόνες ή οδηγίες για το σχεδιασμό ενός δικτύου.
2. Στην περίπτωση προβλημάτων μοντελοποίησης, η φυσική του προβλήματος από μόνη της δεν μπορεί να γίνει κατανοητή χρησιμοποιώντας νευρωνικά δίκτυα.
3. Η ακρίβεια των αποτελεσμάτων εξαρτάται πολύ από το μέγεθος του training σετ.
4. Η εκπαίδευση του δικτύου μπορεί να είναι δύσκολη ή ακόμα και αδύνατη.
5. Δεν είναι εύκολη η πρόβλεψη των μελλοντικών επιδόσεων του δικτύου (γενίκευση) (Walczak 2019).

Σήμερα, τα νευρωνικά δίκτυα χρησιμοποιούνται σε ποικίλες εφαρμογές, όπως ζητήματα αναγνώρισης προτύπων, τα οποία περιλαμβάνουν προβλήματα όπως η αναγνώριση ομιλίας, η επεξεργασία εικόνας καθώς και προβλήματα κατηγοριοποίησης όπως η ταξινόμηση κειμένου ή εικόνας. Άλλες εφαρμογές των νευρωνικών δικτύων περιλαμβάνουν συστήματα ανάλυσης κινδύνου, έλεγχο drones, ανάλυση ποιότητας, δοκιμές δωματίων έκτακτης ανάγκης, εξερεύνηση πετρελαίου και αερίου, συστήματα ανίχνευσης πέδησης φορτηγών, εκτίμηση κινδύνου δανείου, φασματική ανίχνευση, ανίχνευση φαρμάκων. Επίσης χρησιμοποιούνται σε διαδικασίες βιομηχανικού ελέγχου, διαχείριση σφαλμάτων, φωνητική αναγνώριση, ανίχνευση ασθενειών, απομακρυσμένη ανάκτηση πληροφοριών, ανίχνευση υποβρυχίων ναρκών, τρισδιάστατα αντικείμενα, ανίχνευση προσώπου, αναγνώριση μοτίβων, επεξεργασία σήματος κ.α. Η εφαρμογή των ANN σε συστήματα επεξεργασίας σήματος παρουσιάζεται στην εικόνα 10 (Cichocki, Unbehauen, and Swiniarski 1993).



Εικόνα 10. Η εφαρμογή των ANN σε συστήματα επεξεργασίας σήματος (Cichocki, Unbehauen και Swiniarski 1993)

3.3. Κανόνες συσχέτισης

Οι κανόνες συσχέτισης είναι δηλώσεις if/then που βοηθούν στην ανακάλυψη σχέσεων μεταξύ δεδομένων σε μια βάση δεδομένων, σχεσιακή ή μη, καθώς και σε άλλους χώρους αποθήκευσης πληροφοριών. Οι κανόνες συσχέτισης χρησιμοποιούνται για τον εντοπισμό σχέσεων μεταξύ αντικειμένων που χρησιμοποιούνται συχνά μαζί. Εφαρμογές των κανόνων συσχέτισης εντοπίζονται κυρίως στην ανάλυση δεδομένων καλαθιού, ταξινόμηση, cross-marketing, ομαδοποίηση, σχεδιασμός καταλόγου, ανάλυση κεφαλαίου απώλειας κ.λπ. Για παράδειγμα, εάν ένας πελάτης αγοράσει ψωμί, μπορεί επίσης να αγοράσει βούτυρο. Εάν ένας πελάτης αγοράσει φορητό υπολογιστή, τότε πιθανώς να αγοράσει και κάρτα μνήμης. Υπάρχουν **δύο βασικά κριτήρια** που χρησιμοποιούν οι κανόνες συσχέτισης, **η υποστήριξη** και **η εμπιστοσύνη**. Συνήθως απαιτούνται κανόνες συσχέτισης για την ικανοποίηση μιας ελάχιστης υποστήριξης και μιας ελάχιστης εμπιστοσύνης που καθορίζεται από τον χρήστη.

$$\begin{array}{l} \text{Rule: } X \Rightarrow Y \begin{cases} \text{Support} = \frac{\text{freq}(X,Y)}{N} \\ \text{Confidence} = \frac{\text{freq}(X,Y)}{\text{freq}(X)} \end{cases} \end{array}$$

Εικόνα 11. Υποστήριξη και εμπιστοσύνη(Kumbhare et al., 2014)

Ο αλγόριθμος AIS (Artificial Immune System) ήταν ο πρώτος αλγόριθμος που προτάθηκε από τους Agrawal, Imielinski και Swami για κανόνες συσχέτισης. Επικεντρώνεται στη βελτίωση της ποιότητας των βάσεων δεδομένων μαζί με την απαραίτητη λειτουργικότητα για την επεξεργασία ερωτημάτων υποστήριξης αποφάσεων. Σε αυτόν τον αλγόριθμο δημιουργείται μόνο ένας συνακόλουθος κανόνας συσχέτισης, γεγονός που σημαίνει ότι η συνέπεια αυτών των κανόνων περιέχει μόνο ένα στοιχείο, για παράδειγμα, δημιουργούνται κανόνες όπως $X \cap Y \Rightarrow Z$ αλλά όχι κανόνες όπως $X \Rightarrow Y \cap Z$ (Zhao et al, 2003).

Στον AIS οι βάσεις δεδομένων σαρώνονται πολλές φορές για να ληφθούν τα συχνά στοιχειοσύνολα. Προκειμένου να γίνει ο αλγόριθμος πιο αποτελεσματικός, εισάγεται μια μέθοδος εκτίμησης ώστε να κλαδευτούν τα υποψήφια σύνολα αντικειμένων που δεν έχουν καμία πιθανότητα να είναι μεγάλα, και κατά συνέπεια καθίσταται περιττή η προσπάθεια μέτρησης αυτών των στοιχείων. Δεδομένου ότι όλα τα υποψήφια σύνολα στοιχείων και τα συχνά στοιχειοσύνολα θεωρείται ότι αποθηκεύονται στην κύρια μνήμη, η διαχείριση μνήμης μέσω AIS προτείνεται όταν η μνήμη δεν είναι αρκετή.

Ο αριθμός υποστήριξης κάθε μεμονωμένου στοιχείου συγκεντρώνεται κατά το πρώτο πέρασμα στη βάση δεδομένων. Με βάση την ελάχιστη μέτρηση υποστήριξης, τα στοιχεία των οποίων η υποστήριξη είναι λιγότερη από την ελάχιστη τιμή, εξαλείφονται από τη λίστα των αντικειμένων. Τα υποψήφια σύνολα 2 στοιχείων δημιουργούνται με την επέκταση συχνών συνόλων 1 στοιχείου με άλλα στοιχεία. Κατά το δεύτερο πέρασμα στη βάση δεδομένων, ο αριθμός υποστήριξης αυτών των υποψήφιων

συνόλων 2 στοιχείων συσσωρεύεται και ελέγχεται με βάση το όριο υποστήριξης. Παρομοίως, αυτά τα υποψήφια σύνολα $(k+1)$ -στοιχείων δημιουργούνται επεκτείνοντας τα συχνά σύνολα k -ειδών με άλλα στοιχεία.

Το μειονέκτημα αυτού του αλγορίθμου είναι ότι δημιουργούνται πολλά υποψήφια σύνολα στοιχείων που τελικά αποδεικνύονται ιδιαίτερα μικρά. Επίσης απαιτεί αρκετό χρόνο και υπολογιστική δύναμη. Τέλος ο αλγόριθμος απαιτεί πολλά περάσματα σε ολόκληρη τη βάση δεδομένων.

Ο αλγόριθμος Apriori χρησιμοποιείται επίσης για εξόρυξη συχνών στοιχειοσυνόλων και εκμάθηση κανόνων συσχέτισης. Σε αυτόν τον αλγόριθμο, τα συχνά υποσύνολα επεκτείνονται ένα στοιχείο κάθε φορά και αυτό το βήμα είναι γνωστό ως διαδικασία δημιουργίας υποψηφίων. Στη συνέχεια, τα υποψήφια στοιχεία ελέγχονται με βάση τα δεδομένα. Για να μετρήσει αποτελεσματικά τα σύνολα υποψηφίων αντικειμένων, ο Apriori χρησιμοποιεί τη μέθοδο αναζήτησης πρώτα κατά πλάτος και μια δομή δέντρου κατακερματισμού.

Για να αντιμετωπιστούν τα μειονεκτήματα του αλγορίθμου Apriori, χρησιμοποιείται ο αλγόριθμος FP-growth. Ο FP-growth απαιτεί την κατασκευή ενός FP-tree. Για αυτό, απαιτούνται δύο περάσματα. Ο FP-growth χρησιμοποιεί τη στρατηγική διαίρει και βασίλευε. Απαιτεί δύο σαρώσεις στη βάση δεδομένων. Αρχικά υπολογίζει μια λίστα με συχνά στοιχεία ταξινομημένα κατά συχνότητα με φθίνουσα σειρά (F-List) κατά την πρώτη σάρωση της βάσης δεδομένων. Στη δεύτερη σάρωση, η βάση δεδομένων συμπίεζεται σε ένα δέντρο FP. Ο αλγόριθμος εκτελεί εξόρυξη σε FP-δέντρο αναδρομικά. Τα συχνά σύνολα στοιχείων δημιουργούνται με δύο μόνο περάσματα στη βάση δεδομένων και χωρίς καμία διαδικασία δημιουργίας υποψηφίων (Gagandeep et al, 2013).

4. Σχεδιασμός Συστήματος

Στο παρόν κεφάλαιο περιγράφεται ο σχεδιασμός και η λειτουργία του υπό ανάπτυξη συστήματος.

4.1. Ανάκτηση Δεδομένων

Για την καταγραφή των απαιτούμενων στοιχείων και τη συλλογή των δεδομένων επιλέχθηκαν συγκεκριμένα πλοία, κατηγοριοποιημένα ως εξής:

- Πλοία με απευθείας προορισμό, χωρίς ενδιάμεσα λιμάνια: Αριάδνη, Ελευθέριος Βενιζέλος, Ελυρος, Κίσσαμος.
- Πλοία με ενδιάμεσα λιμάνια και ημερήσια διαδρομή: Δήλος, Πάρος.
- Πλοία με ενδιάμεσα λιμάνια και δρομολόγια πέραν του εικοσιτετράωρου: Χίος, Μύκονος.
- Ημερήσια δρομολόγια με πολλαπλά λιμάνια, που εκτελούνται από ταχύπλοα τύπου Catamaran: Dodekanisos Express.

Στη συνέχεια, πραγματοποιήθηκε αναζήτηση και καταγραφή των δρομολογίων των επιλεγμένων πλοίων, καθώς και των προβλεπόμενων ωρών άφιξης και αναχώρησης. Η συλλογή των δεδομένων έγινε μέσω των επίσημων ιστοσελίδων των ναυτιλιακών εταιρειών, όπως αυτές ανακοίνωσαν το πρόγραμμα των δρομολογίων τους. Οι ναυτιλιακές εταιρείες και τα αντίστοιχα πλοία τους παρουσιάζονται στον ακόλουθο πίνακα:

Πίνακας 7. Ναυτιλιακή ανά πλοίο

Αριάδνη (ARIADNE) – HELLENIC SEAWAYS – Atica Group
Χίος (CHIOS) – BLUE STAR FERRIES – Atica Group
Δήλος (DELOS) – BLUE STAR FERRIES – Atica Group
Δωδεκάνησος Express (DODEKANISOS EXPRESS) – DODEKANISOS SEA WAYS
Ελευθέριος Βενιζέλος (EL. VENIZELOS) – ANEK LINES – Atica Group
Ελυρος (ELYROS) – ANEK LINES – Atica Group
Κίσσαμος (KISSAMOS) – ANEK LINES – Atica Group
Μύκονος (MYCONOS) – BLUE STAR FERRIES – Atica Group
Πάρος (PAROS) – BLUE STAR FERRIES – Atica Group

Αφού ολοκληρώθηκε η καταγραφή του προγράμματος των δρομολογίων, διαπιστώθηκε ότι δεν ήταν διαθέσιμα δεδομένα μέσω API ή αρχείων CSV. Κατόπιν

επικοινωνίας με τις ναυτιλιακές εταιρείες, δεν κατέστη δυνατή η λήψη των δεδομένων των δρομολογίων, ενώ η αρμόδια υπηρεσία θαλάσσιων συγκοινωνιών του Υπουργείου Ναυτιλίας επιβεβαίωσε ότι δεν τηρεί τέτοια στοιχεία στο αρχείο της. Επιπλέον, αν και εντοπίστηκε ιδιωτική πλατφόρμα που παρέχει τις σχετικές πληροφορίες, το κόστος απόκτησης των δεδομένων κρίθηκε απαγορευτικό.

Ως εκ τούτου, η μόνη διαθέσιμη λύση ήταν η καθημερινή καταγραφή των δεδομένων μέσω αυτοματοποιημένης διαδικασίας. Για την υλοποίηση αυτής της διαδικασίας, ενεργοποιήθηκε ένας διακομιστής Linux σε συσκευή Raspberry Pi 4 Model B. Αναπτύχθηκε ένα αυτοματοποιημένο σενάριο (automation script) σε γλώσσα προγραμματισμού Python, αξιοποιώντας τη βιβλιοθήκη Playwright της Microsoft. Το εργαλείο αυτό χρησιμοποιήθηκε για τη συλλογή δεδομένων σε τακτά χρονικά διαστήματα, αποθηκεύοντας στιγμιότυπα οθόνης (.png) από έναν διαδικτυακό διακομιστή που εμφάνιζε τη θέση και την τρέχουσα διαδρομή κάθε πλοίου.

Παράλληλα, πραγματοποιήθηκε καθημερινή καταγραφή μετεωρολογικών δεδομένων από αξιόπιστη διαδικτυακή πηγή, με έμφαση στη μέση θερμοκρασία και την ταχύτητα του ανέμου. Τα συλλεχθέντα δεδομένα καταχωρήθηκαν σε αρχεία Excel, ώστε να μπορούν να εισαχθούν ως σύνολα δεδομένων (data sets) για περαιτέρω ανάλυση.

Για κάθε πλοίο, η θέση του καταγραφόταν μέσω της υπηρεσίας Marine Traffic. Οι σχετικοί σύνδεσμοι για την απεικόνιση της θέσης των πλοίων είναι οι ακόλουθοι:

- Αριάδνη: <https://marinetraffic.live/vessels.php/ariadne-position/240580000/#9135262>
- Χίος: <https://marinetraffic.live/vessels.php/blue-star-chios-position/240672000/#9215555>
- Δήλος: <https://marinetraffic.live/vessels.php/blue-star-delos-position/241087000/#9565039>
- Δωδεκάνησος Express: <https://marinetraffic.live/vessels.php/dodekanisos-express-position/237176400/#0>
- Ελευθέριος Βενιζέλος: <https://marinetraffic.live/vessels.php/el-venizelos-position/237628000/#7907673>
- Ελυρος: <https://marinetraffic.live/vessels.php/elyros-position/240685000/#9178599>
- Κίσσαμος: <https://marinetraffic.live/vessels.php/kissamos-position/239672000/#9035876>

- Μύκονος: <https://marinetraffic.live/vessels.php/blue-star-myconos-position/240389000/#9208679>
- Πάρος: <https://marinetraffic.live/vessels.php/blue-star-paros-position/239924000/#9241774>

Όλα τα αρχεία εικόνων (.png) που συλλέχθηκαν αρχειοθετήθηκαν και στη συνέχεια καταχωρήθηκαν σε αρχεία Excel, προκειμένου να χρησιμοποιηθούν ως σύνολα δεδομένων για την περαιτέρω ανάλυση και επεξεργασία της έρευνας.

4.2. Ανάλυση Δεδομένων

Συνολικά καταγράφηκαν 2053 δρομολόγια για το χρονικό διάστημα από Ιούλιο έως Νοέμβριο του 2024. Τα δεδομένα που ανακτήθηκαν για κάθε δρομολόγιο ήταν:

- Month: Ο αύξοντας αριθμός του μήνα που πραγματοποιήθηκε το δρομολόγιο.
- Ship: Το όνομα του πλοίου που πραγματοποίησε το δρομολόγιο.
- Arrival: Ο αύξοντας αριθμός της ώρας της προβλεπόμενης άφιξης
- Departure: Ο αύξοντας αριθμός της ώρας της προβλεπόμενης αναχώρησης
- Time Arr: Ο αύξοντας αριθμός της ώρας της πραγματικής άφιξης
- Time Departure: Ο αύξοντας αριθμός της ώρας της πραγματικής αναχώρησης
- Celsius: Η θερμοκρασία σε βαθμούς Κελσίου την ημέρα πραγματοποίησης του δρομολογίου.
- Wind: Η ένταση του ανέμου σε μποφόρ την ημέρα πραγματοποίησης του δρομολογίου.

Με βάση τις τιμές των Time Departure και Departure υπολογίστηκε σε λεπτά ο χρόνος καθυστέρησης. Η υπολογιζόμενη καθυστέρηση κατηγοριοποιήθηκε σε τέσσερα διακριτά διαστήματα και ειδικότερα:

- 0-15 λεπτά: Μικρή
- 15-30: Μέτρια
- 30-60: Μεγάλη
- >60: Πολύ μεγάλη

Προκειμένου να αντληθούν περαιτέρω πληροφορίες για την ακεραιότητα και εγκυρότητα των δεδομένων αυτά εισήχθησαν στο freeware λογισμικό WEKA. Το Waikato Environment for Knowledge Analysis (WEKA) δημιουργήθηκε για να καλύψει την ανάγκη ενός ενιαίου περιβάλλοντος εργασίας, το οποίο θα επέτρεπε στους ερευνητές να έχουν εύκολη πρόσβαση σε σύγχρονες τεχνικές μηχανικής μάθησης. Όταν το έργο ξεκίνησε το 1992, οι αλγόριθμοι μάθησης ήταν διασκορπισμένοι σε διάφορες γλώσσες προγραμματισμού, χρησιμοποιούνταν σε διαφορετικές πλατφόρμες και υποστήριζαν πολλαπλές μορφές δεδομένων. Η προσπάθεια συγκέντρωσης αυτών των αλγορίθμων για συγκριτική ανάλυση σε διάφορα σύνολα δεδομένων ήταν

ιδιαίτερα απαιτητική. Το WEKA σχεδιάστηκε όχι μόνο ως μια συλλογή εργαλείων για αλγορίθμους μάθησης, αλλά και ως ένα ολοκληρωμένο πλαίσιο, όπου οι ερευνητές θα μπορούσαν να αναπτύξουν νέους αλγορίθμους χωρίς να ανησυχούν για τεχνικά ζητήματα, όπως η διαχείριση δεδομένων και η αξιολόγηση των μοντέλων (Hall et al, 2009).

Το WEKA θεωρείται σήμερα ένα από τα πιο σημαντικά συστήματα στον τομέα της εξόρυξης δεδομένων και της μηχανικής μάθησης. Έχει γίνει ευρέως αποδεκτό τόσο στον ακαδημαϊκό χώρο όσο και στον επιχειρηματικό κόσμο, καθιστώντας το ένα από τα πιο διαδεδομένα εργαλεία για έρευνα στην εξόρυξη δεδομένων. Το συνοδευτικό του βιβλίο αποτελεί ένα από τα πιο δημοφιλή διδακτικά εγχειρίδια στον τομέα και συχνά αναφέρεται σε επιστημονικές δημοσιεύσεις για τη μηχανική μάθηση. Η επιτυχία του αυτή οφείλεται σε μεγάλο βαθμό στο γεγονός ότι διατέθηκε ως λογισμικό ανοιχτού κώδικα. Η ελεύθερη πρόσβαση στον πηγαίο κώδικα έδωσε τη δυνατότητα σε μια ενεργή κοινότητα να αναπτυχθεί και συνέβαλε στη δημιουργία πολλών έργων που είτε ενσωματώνουν είτε επεκτείνουν το WEKA.

Τον Μάιο του 1998 κυκλοφόρησε η τελική έκδοση του συστήματος βασισμένου σε TCL/TK (WEKA 2.3) και, στα μέσα του 1999, παρουσιάστηκε το WEKA 3.0, το οποίο ήταν γραμμένο εξ ολοκλήρου σε Java. Αυτή η μη γραφική έκδοση του WEKA συνόδευε την πρώτη έκδοση του βιβλίου για την εξόρυξη δεδομένων των Witten και Frank [34]. Τον Νοέμβριο του 2003, κυκλοφόρησε η σταθερή έκδοση WEKA 3.4, ενόψει της δημοσίευσης της δεύτερης έκδοσης του βιβλίου [35]. Στο διάστημα μεταξύ των εκδόσεων 3.0 και 3.4, αναπτύχθηκαν οι τρεις κύριες γραφικές διεπαφές χρήστη.

Το 2005, η ομάδα ανάπτυξης του WEKA έλαβε το βραβείο SIGKDD Data Mining and Discovery Service Award [22], το οποίο αναγνώρισε τη μακροχρόνια παρουσία και τη μεγάλη διάδοση του WEKA. Το 2006, η εταιρεία Pentaho Corporation έγινε κύριος χορηγός του λογισμικού και το ενσωμάτωσε ως στοιχείο εξόρυξης δεδομένων και προγνωστικής ανάλυσης στη σουίτα επιχειρηματικής ευφυΐας της. Σήμερα, η Pentaho αποτελεί ενεργό συνεισφέροντα στη βάση κώδικα του WEKA, ενώ ο πρώτος συγγραφέας είναι ο κύριος συντηρητής του λογισμικού. Την περίοδο συγγραφής αυτού του κειμένου, η πιο πρόσφατη έκδοση του WEKA είναι η 3.6, η οποία κυκλοφόρησε τον Δεκέμβριο του 2008 και, σύμφωνα με το σύστημα αρίθμησης εκδόσεων άρτιου-περιττού αριθμού, θεωρείται μια σταθερή έκδοση με πλήρη χαρακτηριστικά.

Μετά την εισαγωγή των δεδομένων στο WEKA παρατηρήθηκε μία μεγάλη ανισοκατανομή μεταξύ των κατηγοριοποιήσεων των κλάσεων, όπως αυτές αναφέρθηκαν παραπάνω, με αποτέλεσμα η συντριπτική πλειοψηφία των παραδειγμάτων να καταλήγει στη κλάση μικρή. Για την αντιμετώπιση του εν λόγω προβλήματος χρησιμοποιήθηκε το φίλτρο resample του WEKA. Το φίλτρο "Resample" στο WEKA αποτελεί ένα χρήσιμο εργαλείο για την αναδιαμόρφωση ενός συνόλου δεδομένων, επιτρέποντας τη δημιουργία μιας νέας δειγματοληπτικής κατανομής. Χρησιμοποιείται κυρίως για τη διαχείριση ανισόρροπων δεδομένων και την αποφυγή της μεροληψίας των μοντέλων μηχανικής μάθησης. Το φίλτρο αυτό εφαρμόζει μια διαδικασία επαναδειγματοληψίας (resampling), όπου επιλέγονται νέες εγγραφές από το αρχικό σύνολο δεδομένων είτε με είτε χωρίς αντικατάσταση. Οι κύριες παράμετροι που μπορούν να ρυθμιστούν είναι:

- **Bias to Uniform Class:** Ενισχύει την ισοκατανομή των κλάσεων, βελτιώνοντας τη μάθηση σε ανισόρροπα δεδομένα.
- **Sample Size Percentage:** Καθορίζει το ποσοστό του αρχικού συνόλου δεδομένων που θα περιέχει το νέο δείγμα.
- **Random Seed:** Ελέγχει την τυχαιότητα της διαδικασίας για αναπαραγώγιμα αποτελέσματα.

Η επεξεργασία των δεδομένων αυτής της μελέτης πραγματοποιήθηκε με χρήση ταξινόμησης, και συγκεκριμένα μέσω του ταξινομητή J48 που προσφέρεται από το Weka. Ο J48 αποτελεί έναν αλγόριθμο κατασκευής δέντρου απόφασης, βασισμένο στο C4.5, το οποίο αποτελεί επέκταση του ID3. Επιπλέον, είναι γνωστός και ως στατιστικός ταξινομητής. Ο C4.5 αναπτύχθηκε από τον Ross Quinlan ως επέκταση του ID3 και είναι ένας από τους πιο διαδεδομένους αλγορίθμους μηχανικής μάθησης για επιβλεπόμενη ταξινόμηση.

Τα βασικά χαρακτηριστικά του J48 είναι:

- **Δημιουργία Δέντρων Απόφασης:** Ο αλγόριθμος κατασκευάζει ένα δέντρο απόφασης με βάση τα δεδομένα εκπαίδευσης.

- **Διαχείριση Δεδομένων με Ελλείψεις:** Μπορεί να χειριστεί ελλιπή δεδομένα, χρησιμοποιώντας εκτιμήσεις για τις απουσιάζουσες τιμές.
- **Κλάδεμα (Pruning):** Περιλαμβάνει διαδικασίες κλαδέματος για τη μείωση της πολυπλοκότητας του δέντρου και την αποφυγή υπερεφαρμογής (overfitting).
- **Διαχείριση Συνεχών και Διακριτών Δεδομένων:** Ο J48 μπορεί να επεξεργαστεί τόσο αριθμητικά όσο και κατηγορικά δεδομένα.
- **Διαχωρισμός βάσει Εντροπίας:** Χρησιμοποιεί το κριτήριο **Information Gain** για την επιλογή των χαρακτηριστικών που θα διαχωρίσουν τα δεδομένα στους κόμβους του δέντρου.

Ο αλγόριθμος J48 κατασκευάζει ένα δέντρο απόφασης χρησιμοποιώντας το κριτήριο του **Information Gain** ή του **Gain Ratio** για να επιλέξει το βέλτιστο χαρακτηριστικό σε κάθε στάδιο διάσπασης. Οι κόμβοι του δέντρου καθορίζονται από αυτή τη μέτρηση, η οποία στοχεύει στην ελαχιστοποίηση της αβεβαιότητας (entropy) των δεδομένων. Ωστόσο, σε περιπτώσεις μεγάλου αριθμού χαρακτηριστικών, η επιλογή του βέλτιστου κόμβου ενδέχεται να οδηγήσει σε αυξημένη πολυπλοκότητα και υπερπροσαρμογή. Η μέθοδος **Best First Search (BFS)** χρησιμοποιείται ως στρατηγική αναζήτησης για την επιλογή των κόμβων με τον πιο αποδοτικό τρόπο. Το BFS επιλέγει να εξετάσει πρώτα τον κόμβο που παρουσιάζει τη μεγαλύτερη ευρετική αξία, μειώνοντας έτσι τον αριθμό των διερευνήσεων που απαιτούνται για τη βέλτιστη επιλογή χαρακτηριστικών. Η διαδικασία αυτή μπορεί να βελτιώσει την απόδοση του J48 μέσω:

- **Μείωσης της υπολογιστικής πολυπλοκότητας:** Η BFS επιτρέπει την εξερεύνηση μόνο των πιο υποσχόμενων κόμβων, μειώνοντας την ανάγκη εξέτασης όλων των πιθανών διασπάσεων.

- **Πρόληψης της υπερπροσαρμογής (overfitting):** Η εστίαση στις βέλτιστες διασπάσεις αποτρέπει την κατασκευή υπερβολικά βαθιών δέντρων που μπορεί να αποτυγχάνουν στη γενίκευση νέων δεδομένων.
- **Βελτίωσης της ακρίβειας:** Επιλέγοντας τους κόμβους με την υψηλότερη διακριτική ικανότητα, η BFS συμβάλλει στη δημιουργία ενός αποδοτικότερου δέντρου απόφασης.

Με βάση τον J48 ως βασικότερα κριτήρια κατηγοριοποίησης την προβλεπόμενη ώρα αναχώρησης και κατόπιν το λιμάνι άφιξης. Συνοπτικά τα αποτελέσματα της ανάλυσης παρουσιάζονται παρακάτω, ενώ η αναλυτική παρουσίαση εκτέλεσης του αλγορίθμου βρίσκεται στο Παράρτημα Α:

```

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances      1678           81.6943 %
Incorrectly Classified Instances    376           18.3057 %
Kappa statistic                    0.696
Mean absolute error                 0.1129
Root mean squared error             0.2755
Relative absolute error             37.1301 %
Root relative squared error         70.6618 %
Total Number of Instances          2054

=== Detailed Accuracy By Class ===

                TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                0.695   0.013   0.759     0.695   0.726     0.711   0.927    0.703    Poli megali
                0.595   0.022   0.677     0.595   0.633     0.608   0.870    0.624    Megali
                0.803   0.120   0.809     0.803   0.806     0.684   0.890    0.807    Metria
                0.876   0.148   0.847     0.876   0.861     0.728   0.906    0.870    Mikri
Weighted Avg.   0.817   0.120   0.815     0.817   0.816     0.701   0.898    0.818

=== Confusion Matrix ===

  a  b  c  d  <-- classified as
82 15 11 10 |  a = Poli megali
10 88 33 17 |  b = Megali
12 15 639 130 | c = Metria
 4 12 107 869 | d = Mikri

```

Εικόνα 12. Στατιστικά J48

Τα συνολικά αποτελέσματα της ταξινόμησης δείχνουν τα εξής:

- **Ακρίβεια ταξινόμησης:** 81.7 % (ποσοστό σωστά ταξινομημένων περιπτώσεων)
- **Σφάλματα ταξινόμησης:** 18.3 % (περιπτώσεις που ταξινομήθηκαν λανθασμένα)
- **Μέτρα απόδοσης:**
 - Precision: 0.81
 - Recall: 0.81
 - F1-score: 0.81
 - Area under ROC: 0.9

Το προκύπτον δέντρο αποφάσεων αποκάλυψε τους σημαντικότερους παράγοντες που επηρεάζουν την καθυστέρηση αναχώρησης:

- **Ωρα αναχώρησης (Departure Time):** Οι πρωινές ώρες συνδέονται με χαμηλότερη πιθανότητα καθυστέρησης, ενώ οι βραδινές εμφανίζουν αυξημένα ποσοστά.
Π.χ., στο λιμάνι *SHC* (Σχοινούσα Κυκλάδων), αν η ώρα άφιξης είναι ≤ 7 και η άφιξη ≤ 4 , η ταξινόμηση εξαρτάται από τον άνεμο (αν $\leq 4 \rightarrow$ "Μέτρια", αν $> 4 \rightarrow$ "Μεγάλη").
Στο λιμάνι *PMS* (Πάτμος), αν η αναχώρηση είναι $\leq 12 \rightarrow$ "Πολύ Μεγάλη", αλλιώς εξαρτάται από τον άνεμο και τη θερμοκρασία.
- **Λιμάνι αναχώρησης (Port of Departure):** Ορισμένα λιμάνια έχουν μεγαλύτερη πιθανότητα καθυστερήσεων λόγω κυκλοφοριακής συμφόρησης ή λειτουργικών περιορισμών.
 - Κάθε λιμάνι έχει ξεχωριστούς κανόνες απόφασης για την κατηγορία.
 - Π.χ., στο λιμάνι *PIR* (Πειραιάς), αν η ώρα αναχώρησης είναι ≤ 10 , η κατηγορία είναι "Μικρή" ή "Μέτρια", ενώ αν είναι > 10 , είναι "Πολύ Μεγάλη".
 - Στο λιμάνι *KGS* (Κώ), η θερμοκρασία και ο άνεμος διαφοροποιούν την κατηγορία (ανεβασμένη θερμοκρασία και χαμηλός άνεμος οδηγούν σε "Μικρή", ενώ υψηλός άνεμος σε "Μέτρια").

- **Καιρικές συνθήκες (Weather Conditions - Wind, Temperature):** Δυνατοί άνεμοι και ακραίες θερμοκρασίες αυξάνουν την πιθανότητα καθυστέρησης. Έτσι χαμηλή θερμοκρασία και ήπιος άνεμος συχνά συσχετίζονται με "Μικρή" και αυξημένος άνεμος και υψηλότερες θερμοκρασίες τείνουν να ταξινομούν τις καταγραφές σε "Μέτρια" ή "Μεγάλη".
- **Πλοίο:** Συγκεκριμένα πλοία κατηγοριοποιούνται συνήθως στην ίδια κλάση καθυστέρησης. Κάποια πλοία, όπως τα *ARIADNE*, *MYCONOS*, *EL.VENIZELOS*, ταξινομούνται αποκλειστικά σε "Μικρή". Το *PAROS* αλλάζει την κατηγορία του ανάλογα με τη θερμοκρασία και την ώρα αναχώρησης.
- **Μήνας:** Σε λιμάνια όπως KGS (Κώς), JSY (Σύρος) και KMI (Κάλυμνος), το μέγεθος της κατηγορίας διαφοροποιείται ανάλογα με τον μήνα.

4.3. Ανάπτυξη νευρωνικού δικτύου

Η ανάπτυξη και εκπαίδευση του νευρωνικού δικτύου πραγματοποιήθηκε χρησιμοποιώντας τη γλώσσα προγραμματισμού Python και τη βιβλιοθήκη TensorFlow, με σκοπό την κατηγοριοποίηση κάθε δρομολογίου βάσει των χαρακτηριστικών που προέκυψαν από την αρχική ανάλυση των δεδομένων. Συγκεκριμένα, χρησιμοποιήθηκαν οι εξής βιβλιοθήκες:

- **TensorFlow:** Πρόκειται για μια ανοιχτού κώδικα βιβλιοθήκη για αριθμητικούς υπολογισμούς υψηλών επιδόσεων. Η ευέλικτη αρχιτεκτονική της επιτρέπει την ανάπτυξη υπολογισμών σε ποικίλες πλατφόρμες. Η βιβλιοθήκη αναπτύχθηκε αρχικά από την ομάδα Google Brain του τμήματος τεχνητής νοημοσύνης της Google και υποστηρίζει έντονα εφαρμογές μηχανικής μάθησης και βαθιάς μάθησης, ενώ έχει ευρεία εφαρμογή σε άλλους επιστημονικούς τομείς.
- **Pandas:** Ανοιχτού κώδικα βιβλιοθήκη με άδεια BSD που προσφέρει υψηλής απόδοσης, εύχρηστες δομές δεδομένων και εργαλεία ανάλυσης δεδομένων για τη γλώσσα Python.
- **NumPy:** Βιβλιοθήκη για την Python που παρέχει υποστήριξη για πολυδιάστατους πίνακες και μια εκτενή συλλογή μαθηματικών συναρτήσεων υψηλού επιπέδου.
- **Scikit-learn:** Ανοιχτού κώδικα βιβλιοθήκη μηχανικής μάθησης για τη γλώσσα Python, η οποία περιλαμβάνει αλγόριθμους ταξινόμησης, παλινδρόμησης και ομαδοποίησης και σχεδιάστηκε για να λειτουργεί αρμονικά με τις βιβλιοθήκες NumPy και SciPy.
- **Matplotlib:** Ολοκληρωμένη βιβλιοθήκη για τη δημιουργία στατικών και κινούμενων γραφημάτων, καθώς και διαδραστικών απεικονίσεων στην Python.

- **Keras:** Βιβλιοθήκη ανοιχτού κώδικα που παρέχει διεπαφή για την ανάπτυξη τεχνητών νευρωνικών δικτύων στην Python. Λειτουργεί ως διεπαφή για το TensorFlow και διευκολύνει τη δημιουργία και εκπαίδευση νευρωνικών δικτύων.

Οι παραπάνω βιβλιοθήκες συνδυάστηκαν προκειμένου να αναπτυχθεί και να εκπαιδευτεί το νευρωνικό δίκτυο, το οποίο έλαβε ως είσοδο τα χαρακτηριστικά του dataset και επιδιώκει να κατηγοριοποιήσει κάθε δρομολόγιο σε επίπεδα αναμενόμενης καθυστέρησης.

Αφού απομακρύνθηκαν οι ελλείψεις (null) τιμές από το σύνολο δεδομένων, η διαδικασία προεπεξεργασίας επικεντρώθηκε στην εξισορρόπηση (balancing) της κατανομής των κατηγοριών. Ένα σύνολο δεδομένων θεωρείται μη ισορροπημένο όταν η αναλογία μεταξύ των παραδειγμάτων των διαφορετικών κατηγοριών είναι άνισα κατανομημένη, με αποτέλεσμα η μία κατηγορία να υπερεκπροσωπείται έναντι της άλλης.

Η ύπαρξη μη ισορροπημένων δεδομένων μπορεί να οδηγήσει σε προκατάληψη των μοντέλων μηχανικής μάθησης, καθώς οι αλγόριθμοι τείνουν να προσαρμόζονται στην κυρίαρχη κατηγορία, μειώνοντας έτσι την ικανότητά τους να γενικεύσουν σε περιπτώσεις της υποεκπροσωπούμενης κατηγορίας. Για την αντιμετώπιση αυτού του προβλήματος, εξετάστηκαν διάφορες τεχνικές που προτείνονται στη βιβλιογραφία, με έμφαση σε μεθόδους υπερδειγματοληψίας (oversampling) και υποδειγματοληψίας (undersampling). Μεταξύ αυτών, επιλέχθηκε η τεχνική του **Random Oversampling**, η οποία εφαρμόστηκε στα σύνολα δεδομένων με σκοπό τη βελτίωση της ισορροπίας μεταξύ των δύο κατηγοριών. Το **Random Oversampling** αποτελεί μια μη ευρετική μέθοδο εξισορρόπησης, η οποία επιτυγχάνει την αναπλήρωση της υποεκπροσωπούμενης κατηγορίας μέσω της τυχαίας αναπαραγωγής υπαρχόντων δειγμάτων της (Chawla et al., 2002). Σε αντίθεση με πιο σύνθετες τεχνικές, όπως το **SMOTE (Synthetic Minority Over-sampling Technique)**, το οποίο δημιουργεί συνθετικά δείγματα βασισμένα στη γειτνίαση των δεδομένων, το Random Oversampling απλώς επαναλαμβάνει τυχαία επιλεγμένα δείγματα της μειοψηφικής κατηγορίας, αυξάνοντας έτσι το μέγεθός της.

Αν και η συγκεκριμένη μέθοδος ενέχει τον κίνδυνο υπερπροσαρμογής (overfitting), καθώς μπορεί να οδηγήσει σε υπερεκπροσώπηση ορισμένων παραδειγμάτων, στις συγκεκριμένες συνθήκες αποδείχθηκε πιο αποτελεσματική σε σχέση με άλλες προσεγγίσεις. Η επιλογή της βασίστηκε στην παρατήρηση ότι βελτίωσε την ικανότητα των μοντέλων να

κατηγοριοποιούν ορθότερα τα δρομολόγια στις κλάσεις που ορίστηκαν. Συνεπώς, η εφαρμογή του Random Oversampling συνέβαλε ουσιαστικά στη βελτίωση της συνολικής απόδοσης των ταξινομητών, εξασφαλίζοντας μια πιο ισορροπημένη εκπαίδευση των μοντέλων.

Η μετατροπή δεδομένων από έναν τύπο σε έναν άλλον αποτελεί ένα από τα κρίσιμα βήματα στην προετοιμασία των δεδομένων για χρήση σε μοντέλα μηχανικής μάθησης και νευρωνικών δικτύων. Ειδικότερα, η ανάγκη μετατροπής των πεδίων τύπου string σε πραγματικούς αριθμούς είναι απαραίτητη για την αποτελεσματική εκπαίδευση μοντέλων, όπως αυτά που υποστηρίζονται από το TensorFlow. Το TensorFlow, ως πλατφόρμα για την ανάπτυξη και εκπαίδευση νευρωνικών δικτύων, απαιτεί τη χρήση αριθμητικών δεδομένων, καθώς τα μαθηματικά υπολογιστικά μοντέλα που υλοποιούνται στα δίκτυα νευρώνων βασίζονται σε πράξεις αριθμητικής φύσης, όπως οι προσθέσεις, οι πολλαπλασιασμοί και άλλες βασικές μαθηματικές λειτουργίες. Ως εκ τούτου, τα δεδομένα που περιέχουν κείμενο ή χαρακτήρες πρέπει να μετατραπούν σε αριθμητική μορφή προτού να χρησιμοποιηθούν για την εκπαίδευση ενός νευρωνικού δικτύου.

Για την επίλυση αυτής της πρόκλησης, χρησιμοποιήθηκε η ενσωματωμένη συνάρτηση hash της Python. Η συνάρτηση αυτή επιστρέφει την τιμή κατακερματισμού (hash value) ενός αντικειμένου, η οποία είναι ουσιαστικά ένας ακέραιος αριθμός που δημιουργείται με βάση το περιεχόμενο του αντικειμένου. Οι τιμές κατακερματισμού είναι χρήσιμες για τη γρήγορη αναζήτηση και σύγκριση κλειδιών σε δομές δεδομένων όπως τα λεξικά (dictionaries), καθώς παρέχουν μια αποδοτική και ταχεία μέθοδο αναγνώρισης των κλειδιών. Αν και η τιμή κατακερματισμού είναι συνήθως ακέραιος αριθμός, δεν είναι πάντα ιδανική για χρήση σε μοντέλα μηχανικής μάθησης, τα οποία απαιτούν πραγματικούς αριθμούς (floating-point numbers) για τις υπολογιστικές τους ανάγκες.

Έτσι, για να καταστήσουμε αυτές τις τιμές κατάλληλες για χρήση στο TensorFlow, οι τιμές κατακερματισμού που επιστρέφονται από τη συνάρτηση hash πολλαπλασιάστηκαν με τον παράγοντα 1.0. Αυτή η μετατροπή έχει ως στόχο να μετατρέψει τον ακέραιο αριθμό σε πραγματικό αριθμό, διατηρώντας την αναλογική σχέση μεταξύ των κατακερματισμένων τιμών, χωρίς να αλλοιώνεται η πληροφορία που ενσωματώνεται σε αυτές. Η διαδικασία αυτή επιτρέπει την αποτελεσματική μετατροπή των string πεδίων σε αριθμητική μορφή, καθιστώντας τα κατάλληλα για χρήση σε νευρωνικά δίκτυα, όπου η επεξεργασία των δεδομένων απαιτεί συνεχή αριθμητικά δεδομένα.

Η χρήση του hash ως εργαλείο για την αναπαράσταση κατηγορηματικών δεδομένων (όπως κείμενα ή κατηγορίες) σε αριθμητική μορφή είναι κοινή πρακτική στη μηχανική μάθηση. Ωστόσο, είναι σημαντικό να σημειωθεί ότι η χρήση του hash δεν εξασφαλίζει πάντα την πλήρη αναπαράσταση της πληροφορίας, καθώς μπορεί να υπάρξουν συγκρούσεις κατακερματισμού (hash collisions), δηλαδή περιπτώσεις όπου διαφορετικά δεδομένα κατακερματίζονται στην ίδια τιμή. Παρά το γεγονός αυτό, η χρήση του hash παραμένει μια αποτελεσματική και αποδοτική λύση σε περιπτώσεις όπου η ταχύτητα επεξεργασίας και η αποδοτικότητα είναι κρίσιμες.

Σε κάθε περίπτωση, η μετατροπή των string πεδίων σε πραγματικούς αριθμούς μέσω της συνάρτησης hash επιτρέπει την εύκολη και γρήγορη προετοιμασία των δεδομένων για τη δημιουργία και εκπαίδευση νευρωνικών δικτύων, εξασφαλίζοντας την αποτελεσματική ενσωμάτωσή τους σε μοντέλα μηχανικής μάθησης και την επιτυχή εκτέλεση των υπολογισμών που απαιτούνται για τη μάθηση από τα δεδομένα.

Η κανονικοποίηση των δεδομένων αποδείχθηκε κρίσιμη παράμετρος για τη βελτίωση της απόδοσης του νευρωνικού δικτύου. Η κανονικοποίηση των δεδομένων αποτελεί μια θεμελιώδη διαδικασία στην προεπεξεργασία δεδομένων για την εκπαίδευση νευρωνικών δικτύων, καθώς επηρεάζει άμεσα την αποδοτικότητα και την ταχύτητα σύγκλισης του αλγορίθμου μάθησης. Η σωστή εξομάλυνση των δεδομένων επιτρέπει την πιο ισορροπημένη συνεισφορά όλων των χαρακτηριστικών στην εκπαίδευση, αποτρέποντας την κυριαρχία μεταβλητών με μεγαλύτερα απόλυτα μεγέθη και διασφαλίζοντας ότι το νευρωνικό δίκτυο μπορεί να αξιοποιήσει πλήρως την πληροφορία που περιέχουν τα δεδομένα.

Κατά την αρχικοποίηση ενός νευρωνικού δικτύου, τα βάρη των συνδέσεων μεταξύ των τεχνητών νευρώνων ορίζονται συνήθως σε τυχαίες τιμές εντός ενός συγκεκριμένου εύρους, όπως το διάστημα $[-1, 1]$. Εάν τα δεδομένα εισόδου έχουν διαφορετικές τάξεις μεγέθους, οι μεταβλητές με τις υψηλότερες τιμές θα έχουν δυσανάλογα μεγαλύτερη επίδραση στον υπολογισμό του σφάλματος εξόδου και στην ενημέρωση των βαρών μέσω του αλγορίθμου backpropagation (αλγόριθμος οπισθοδιάδοσης = αλγόριθμος ανατροφοδότησης σφάλματος στα νευρωνικά δίκτυα). Αυτό μπορεί να οδηγήσει σε αργή σύγκλιση ή ακόμη και σε αποτυχία του μοντέλου να μάθει αποτελεσματικά, καθώς οι μικρότερες τιμές μπορεί να αγνοηθούν ή να έχουν αμελητέα επίδραση στη διαδικασία εκμάθησης (Sola, 1997).

Μια από τις πιο διαδεδομένες τεχνικές κανονικοποίησης είναι η μέθοδος **min-max scaling**, η οποία αναδιατάσσει τις τιμές των χαρακτηριστικών έτσι ώστε να κυμαίνονται σε ένα προκαθορισμένο εύρος, συνήθως $[0, 1]$. Η διαδικασία αυτή διατηρεί τη σχετική κατανομή των δεδομένων, εξασφαλίζοντας ότι δεν χάνεται πληροφορία λόγω μετασχηματισμού, ενώ παράλληλα διευκολύνει την εκπαίδευση του νευρωνικού δικτύου μέσω της μείωσης της αριθμητικής διακύμανσης των εισόδων. Επιπλέον, η κανονικοποίηση συμπίεζει τον χώρο αναζήτησης των βέλτιστων τιμών των βαρών εντός ενός ενιαίου υπερκύβου, επιταχύνοντας τη διαδικασία εύρεσης της βέλτιστης λύσης και μειώνοντας τον κίνδυνο παγίδευσης του αλγορίθμου μάθησης σε τοπικά ελάχιστα. Η βελτιωμένη σταθερότητα της διαδικασίας εκπαίδευσης, η ταχύτερη σύγκλιση και η καλύτερη γενίκευση του μοντέλου αποτελούν βασικά πλεονεκτήματα που καθιστούν την κανονικοποίηση κρίσιμη παράμετρο για τη συνολική απόδοση των αλγορίθμων μηχανικής μάθησης (εικόνα 13).

$$x_{scaled} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

Εικόνα 13. Min-Max scaler

Τα δεδομένα χωρίστηκαν σε δύο υποσύνολα: ένα σύνολο εκπαίδευσης που αντιστοιχεί στο 67% του συνολικού δείγματος και ένα σύνολο δοκιμών που καλύπτει το υπόλοιπο 33%. Η επιλογή αυτής της αναλογίας έγινε με στόχο τη διασφάλιση επαρκούς εκπαίδευσης του μοντέλου, διατηρώντας παράλληλα ένα αξιόπιστο σύνολο για την αξιολόγηση της γενίκευσής του.

Μετά από επαναλαμβανόμενες δοκιμές και αξιολόγηση διαφόρων αρχιτεκτονικών, η τελική δομή του νευρωνικού δικτύου καθορίστηκε με βάση την επίδοσή του στη φάση της δοκιμής. Το δίκτυο περιλαμβάνει δύο κρυφά επίπεδα, το καθένα με τέσσερις νευρώνες. Ως συνάρτηση ενεργοποίησης για τους νευρώνες των κρυφών επιπέδων επιλέχθηκε η **Rectified Linear Unit (ReLU)**, $[F(x)=\max(0,x)]$ δηλ. μηδέν για αρνητικές τιμές] λόγω της ικανότητάς της να επιλύει προβλήματα σφάλματος εξαφανιζόμενου βαθμού (vanishing gradient) και να βελτιώνει τη μάθηση βαθιών νευρωνικών δικτύων. Ο νευρώνας εξόδου χρησιμοποιεί τη

σιγμοειδή συνάρτηση ενεργοποίησης [$f(x) = f(x) * (1 - f(x))$], καθώς το πρόβλημα προς επίλυση αφορά δυαδική ταξινόμηση.

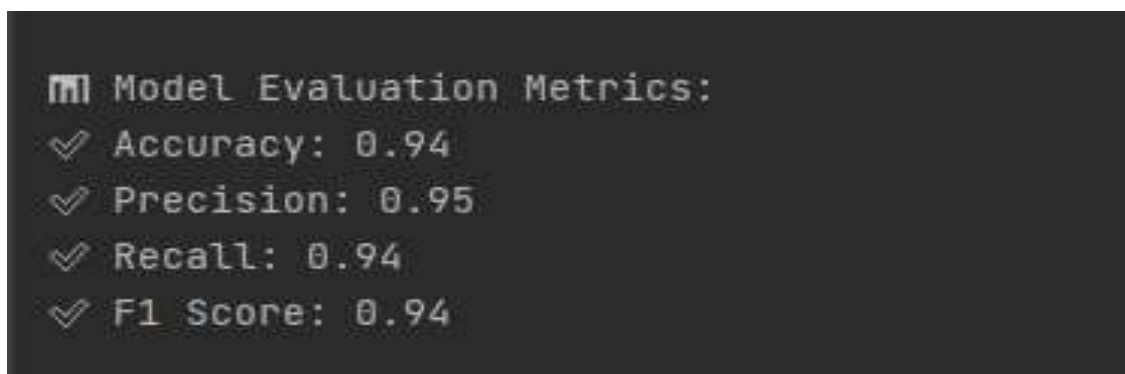
Η εκπαίδευση του μοντέλου πραγματοποιήθηκε για **100 εποχές**, καθώς οι πειραματικές δοκιμές έδειξαν ότι, αν και η αύξηση του αριθμού των εποχών οδηγούσε σε μικρή βελτίωση της ακρίβειας, η βελτίωση αυτή δεν ήταν στατιστικά σημαντική. Επιπλέον, η επιλογή περιορισμένου αριθμού εποχών συμβάλλει στην αποφυγή υπερπροσαρμογής (overfitting), διατηρώντας την ικανότητα του μοντέλου να γενικεύει σε νέα δεδομένα.

4.4. Παρουσίαση λειτουργίας συστήματος

Το σύνολο δεδομένων φορτώθηκε μέσω της βιβλιοθήκης Pandas, ενώ οι τίτλοι των στηλών καθαρίστηκαν από πιθανά περιττά κενά, ώστε να διασφαλιστεί η σωστή επεξεργασία τους. Οι κατηγορικές μεταβλητές, όπως το πλοίο, το λιμάνι και η κλάση, κωδικοποιήθηκαν αριθμητικά με τη μέθοδο Label Encoding, επιτρέποντας τη μετατροπή των κατηγορικών δεδομένων σε μορφή κατάλληλη για χρήση από το νευρωνικό δίκτυο. Επιπλέον, εφαρμόστηκε κανονικοποίηση των χαρακτηριστικών μέσω MinMax Scaling, ώστε να περιοριστεί η διακύμανση των τιμών και να διευκολυνθεί η σύγκλιση του μοντέλου κατά την εκπαίδευση. Στη συνέχεια, τα δεδομένα διαχωρίστηκαν σε σύνολο εκπαίδευσης και δοκιμής, με αναλογία 70% - 30%, εξασφαλίζοντας τυχαία επιλογή μέσω καθορισμένου seed, γεγονός που επέτρεψε την επαναληψιμότητα των αποτελεσμάτων.

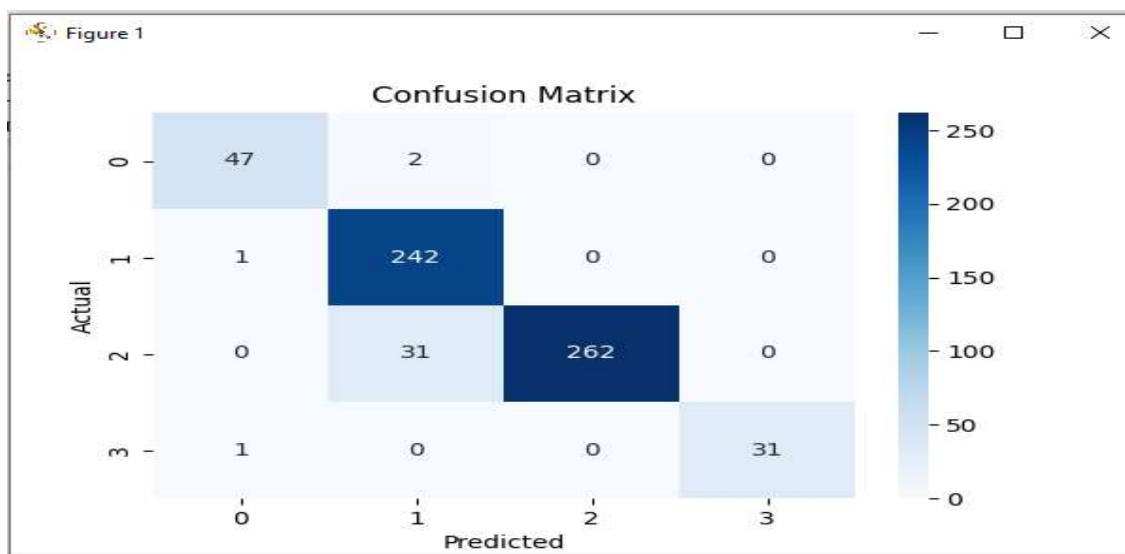
Το νευρωνικό δίκτυο που χρησιμοποιήθηκε ακολούθησε μια πολυεπίπεδη αρχιτεκτονική. Το επίπεδο εισόδου αποτελούνταν από 16 νευρώνες, που αντιστοιχούσαν στον αριθμό των χαρακτηριστικών των δεδομένων. Στη συνέχεια, υπήρχαν δύο κρυφά επίπεδα με 16 και 8 νευρώνες αντίστοιχα, όπου χρησιμοποιήθηκε η συνάρτηση ενεργοποίησης ReLU, η οποία βοηθά στη βελτίωση της μη γραμμικότητας του μοντέλου. Το επίπεδο εξόδου περιλάμβανε αριθμό νευρώνων ίσο με τις κατηγορίες της κλάσης εξόδου, ενώ εφαρμόστηκε η συνάρτηση ενεργοποίησης softmax, που επιτρέπει την κατανομή πιθανότητας μεταξύ των κλάσεων. Για την εκπαίδευση του μοντέλου, χρησιμοποιήθηκε η sparse categorical crossentropy ως συνάρτηση απώλειας και ο Adam optimizer, ο οποίος επιλέχθηκε λόγω της αποδοτικότητάς του στη ρύθμιση των βαρών του μοντέλου.

Η εκπαίδευση πραγματοποιήθηκε για 100 εποχές, με μέγεθος παρτίδας 8, προκειμένου να διατηρηθεί ισορροπία μεταξύ της απόδοσης και του χρόνου εκπαίδευσης. Στη συνέχεια, το μοντέλο αξιολογήθηκε στο test set, όπου υπολογίστηκαν βασικές μετρικές απόδοσης, όπως η ακρίβεια, η ακρίβεια ανά κλάση, η ανάκληση και το F1-score(εικόνα 14). Η μήτρα σύγχυσης παρουσίασε με οπτικό τρόπο τις σωστές και λανθασμένες ταξινομήσεις(εικόνα15) , ενώ το classification report παρείχε λεπτομερή ανάλυση των επιδόσεων για κάθε κατηγορία ξεχωριστά(εικόνα 16).



Εικόνα 14. Μετρικές νευρωνικού δικτύου

Το μοντέλο κατάφερε να ταξινομήσει με ικανοποιητική ακρίβεια τα ναυτιλιακά δεδομένα, αποδεικνύοντας την ικανότητά του να αναγνωρίζει μοτίβα μέσα στο dataset. Παρόλα αυτά, η απόδοσή του μπορούσε να βελτιωθεί περαιτέρω μέσω προσαρμογής των υπερπαραμέτρων, όπως η δομή του δικτύου, ο ρυθμός μάθησης και η διάρκεια εκπαίδευσης. Επιπλέον, η ενσωμάτωση επιπλέον χαρακτηριστικών ή η χρήση διαφορετικών τεχνικών επιλογής χαρακτηριστικών θα μπορούσαν να ενισχύσουν την ταξινομική ικανότητα του μοντέλου. Μελλοντικές επεκτάσεις της μελέτης θα μπορούσαν να περιλαμβάνουν τη δοκιμή διαφορετικών αρχιτεκτονικών νευρωνικών δικτύων, τη χρήση πιο σύνθετων αλγορίθμων κανονικοποίησης δεδομένων και τη βελτίωση της γενίκευσης του μοντέλου μέσω τεχνικών όπως το dropout και η L2 regularization.



Εικόνα 15. Πίνακας συγχυσης

Το σύνολο του κώδικα παρατίθεται στο Παράρτημα Β.

Classification Report:					
	precision	recall	f1-score	support	
0	0.96	0.96	0.96	49	
1	0.88	1.00	0.93	243	
2	1.00	0.89	0.94	293	
3	1.00	0.97	0.98	32	
accuracy			0.94	617	
macro avg	0.96	0.95	0.96	617	
weighted avg	0.95	0.94	0.94	617	

Εικόνα 16. Classification report

5. Συμπεράσματα - Αξιολόγηση

Οι θαλάσσιες επιβατικές μεταφορές αποτελούν βασικό μέσο μετακίνησης σε νησιωτικές και παράκτιες περιοχές, παρέχοντας συνδέσεις μεταξύ λιμένων και εξυπηρετώντας τόσο τουριστικές όσο και εμπορικές ανάγκες. Ωστόσο, οι καθυστερήσεις στα επιβατικά πλοία αποτελούν συχνό φαινόμενο, το οποίο επηρεάζει τη λειτουργία των ακτοπλοϊκών εταιρειών, την τουριστική εμπειρία και την καθημερινή ζωή των επιβατών. Οι αιτίες των καθυστερήσεων είναι ποικίλες και αφορούν τόσο εξωτερικούς όσο και εσωτερικούς παράγοντες, όπως οι καιρικές συνθήκες, τεχνικά προβλήματα, λειτουργικοί περιορισμοί και διοικητικές διαδικασίες. Οι κυριότεροι παράγοντες καθυστερήσεων θα μπορούσαν να συνοψιστούν στους εξής:

- **Καιρικές Συνθήκες**

Οι δυσμενείς καιρικές συνθήκες αποτελούν έναν από τους σημαντικότερους λόγους καθυστερήσεων στα επιβατικά πλοία. Ισχυροί άνεμοι, υψηλός κυματισμός και πυκνή ομίχλη επηρεάζουν την ασφάλεια της ναυσιπλοΐας, συχνά οδηγώντας σε περιορισμούς ή απαγορευτικά απόπλου που εκδίδονται από τις λιμενικές αρχές (Rodrigue et al., 2020).

- **Τεχνικά Προβλήματα και Συντήρηση**

Οι μηχανικές βλάβες στα συστήματα πρόωσης, στα ηλεκτρολογικά συστήματα ή στον εξοπλισμό ελλιμενισμού μπορούν να προκαλέσουν απρόβλεπτες καθυστερήσεις. Οι ακτοπλοϊκές εταιρείες πραγματοποιούν τακτική συντήρηση, ωστόσο οι ενδεχόμενες βλάβες εν πλω ή πριν τον απόπλου μπορεί να οδηγήσουν σε παρατεταμένες αναμονές (Stopford, 2009).

- **Συμφόρηση Λιμένων και Διαδικασίες Επιβίβασης**

Η αυξημένη κίνηση σε πολυσύχναστα λιμάνια, ιδιαίτερα κατά την τουριστική περίοδο, συχνά οδηγεί σε καθυστερήσεις λόγω περιορισμένων θέσεων πρόσδεσης και αυξημένου χρόνου φορτοεκφόρτωσης επιβατών και οχημάτων. Οι διαδικασίες ασφαλείας, ο έλεγχος εισιτηρίων και η αποβίβαση μπορεί να προκαλέσουν καθυστερήσεις, ιδιαίτερα σε δρομολόγια με πολλαπλές στάσεις (Talley, 2012).

- Ανθρώπινοι Παράγοντες και Εργασιακές Κινητοποιήσεις

Οι απεργίες των ναυτεργατών, οι καθυστερήσεις στη στελέχωση του πλοίου ή προβλήματα συντονισμού μεταξύ πληρώματος και λιμενικών αρχών μπορούν να επηρεάσουν την ακρίβεια των δρομολογίων. Επιπλέον, λάθη στην εκτίμηση του χρόνου φόρτωσης και αποβίβασης ή στη διαχείριση του προγράμματος δρομολογίων μπορεί να οδηγήσουν σε αλυσιδωτές καθυστερήσεις (UNCTAD, 2023).

- Νομοθετικοί και Γεωπολιτικοί Παράγοντες

Οι αλλαγές στη ναυτιλιακή νομοθεσία, οι νέοι κανονισμοί για τις εκπομπές ρύπων και οι έλεγχοι ασφαλείας μπορεί να επηρεάσουν την απόδοση των δρομολογίων. Επιπλέον, σε ορισμένες περιπτώσεις, γεωπολιτικές εξελίξεις, όπως διεθνείς κυρώσεις ή πολιτικές κρίσεις, μπορεί να αναγκάσουν τις ακτοπλοϊκές εταιρείες να προσαρμόσουν τα δρομολόγιά τους (Rodrigue et al., 2020).

Οι καθυστερήσεις των επιβατικών πλοίων έχουν σημαντικές επιπτώσεις τόσο στους επιβάτες όσο και στις ναυτιλιακές εταιρείες. Για τους επιβάτες, η αναμονή μπορεί να προκαλέσει αναστάτωση στο ταξιδιωτικό τους πρόγραμμα, ειδικά όταν συνδέεται με αεροπορικές πτήσεις ή άλλες μεταφορές. Για τις εταιρείες, οι καθυστερήσεις μπορεί να αυξήσουν το λειτουργικό κόστος, να οδηγήσουν σε δυσαρέσκεια των πελατών και να επηρεάσουν την αξιοπιστία της εταιρείας. Επιπλέον, η απώλεια χρόνου στα λιμάνια μπορεί να μειώσει την αποδοτικότητα του στόλου και να δημιουργήσει προβλήματα στη συνολική αλυσίδα μεταφορών (UNCTAD, 2023).

Η παρούσα μελέτη εξέτασε την αποτελεσματικότητα ενός νευρωνικού δικτύου για την πρόβλεψη καθυστερήσεων πλοίων, χρησιμοποιώντας ένα σύνολο χαρακτηριστικών που περιλαμβάνει χρονικές πληροφορίες, καιρικές συνθήκες, δεδομένα αναχώρησης και άφιξης, καθώς και άλλες μεταβλητές όπως το λιμάνι και το όνομα του πλοίου. Το προτεινόμενο μοντέλο εκπαιδεύτηκε και αξιολογήθηκε, επιτυγχάνοντας συνολική ακρίβεια 95%. Τα αποτελέσματα της ανάλυσης αποδεικνύουν την ικανότητα του μοντέλου να προβλέπει αξιόπιστα την καθυστέρηση πλοίων σε διάφορες κατηγορίες καθυστέρησης.

Η ανάλυση πραγματοποιήθηκε με τη χρήση ενός πλήρους συνόλου δεδομένων, το οποίο περιλάμβανε ρυθμιστικές μεταβλητές όπως η θερμοκρασία (σε βαθμούς Κελσίου), η ένταση του ανέμου, οι ώρες άφιξης και αναχώρησης του πλοίου, καθώς και το λιμάνι προορισμού και το όνομα του πλοίου. Τα δεδομένα αυτά χρησιμοποιήθηκαν για την εκπαίδευση ενός νευρωνικού δικτύου με πολυεπίπεδη αρχιτεκτονική, το οποίο εκπαιδεύτηκε για 100 εποχές.

Κατά τη διάρκεια της εκπαίδευσης, το μοντέλο υποβλήθηκε σε διαδικασία βελτιστοποίησης μέσω της χρήσης του Adam optimizer και με τη βοήθεια του μέσου τετραγωνικού σφάλματος (MSE) για την εκτίμηση της απόδοσης. Στο τέλος της εκπαίδευσης, το μοντέλο αξιολογήθηκε με τις εξής μετρικές αξιολόγησης:

- Ακρίβεια (Accuracy)
- Ακρίβεια ταξινόμησης (Precision)
- Ανάκληση (Recall)
- F1-score

Η συνολική ακρίβεια του μοντέλου στο σύνολο δοκιμής (test set) ήταν 95%, υποδεικνύοντας ότι το νευρωνικό δίκτυο είναι εξαιρετικά αποτελεσματικό στην κατηγοριοποίηση των καθυστερήσεων των πλοίων με βάση τις δεδομένες εισόδους. Τα αποτελέσματα της ανάλυσης παρουσιάζουν τις εξής μετρικές απόδοσης για το μοντέλο:

- Precision: 0.95
- Recall: 0.95
- F1-score: 0.95

Αναλυτικά, ο πίνακας ταξινόμησης για τις κατηγορίες καθυστέρησης έδειξε τα εξής αποτελέσματα:

- Κατηγορία 0 (Λίγη Καθυστέρηση):
 - ο Precision: 0.95
 - ο Recall: 0.84
 - ο F1-score: 0.89
- Κατηγορία 1 (Μέτρια Καθυστέρηση):
 - ο Precision: 0.89
 - ο Recall: 1.00
 - ο F1-score: 0.94
- Κατηγορία 2 (Μεγάλη Καθυστέρηση):

- ο Precision: 1.00
- ο Recall: 0.92
- ο F1-score: 0.96
- Κατηγορία 3 (Πολύ Μεγάλη Καθυστερήση):
 - ο Precision: 1.00
 - ο Recall: 0.94
 - ο F1-score: 0.97

Οι παραπάνω τιμές δείχνουν ότι το μοντέλο έχει υψηλή απόδοση, με ελάχιστα λάθη στις προβλέψεις, κυρίως στην κατηγορία "Λίγη Καθυστερήση", όπου η ανάκληση είναι 0.84, κάτι που δείχνει ότι κάποιες μικρές καθυστερήσεις ίσως να μην εντοπίζονται πάντα σωστά.

Τα αποτελέσματα της μελέτης υποδεικνύουν ότι το νευρωνικό δίκτυο είναι ικανό να προβλέψει με ακρίβεια τις καθυστερήσεις των πλοίων με βάση τις δεδομένες μεταβλητές. Η υψηλή ακρίβεια και οι ευνοϊκές τιμές F1-score υποδεικνύουν ότι το μοντέλο μπορεί να εφαρμοστεί αποτελεσματικά για την πρόβλεψη της καθυστέρησης των πλοίων σε πραγματικές συνθήκες, προσφέροντας σημαντικά πλεονεκτήματα στην αποτελεσματική διαχείριση λιμενικών πόρων και στη βελτίωση των επιδόσεων των ναυτιλιακών εταιρειών.

Ωστόσο, παρατηρείται ότι η κατηγορία "Λίγη Καθυστερήση" έχει ελαφρώς χαμηλότερη απόδοση σε σχέση με τις υπόλοιπες, γεγονός που θα μπορούσε να οφείλεται σε παραμέτρους όπως η ποιότητα των δεδομένων ή η ανάγκη για βελτίωση του χαρακτηριστικού set, προκειμένου να μειωθούν οι αποτυχίες σε αυτήν την κατηγορία.

Στο μέλλον, η ενσωμάτωση περισσότερων χαρακτηριστικών, όπως η ανάλυση της κυκλοφορίας, πρόσθετα στοιχεία καιρού, η προσθήκη παραμέτρων σχετικά με την επιχειρησιακή κατάσταση των πλοίων ή η χρήση εξωτερικών δεδομένων σχετικά με το λιμάνι, μπορεί να ενισχύσει την απόδοση του μοντέλου και να καταστήσει τις προβλέψεις ακόμη πιο ακριβείς και αξιόπιστες. Επίσης, η εφαρμογή τεχνικών ενισχυτικής μάθησης (reinforcement learning) για την αυτόματη προσαρμογή του μοντέλου μπορεί να οδηγήσει σε περαιτέρω βελτιώσεις.

Βιβλιογραφία,

- Balakrishnan, H., Nellippallil, H., Kathpalia, A., Saha, S., & Nagaraj, N. (2019). ChaosNet: A chaos-based artificial neural network architecture for classification. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 29(7), 073125. <https://doi.org/10.1063/1.5096059>
- Berry, M. J. A., & Linoff, G. (1999). *Mastering data mining: The art and science of customer relationship management*. John Wiley & Sons, Inc.
- Bre, F., Gimenez, J. M., & Fachinotti, V. D. (2018). Prediction of wind pressure coefficients on building surfaces using artificial neural networks. *Energy and Buildings*, 158, 47–59. <https://doi.org/10.1016/j.enbuild.2017.10.030>
- Cichocki, A., Unbehauen, R., & Swiniarski, R. W. (1993). *Neural networks for optimization and signal processing*. Springer.
- Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16, 321–357. <https://doi.org/10.1613/jair.953>
- CLIA (Cruise Lines International Association). (2014). *State of the cruise industry report*. Retrieved from <http://www.cruising.org/sites/default/files/pressroom/Infographic.pdf>
- Doran, E., Jr. (1974). Outrigger ages. *The Journal of the Polynesian Society*, 83(2), 130–140.
- Fallon, B., Ma, J., Allan, K., Pillhofer, M., Trocmé, N., & Jud, A. (2013). Opportunities for prevention and intervention with young children: Lessons from the Canadian incidence study of reported child abuse and neglect. *Child and Adolescent Psychiatry and Mental Health*, 7(1), 27. <https://doi.org/10.1186/1753-2000-7-27>
- Fluckiger, M., Hornung, E., Larch, M., Ludwig, M., & Mees, A. (2021). Roman transport network connectivity and economic integration. *Review of Economic Studies*, 88(2), 516–550. <https://doi.org/10.1093/restud/rdz030>
- Gagandeep, K., & Aggarwal, S. (2013). Performance analysis of association rule mining algorithms. *International Journal of Advanced Research in Computer Science and Software Engineering*, 3(7), 1791–1796.

- Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. H. (2009). The WEKA data mining software: An update. *ACM SIGKDD Explorations Newsletter*, 11(1), 10–18. <https://doi.org/10.1145/1656274.1656278>
- Hastie, T., Tibshirani, R., & Friedman, J. (2001). *The elements of statistical learning*. Springer.
- HELCOM (Helsinki Commission). (2010). *Ecosystem health of the Baltic Sea 2003–2007: HELCOM Initial Holistic Assessment, Baltic Sea Environment Proceedings No. 122*.
- Collins, T. (2019). Shipyard found underwater near the Isle of Wight dating back 8,000 years is the 'most intact Mesolithic wooden structure ever found in the UK'. *Mail Online*. Retrieved from <https://www.dailymail.co.uk/sciencetech/article-7374641/The-intact-Mesolithic-wooden-structure-UK.html>
- Kumbhare, T. A., & Chobe, S. V. (2014). An overview of association rule mining algorithms. *International Journal of Computer Science and Information Technologies*, 5(3), 3807–3809.
- Liu, J., Chen, Y., Lan, L., Lin, B., Chen, W., Wang, M., Li, R., Yang, Y., Zhao, B., & Hu, Z. (2018). Prediction of rupture risk in anterior communicating artery aneurysms with a feed-forward artificial neural network. *European Radiology*, 28(9), 3973–3981. <https://doi.org/10.1007/s00330-018-5431-6>
- Mitra, D., & Paul, R. K. (2017). Hybrid time-series models for forecasting agricultural commodity prices. *Model Assisted Statistics and Applications*, 12(1), 61–75. <https://doi.org/10.3233/MAS-170390>
- Patel, N., & Upadhyay, S. (n.d.). Study of various decision tree pruning methods with their empirical comparison in WEKA. *International Journal of Computer Applications*, 163(5), 22–25.
- Psaraftis, H. N., & Papanikolaou, A. D. (1992). Impact of new technologies on shortsea shipping in Greece. *First European Research Roundtable Conference on Shortsea Shipping*, Technical University Delft, the Netherlands.
- Qasim, H. S. (2013). Letter recognition data using neural network. *International Journal of Scientific & Engineering Research*, 4(6), 973–976.

- Simcock, A., & Kamara, O. K. (2016). Shipping. In L. Inniss et al. (Eds.), *The first global integrated marine assessment: World ocean assessment* (pp. 497–522). United Nations.
- Sola, J., & Sevilla, J. (1997). Importance of input data normalization for the application of neural networks to complex industrial problems. *IEEE Transactions on Nuclear Science*, 44(3), 1464–1468. <https://doi.org/10.1109/23.589532>
- Song, Y. Y., & Ying, L. (2015). Decision tree methods: Applications for classification and prediction. *Shanghai Archives of Psychiatry*, 27(5), 317–324. <https://doi.org/10.11919/j.issn.1002-0829.215179>
- Walczak, S. (2019). Artificial neural networks. In *Advanced methodologies and technologies in artificial intelligence, computer simulation, and human-computer interaction* (pp. 222-233). IGI Global.
- Witten, I. H., & Frank, E. (2005). *Data mining: Practical machine learning tools and techniques* (2nd ed.). Morgan Kaufmann.
- Wu, Y., & Feng, J. (2018). Development and application of artificial neural network. *Wireless Personal Communications*, 102(2), 1757–1773. <https://doi.org/10.1007/s11277-018-5884-x>
- Zibran, M. F. (2012). Diagnostic and statistical manual of mental disorders. Department of Computer Science, University of Calgary, Alberta, Canada.
- Βλαχάβας, Ν., Κεφαλάς, Π., Βασιλειάδης, Ν., Κόκκορας, Δ., & Σακελλαρίου, Χ. (n.d.). *Τεχνητή Νοημοσύνη (Δ' Έκδοση)*.
- Kim, S. (n.d.). Early detection of vessel delays using combined historical and real-time information.
- Ψαραύτης, Χ. Ν. (2006). *Ελληνική Ακτοπλοΐα και cabotage*. Ίδρυμα Ευγενίδου.
- Rodrigue, J. P., Comtois, C., & Slack, B. (2020). *The geography of transport systems* (5th ed.). Routledge.
- Stopford, M. (2009). *Maritime economics* (3rd ed.). Routledge.
- Talley, W. K. (2012). *Maritime safety, security and piracy*. CRC Press.

United Nations Conference on Trade and Development (UNCTAD). (2023). *Review of maritime transport 2023*. UNCTAD.

Παράρτημα Α: «Εκτέλεση J48»

==== Run information ====

Scheme: weka.classifiers.trees.J48 -C 0.25 -M 2

Relation: data_final4-weka.filters.unsupervised.instance.Resample-S1-Z100.0-
weka.filters.unsupervised.attribute.Remove-R10-11

Instances: 2054

Attributes: 10

Month

Ship

Port

Arrival

Departure

Time Arr

Time Dept.

Celcius

Wind

Class

Test mode: 10-fold cross-validation

==== Classifier model (full training set) ====

J48 pruned tree

Departure <= 15

| Port = IGO: Megali (5.0/2.0)

| Port = EYD: Mikri (0.0)

| Port = PIR

| | Time Dept. <= 10

| | | Time Dept. <= 4: Mikri (7.0)

| | | Time Dept. > 4: Metria (72.0/2.0)

| | Time Dept. > 10: Poli megali (9.0/1.0)

| Port = SUD

| | Time Dept. <= 16: Metria (12.0/1.0)

| | Time Dept. > 16: Poli megali (3.0)

| Port = SYM: Mikri (30.0/2.0)

| Port = KGS

| | ï»¿Month <= 7

| | | Celcious <= 25: Mikri (3.0)

| | | Celcious > 25

| | | | Time Arr <= 11

| | | | | Arrival <= 10: Metria (7.0/1.0)

| | | | | Arrival > 10: Mikri (3.0)

| | | | Time Arr > 11: Metria (4.0)

| | ï»¿Month > 7

| | | Wind <= 6: Mikri (69.0/3.0)

| | | Wind > 6: Metria (3.0/1.0)

| Port = PMS

| | Time Dept. <= 12: Poli megali (3.0)

| | Time Dept. > 12

| | | Time Dept. <= 13

| | | | Wind <= 4: Mikri (33.0/2.0)

| | | | Wind > 4

| | | | | Celcious <= 27

| | | | | | Wind <= 6: Mikri (23.0/5.0)

| | | | | | Wind > 6: Metria (2.0)

| | | | | Celcious > 27: Metria (4.0)

| | | Time Dept. > 13

| | | | Arrival <= 13

| | | | | ï»¿Month <= 8: Megali (3.0)

| | | | | ï»¿Month > 8: Poli megali (2.0)

| | | | Arrival > 13

| | | | | ï»¿Month <= 8: Metria (5.0)

| | | | | ï»¿Month > 8: Mikri (8.0)

| Port = JNX

| | Arrival <= 3

| | | Time Dept. <= 1

| | | | Celcious <= 20

| | | | | ï»¿Month <= 10: Metria (5.0/1.0)

| | | | | ï»¿Month > 10: Mikri (3.0)

| | | | Celcious > 20: Mikri (24.0/1.0)

| | | Time Dept. > 1: Megali (5.0)

| | Arrival > 3

| | | Departure <= 9: Metria (24.0)

| | | Departure > 9

| | | | Wind <= 4

| | | | | Ship = ARIADNE: Mikri (0.0)

| | | | | Ship = MYCONOS: Mikri (0.0)

| | | | | Ship = EL.VENIZELOS: Mikri (0.0)

| | | | | Ship = DOD EXPRESS: Mikri (0.0)

| | | | | Ship = PAROS

| | | | | | Celcious <= 22: Mikri (9.0)

| | | | | | Celcious > 22

| | | | | | | Departure <= 10

| | | | | | | | Arrival <= 9: Metria (3.0/1.0)

| | | | | | | | Arrival > 9: Poli megali (7.0/1.0)

| | | | | | | | Departure > 10: Mikri (3.0)

| | | | | | Ship = KISSAMOS: Mikri (0.0)

| | | | | | Ship = CHIOS: Mikri (0.0)

| | | | | | Ship = DELOS: Metria (2.0)

| | | | | | Ship = ELYROS: Mikri (0.0)

| | | | | Wind > 4: Metria (8.0/1.0)

| | Port = DON

| | | Wind <= 3: Mikri (8.0)

| | | Wind > 3

| | | | Arrival <= 4

| | | | | Celcious <= 22: Megali (5.0)

| | | | | Celcious > 22

| | | | | | Celcious <= 26: Metria (9.0/2.0)

| | | | Celcious > 26: Megali (2.0/1.0)

| | | Arrival > 4

| | | | Arrival <= 8

| | | | Celcious <= 23: Mikri (4.0/1.0)

| | | | Celcious > 23: Poli megali (5.0/1.0)

| | | | Arrival > 8: Mikri (4.0/1.0)

| Port = ARD

| | Arrival <= 14: Poli megali (2.0)

| | Arrival > 14: Megali (2.0)

| Port = PAS

| | Departure <= 7: Poli megali (3.0)

| | Departure > 7

| | | Time Arr <= 12

| | | | ï»¿Month <= 10

| | | | Time Dept. <= 11

| | | | | Arrival <= 10

| | | | | | ï»¿Month <= 9: Metria (10.0)

| | | | | | ï»¿Month > 9

| | | | | | Celcious <= 20

| | | | | | | Celcious <= 19: Mikri (2.0)

| | | | | | | Celcious > 19: Metria (5.0)

| | | | | | | Celcious > 20: Mikri (4.0)

| | | | | | Arrival > 10: Mikri (5.0/1.0)

| | | | | Time Dept. > 11

| | | | | | Arrival <= 11: Metria (6.0/1.0)

| | | | | | Arrival > 11

| | | | | | Celcious <= 24: Metria (2.0/1.0)

| | | | | | Celcious > 24: Megali (2.0)

| | | | | »ζMonth > 10: Mikri (7.0)

| | | Time Arr > 12: Poli megali (2.0)

| Port = RHO

| | Time Dept. <= 8: Mikri (90.0/4.0)

| | Time Dept. > 8: Poli megali (4.0/1.0)

| Port = JSY

| | »ζMonth <= 8

| | | Wind <= 3: Mikri (13.0/1.0)

| | | Wind > 3

| | | | Wind <= 4: Metria (14.0/5.0)

| | | | Wind > 4

| | | | | Celcious <= 30: Mikri (17.0/4.0)

| | | | | Celcious > 30: Metria (3.0)

| | »ζMonth > 8: Mikri (17.0/1.0)

| Port = SHC

| | Time Arr <= 7

| | | Arrival <= 4

| | | | Time Arr <= 2: Mikri (16.0/1.0)

| | | | Time Arr > 2

| | | | | Wind <= 4: Metria (6.0/1.0)

| | | | | Wind > 4: Megali (4.0)

| | | Arrival > 4: Mikri (25.0/2.0)

| | Time Arr > 7: Metria (3.0)

| Port = JMK

| | Ship = ARIADNE: Mikri (0.0)

| | Ship = MYCONOS: Metria (3.0)

| | Ship = EL.VENIZELOS: Mikri (0.0)

| | Ship = DOD EXPRESS: Mikri (0.0)

| | Ship = PAROS

| | | Time Dept. <= 14: Mikri (59.0)

| | | Time Dept. > 14: Megali (3.0)

| | Ship = KISSAMOS: Mikri (0.0)

| | Ship = CHIOS: Mikri (0.0)

| | Ship = DELOS: Mikri (0.0)

| | Ship = ELYROS: Mikri (0.0)

| Port = LIP

| | Time Dept. <= 14

| | | Wind <= 4

| | | | ï»¿Month <= 7

| | | | Wind <= 3: Metria (3.0)

| | | | Wind > 3: Mikri (5.0/1.0)

| | | | ï»¿Month > 7: Mikri (50.0)

| | | Wind > 4

| | | | ï»¿Month <= 8

| | | | Time Dept. <= 12: Mikri (3.0)

| | | | Time Dept. > 12: Metria (12.0/3.0)

| | | | ï»¿Month > 8: Mikri (17.0/4.0)

| | Time Dept. > 14

| | | Arrival <= 13: Poli megali (7.0)

| | | Arrival > 13

| | | | Time Arr <= 14: Metria (11.0)

| | | | Time Arr > 14: Megali (2.0)

| Port = KAR: Megali (3.0/1.0)

| Port = KOF

| | Celcious <= 24

| | | Arrival <= 4

| | | | Time Arr <= 3

| | | | | Time Dept. <= 3: Mikri (11.0)

| | | | | Time Dept. > 3: Metria (2.0)

| | | | Time Arr > 3: Megali (2.0)

| | | Arrival > 4: Mikri (17.0)

| | Celcious > 24

| | | ï»¿Month <= 9

| | | | Arrival <= 4

| | | | | Time Arr <= 3: Megali (3.0/2.0)

| | | | | Time Arr > 3: Poli megali (2.0)

| | | | Arrival > 4: Metria (6.0)

| | | ï»¿Month > 9: Mikri (5.0)

| Port = AGN

| | Celcious <= 22: Megali (3.0)

| | Celcious > 22

| | | ï»¿Month <= 8: Metria (19.0/1.0)

| | | ï»¿Month > 8

| | | | Celcious <= 29: Mikri (7.0)

| | | | Celcious > 29: Metria (3.0)

| Port = VTH: Metria (3.0)

| Port = KMI

| | Time Arr <= 17

| | | Wind <= 4

| | | | ï»¿Month <= 9

| | | | | Celcious <= 28

| | | | | | Wind <= 3

| | | | | | | Celcious <= 25: Metria (8.0)

| | | | | | | Celcious > 25: Mikri (7.0)

| | | | | | | Wind > 3

| | | | | | | Celcious <= 26: Mikri (13.0)

| | | | | | | Celcious > 26: Metria (9.0/2.0)

| | | | | Celcious > 28: Mikri (8.0)

| | | | ï»¿Month > 9: Mikri (18.0/1.0)

| | | Wind > 4

| | | | Arrival <= 13: Metria (16.0/4.0)

| | | | Arrival > 13

| | | | | Time Dept. <= 15

| | | | | | Wind <= 6: Mikri (4.0)

| | | | | | Wind > 6: Metria (4.0/1.0)

| | | | | Time Dept. > 15: Metria (2.0)

| | Time Arr > 17: Poli megali (3.0)

```
| Port = ITBRI: Mikri (1.0)

| Port = MJT

| | Arrival <= 6

| | | Departure <= 6: Metria (2.0)

| | | Departure > 6: Poli megali (2.0)

| | Arrival > 6

| | | Arrival <= 8: Megali (2.0)

| | | Arrival > 8: Mikri (2.0)

| Port = TIN

| | Time Dept. <= 15

| | | Arrival <= 14

| | | | iMonth <= 7

| | | | | Wind <= 4: Metria (3.0)

| | | | | Wind > 4: Mikri (20.0/7.0)

| | | | iMonth > 7

| | | | | iMonth <= 8: Metria (32.0/7.0)

| | | | | iMonth > 8: Mikri (4.0/1.0)

| | | Arrival > 14

| | | | iMonth <= 7

| | | | | Wind <= 4: Metria (5.0/1.0)

| | | | | Wind > 4: Mikri (6.0)

| | | | iMonth > 7

| | | | | Wind <= 4: Mikri (26.0/4.0)

| | | | | Wind > 4

| | | | | Wind <= 5: Metria (8.0/2.0)
```

| | | | | Wind > 5

| | | | | Celcius <= 29: Metria (2.0)

| | | | | Celcius > 29: Mikri (2.0)

| | Time Dept. > 15

| | | Arrival <= 14: Poli megali (2.0)

| | | Arrival > 14: Megali (3.0)

| Port = KAS: Poli megali (4.0/1.0)

| Port = PKK

| | Celcius <= 20: Megali (3.0/1.0)

| | Celcius > 20

| | | ï»¿Month <= 7: Metria (6.0)

| | | ï»¿Month > 7

| | | | Time Arr <= 14

| | | | | Wind <= 4: Mikri (26.0/2.0)

| | | | | Wind > 4

| | | | | Celcius <= 21: Metria (2.0)

| | | | | Celcius > 21

| | | | | Celcius <= 24: Mikri (7.0)

| | | | | Celcius > 24

| | | | | | Arrival <= 13

| | | | | | Celcius <= 27: Metria (4.0)

| | | | | | Celcius > 27: Mikri (5.0/1.0)

| | | | | | Arrival > 13

| | | | | | Celcius <= 27: Mikri (2.0)

| | | | | | Celcius > 27: Metria (3.0/1.0)

| | | | Time Arr > 14

| | | | | ï»¿Month <= 8: Metria (9.0/1.0)

| | | | | ï»¿Month > 8: Mikri (5.0/1.0)

| Port = AIG

| | Arrival <= 2

| | | Time Arr <= 2: Metria (7.0/2.0)

| | | Time Arr > 2

| | | | ï»¿Month <= 9: Poli megali (2.0)

| | | | ï»¿Month > 9: Megali (5.0)

| | Arrival > 2

| | | Wind <= 4

| | | | Celcious <= 25: Mikri (18.0)

| | | | Celcious > 25

| | | | | Celcious <= 26

| | | | | | Time Dept. <= 3: Mikri (2.0)

| | | | | | Time Dept. > 3: Metria (5.0)

| | | | | Celcious > 26: Mikri (4.0)

| | | Wind > 4: Metria (8.0/3.0)

| Port = HRK

| | ï»¿Month <= 9

| | | Time Dept. <= 2: Mikri (5.0)

| | | Time Dept. > 2

| | | | Arrival <= 4

| | | | | Time Arr <= 2: Metria (2.0)

| | | | | Time Arr > 2: Poli megali (2.0)

| | | | Arrival > 4: Metria (9.0/2.0)

| | | | > 9

| | | | Arrival <= 4

| | | | Time Arr <= 2: Mikri (6.0/1.0)

| | | | Time Arr > 2: Megali (2.0)

| | | | Arrival > 4: Mikri (15.0/1.0)

| | Port = JTR

| | Ship = ARIADNE: Metria (0.0)

| | Ship = MYCONOS: Metria (0.0)

| | Ship = EL.VENIZELOS: Metria (0.0)

| | Ship = DOD EXPRESS: Metria (0.0)

| | Ship = PAROS

| | | | > 9: Metria (3.0)

| | | | > 9: Mikri (5.0/2.0)

| | Ship = KISSAMOS: Metria (0.0)

| | Ship = CHIOS: Megali (3.0)

| | Ship = DELOS: Metria (0.0)

| | Ship = ELYROS: Metria (0.0)

| | Port = PYT: Megali (1.0)

| | Port = AKI: Megali (2.0/1.0)

| | Port = NIS: Mikri (12.0/1.0)

| | Port = DIA: Megali (1.0)

| | Port = HAL

| | Time Dept. <= 9: Mikri (16.0)

| | Time Dept. > 9

| | | Arrival <= 12: Metria (3.0)

| | | Arrival > 12: Mikri (4.0)

| Port = JSH: Metria (3.0/1.0)

| Port = JKH: Metria (4.0/1.0)

| Port = AOK: Mikri (0.0)

| Port = KSJ: Megali (1.0)

| Port = JTY

| | Time Dept. <= 5: Metria (11.0/2.0)

| | Time Dept. > 5

| | | Arrival <= 4: Megali (4.0)

| | | Arrival > 4: Mikri (8.0)

| Port = KAK

| | >=Month <= 10: Metria (20.0/2.0)

| | >=Month > 10

| | | Celcious <= 19: Metria (2.0)

| | | Celcious > 19: Mikri (5.0)

| Port = GPA: Mikri (0.0)

| Port = CHQ: Mikri (0.0)

| Port = IOS: Metria (3.0)

| Port = HER: Metria (4.0)

| Port = TIL: Mikri (6.0)

| Port = MYR: Mikri (5.0/1.0)

| Port = ANA: Mikri (4.0/1.0)

| Port = PER: Mikri (2.0)

| Port = SKG: Mikri (0.0)

| Port = KVA: Mikri (0.0)

Departure > 15

| Port = IGO

| | Time Arr <= 11: Metria (3.0)

| | Time Arr > 11: Poli megali (8.0)

| Port = EYD: Metria (0.0)

| Port = PIR

| | Time Dept. <= 21

| | | Time Arr <= 15

| | | | Celcious <= 10: Mikri (3.0)

| | | | Celcious > 10

| | | | | Departure <= 17

| | | | | | Arrival <= 6: Metria (2.0)

| | | | | | Arrival > 6: Mikri (2.0)

| | | | | | Departure > 17: Metria (60.0/2.0)

| | | Time Arr > 15

| | | | Time Dept. <= 18: Metria (7.0/2.0)

| | | | Time Dept. > 18: Poli megali (8.0)

| | Time Dept. > 21

| | | Departure <= 21

| | | | Time Arr <= 18: Megali (9.0)

| | | | Time Arr > 18: Poli megali (2.0)

| | | Departure > 21

| | | | Time Dept. <= 22: Metria (20.0)

| | | | Time Dept. > 22: Megali (6.0)

```

| Port = SUD
| | Time Arr <= 19
| | | Time Dept. <= 15: Poli megali (2.0/1.0)
| | | Time Dept. > 15
| | | | Time Dept. <= 21: Metria (36.0)
| | | | Time Dept. > 21
| | | | | Departure <= 21: Megali (3.0)
| | | | | Departure > 21: Metria (22.0/4.0)
| | Time Arr > 19: Megali (2.0)
| Port = SYM
| | Time Arr <= 17: Mikri (9.0)
| | Time Arr > 17
| | | Celcious <= 26: Megali (4.0)
| | | Celcious > 26: Metria (7.0)
| Port = KGS
| | Time Arr <= 16
| | | Time Dept. <= 16
| | | | Wind <= 4
| | | | | ï»¿Month <= 8: Mikri (20.0/1.0)
| | | | | ï»¿Month > 8: Metria (12.0/5.0)
| | | | Wind > 4
| | | | | ï»¿Month <= 7
| | | | | | Celcious <= 26: Megali (6.0/2.0)
| | | | | | Celcious > 26: Metria (5.0)
| | | | | ï»¿Month > 7

```

| | | | | Celcious <= 22: Metria (3.0)

| | | | | Celcious > 22

| | | | | Celcious <= 26: Mikri (8.0)

| | | | | Celcious > 26: Metria (4.0/1.0)

| | | Time Dept. > 16: Metria (22.0)

| | Time Arr > 16

| | | ÿMonth <= 9: Metria (4.0/1.0)

| | | ÿMonth > 9: Poli megali (9.0)

| Port = PMS: Megali (2.0)

| Port = JNX

| | Time Dept. <= 21

| | | ÿMonth <= 10

| | | | ÿMonth <= 9: Metria (4.0/1.0)

| | | | ÿMonth > 9: Megali (8.0/1.0)

| | | ÿMonth > 10: Metria (5.0)

| | Time Dept. > 21: Mikri (5.0)

| Port = DON: Metria (0.0)

| Port = ARD: Metria (0.0)

| Port = PAS

| | Time Arr <= 9: Poli megali (4.0)

| | Time Arr > 9

| | | Time Dept. <= 23

| | | | Wind <= 3: Metria (5.0)

| | | | Wind > 3: Megali (17.0/4.0)

| | | Time Dept. > 23: Mikri (23.0/4.0)

```

| Port = RHO: Mikri (1.0)

| Port = JSY

| | Time Dept. <= 18

| | | ï»¿Month <= 8: Metria (55.0/5.0)

| | | ï»¿Month > 8

| | | | Celcious <= 29: Metria (6.0/2.0)

| | | | Celcious > 29: Mikri (3.0)

| | Time Dept. > 18

| | | Time Dept. <= 22

| | | | Arrival <= 19: Poli megali (6.0/1.0)

| | | | Arrival > 19

| | | | | Time Arr <= 21: Mikri (5.0)

| | | | | Time Arr > 21

| | | | | | Celcious <= 23: Metria (5.0)

| | | | | | Celcious > 23: Mikri (18.0/2.0)

| | | Time Dept. > 22

| | | | Time Arr <= 22: Megali (6.0)

| | | | Time Arr > 22: Poli megali (3.0)

| Port = SHC: Metria (0.0)

| Port = JMK: Mikri (3.0/1.0)

| Port = LIP: Megali (2.0)

| Port = KAR: Metria (0.0)

| Port = KOF: Metria (0.0)

| Port = AGN: Metria (0.0)

| Port = VTH: Metria (0.0)

```

- | Port = KMI
- | | Time Arr <= 15: Mikri (6.0/1.0)
- | | Time Arr > 15: Metria (8.0/1.0)
- | Port = ITBRI: Poli megali (6.0/1.0)
- | Port = MJT: Metria (0.0)
- | Port = TIN: Metria (0.0)
- | Port = KAS: Metria (0.0)
- | Port = PKK: Metria (0.0)
- | Port = AIG: Metria (0.0)
- | Port = HRK: Metria (0.0)
- | Port = JTR: Metria (5.0/1.0)
- | Port = PYT: Metria (0.0)
- | Port = AKI: Metria (0.0)
- | Port = NIS
- | | Wind <= 4
- | | | Wind <= 3: Metria (3.0)
- | | | Wind > 3: Mikri (6.0/1.0)
- | | Wind > 4: Metria (6.0)
- | Port = DIA: Metria (0.0)
- | Port = HAL
- | | $\ddot{y}_{\zeta} \text{Month} \leq 8$
- | | | $\ddot{y}_{\zeta} \text{Month} \leq 7$: Metria (2.0)
- | | | $\ddot{y}_{\zeta} \text{Month} > 7$
- | | | Celcious <= 28: Megali (2.0)
- | | | Celcious > 28: Metria (2.0)

| | »Month > 8: Mikri (5.0)

| Port = JSH: Metria (1.0)

| Port = JKH: Metria (0.0)

| Port = AOK: Metria (4.0/1.0)

| Port = KSJ: Metria (2.0)

| Port = JTY: Metria (0.0)

| Port = KAK: Metria (0.0)

| Port = GPA: Metria (10.0)

| Port = CHQ: Metria (6.0)

| Port = IOS: Metria (1.0)

| Port = HER: Metria (0.0)

| Port = TIL: Metria (1.0)

| Port = MYR: Metria (0.0)

| Port = ANA: Metria (0.0)

| Port = PER: Metria (3.0)

| Port = SKG: Mikri (1.0)

| Port = KVA: Metria (1.0)

Number of Leaves : 293

Size of the tree : 468

Time taken to build model: 0.23 seconds

=== Stratified cross-validation ===

=== Summary ===

Correctly Classified Instances	1678	81.6943 %
Incorrectly Classified Instances	376	18.3057 %
Kappa statistic	0.696	
Mean absolute error	0.1129	
Root mean squared error	0.2755	
Relative absolute error	37.1301 %	
Root relative squared error	70.6618 %	
Total Number of Instances	2054	

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	
Class									
	0.695	0.013	0.759	0.695	0.726	0.711	0.927	0.703	Poli megali
	0.595	0.022	0.677	0.595	0.633	0.608	0.870	0.624	Megali
	0.803	0.120	0.809	0.803	0.806	0.684	0.890	0.807	Metria
	0.876	0.148	0.847	0.876	0.861	0.728	0.906	0.870	Mikri
Weighted Avg.	0.817	0.120	0.815	0.817	0.816	0.701	0.898	0.818	

=== Confusion Matrix ===

a b c d <-- classified as

82 15 11 10 | a = Poli megali

10 88 33 17 | b = Megali

12 15 639 130 | c = Metria

4 12 107 869 | d = Mikri

Παράρτημα Β: «Κώδικας Python»

```
import pandas as pd

import numpy as np

import seaborn as sns

import matplotlib.pyplot as plt

from sklearn.model_selection import train_test_split

from sklearn.preprocessing import LabelEncoder, MinMaxScaler

from sklearn.metrics import confusion_matrix, classification_report, accuracy_score,
precision_score, recall_score, f1_score

import tensorflow as tf

from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import Dense


# Load dataset

df = pd.read_csv("data_final4.csv", encoding='cp1252')


# Debug column names

df.columns = df.columns.str.strip() # Remove extra spaces


# Print columns to check

print("Columns in dataset:", df.columns.tolist())


# Encode categorical columns (Ship, Port, Class)

categorical_cols = ["Ship", "Port", "Class"]

label_encoders = {}
```

```
for col in categorical_cols:

    le = LabelEncoder()

    df[col] = le.fit_transform(df[col])

    label_encoders[col] = le # Store for inverse transform later


# Features (X) & Target (y)

X = df.drop(columns=["Class"]).values # Features

y = df["Class"].values # Target


# Normalize features

scaler = MinMaxScaler()

X = scaler.fit_transform(X)


# Train-Test Split

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)


# Model definition

model = Sequential([

    Dense(16, activation='relu', input_shape=(X_train.shape[1],)),

    Dense(8, activation='relu'),

    Dense(len(np.unique(y)), activation='softmax') # Multi-class classification

])


# Compile model
```

```
model.compile(optimizer='adam',loss='sparse_categorical_crossentropy',  
metrics=['accuracy'])
```

```
# Train model
```

```
model.fit(X_train, y_train, epochs=100, batch_size=8, verbose=1)
```

```
# Evaluate model
```

```
loss, accuracy = model.evaluate(X_test, y_test)
```

```
print(f"Test Accuracy: {accuracy:.2f}")
```

```
# Predictions
```

```
y_pred = model.predict(X_test).argmax(axis=1)
```

```
# Confusion Matrix
```

```
cm = confusion_matrix(y_test, y_pred)
```

```
# Compute Metrics
```

```
accuracy = accuracy_score(y_test, y_pred)
```

```
precision = precision_score(y_test, y_pred, average='weighted')
```

```
recall = recall_score(y_test, y_pred, average='weighted')
```

```
f1 = f1_score(y_test, y_pred, average='weighted')
```

```
# Print Metrics
```

```
print("\n📊 Model Evaluation Metrics:")
```

```
print(f"✅ Accuracy: {accuracy:.2f}")
```

```
print(f"✅ Precision: {precision:.2f}")
```

```
print(f"✅ Recall: {recall:.2f}")
```

```
print(f"✅ F1 Score: {f1:.2f}")
```

```
# Classification Report
```

```
print("\n🔍 Classification Report:")
```

```
print(classification_report(y_test, y_pred))
```

```
# Plot Confusion Matrix
```

```
plt.figure(figsize=(6, 4))
```

```
sns.heatmap(cm, annot=True, fmt="d", cmap="Blues", xticklabels=np.unique(y),  
yticklabels=np.unique(y))
```

```
plt.xlabel("Predicted")
```

```
plt.ylabel("Actual")
```

```
plt.title("Confusion Matrix")
```

```
plt.show()
```

Στην συνέχεια αναλύουμε τον κώδικα python που αναφέρθηκε παραπάνω

1. Εισαγωγή Βιβλιοθηκών

```
import pandas as pd
```

```
import numpy as np
```

```
import seaborn as sns
```

```
import matplotlib.pyplot as plt
```

```
from sklearn.model_selection import train_test_split
```

```
from sklearn.preprocessing import LabelEncoder, MinMaxScaler
```

```
from sklearn.metrics import confusion_matrix, classification_report,  
accuracy_score, precision_score, recall_score, f1_score  
  
import tensorflow as tf  
  
from tensorflow.keras.models import Sequential  
from tensorflow.keras.layers import Dense
```

- **Pandas:** Για τη διαχείριση και επεξεργασία δεδομένων.
- **NumPy:** Για αριθμητικές εργασίες και επεξεργασία των δεδομένων.
- **Seaborn και Matplotlib:** Για την οπτικοποίηση των αποτελεσμάτων.
- **Scikit-learn:** Για τη διαχείριση και ανάλυση δεδομένων, καθώς και για την αξιολόγηση του μοντέλου.
- **TensorFlow και Keras:** Για την κατασκευή και εκπαίδευση του νευρωνικού δικτύου.

2. Φόρτωση Δεδομένων

```
df = pd.read_csv("data_final4.csv", encoding='cp1252')
```

- Φορτώνει το dataset από ένα αρχείο CSV σε ένα DataFrame.

Ένα DataFrame (Πλαίσιο Δεδομένων) ουσιαστικά είναι μια δομή δεδομένων της Pandas παρόμοια με έναν πίνακα στο Excel και παρέχει την δυνατότητα επεξεργασίας και αποθήκευσης των δεδομένων με τα εξής χαρακτηριστικά 1. Οργάνωση σε στήλες και γραμμές 2. Ετικέτες στηλών (columns) και ετικέτες γραμμών (indexes) 3. Διαφορετικοί τύποι δεδομένων σε κάθε column 4. Ευέλικτη επεξεργασία όπως εισαγωγή, μετατροπή, διαγραφή και υπολογισμό δεδομένων

3. Εξέταση Στηλών

```
df.columns = df.columns.str.strip() # Remove extra spaces  
print("Columns in dataset:", df.columns.tolist())
```

- Αφαιρεί τυχόν κενά από τα ονόματα των στηλών και εκτυπώνει τη λίστα με τις στήλες για επαλήθευση. Αυτό γίνεται ως εξής :

`df.columns` προσπέλαση στα ονόματα των στηλών του DataFrame

`str.strip()` μέθοδος συμβολοσειρών που αφαιρεί τυχόν κενά από την αρχή και το τέλος κάθε ονόματος στήλης

4. Κωδικοποίηση Κατηγορικών Στηλών

```
categorical_cols = ["Ship", "Port", "Class"]
```

```
label_encoders = {}
```

```
for col in categorical_cols:
```

```
    le = LabelEncoder()
```

```
    df[col] = le.fit_transform(df[col])
```

```
    label_encoders[col] = le # Store for inverse transform later
```

- Ο **LabelEncoder** είναι μια κλάση από την βιβλιοθήκη **sklearn** που κάθε κατηγορική στήλη (Ship, Port, Class) κωδικοποιείται σε αριθμό (encoder), οι οποίοι αποθηκεύονται για μελλοντική χρήση κυρίως για την διαδικασία του inverse transform

5. Ορισμός Χαρακτηριστικών και Στόχου

```
X = df.drop(columns=["Class"]).values # Features
```

```
y = df["Class"].values # Target
```

- Με την χρήση της μεθόδου `drop(columns=["Class"])` αφαιρούμε την στήλη-στόχος **“Class”** από το DataFrame ώστε η στήλη-στόχος να μην περιλαμβάνεται στα χαρακτηριστικά εκπαίδευσης του μοντέλου
- Έτσι δημιουργούμε δύο **numpy array** το X και το y όπου το X περιέχει όλα τα δεδομένα των στηλών “Ship” και “Port” (κωδικοποιημένα με την LabelEncoder) για τις προβλέψεις και το y με την στήλη με τις κατηγορίες μεταβλητής-στόχος “Class” για την εκπαίδευση του μοντέλου.

6. Κανονικοποίηση Χαρακτηριστικών

```
scaler = MinMaxScaler()
```

```
X = scaler.fit_transform(X)
```

- Κανονικοποιεί τα χαρακτηριστικά ώστε οι τιμές τους να είναι στο εύρος [0, 1] ως εξής: α. Αφαιρούμε από την τιμή που θέλουμε να ανάγουμε την ελάχιστη τιμή του χαρακτηριστικού ώστε να μετακινήσουμε την βάση μας στο 0. Στην συνέχεια το αποτέλεσμα το διαιρούμε με την διαφορά max-min ώστε να βρούμε την κανονικοποιημένη τιμή στο διάστημα [0,1]. Δημιουργούμε λοιπόν ένα αντικείμενο *“scaler”* της κλάσης *“MinMaxScaler()”* και στην συνέχεια με την μέθοδο *fit_transform(X)* κανονικοποιούμε τις τιμές του πίνακα X και το αποτέλεσμα καταχωρείται εκ νέου στον πίνακα X.

7. Διαχωρισμός Δεδομένων για Εκπαίδευση και Δοκιμή

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3,  
random_state=42)
```

- Διαχωρίζει τα δεδομένα σε εκπαιδευτικά (70%) και δεδομένα δοκιμής (30%). Εδώ χρησιμοποιούμε την συνάρτηση *“train_test_split”* για να διαχωρίσουμε τα δεδομένα μας. Ο διαχωρισμός είναι τυχαίος και χρησιμοποιούμε την παράμετρο 42 για να ορίσουμε ότι η διαίρεση θα είναι η ίδια εάν τρέξουμε τον κώδικα περισσότερες από μία φορές. Έτσι ο διαχωρισμός την πρώτη φορά θα είναι τυχαίος αλλά η διαίρεση θα είναι αναπαραγώγιμη. Η παράμετρος 0.3 ορίζει ότι το 30% των δεδομένων θα διαχωριστεί για τη δοκιμή, ενώ το 70% θα χρησιμοποιηθεί για την εκπαίδευση του μοντέλου. Ο πίνακας X είναι ο πίνακας των χαρακτηριστικών που περιέχει τα δεδομένα και ο πίνακας y είναι ο πίνακας στόχου που περιέχει τις τιμές που πρέπει να προβλέψουμε. Η συνάρτηση επιστέφει τέσσερις διαφορετικές μεταβλητές
- X_train* : Το σύνολο χαρακτηριστικών για εκπαίδευση.
- X_test* : Το σύνολο χαρακτηριστικών για δοκιμή.
- y_train* : Οι στόχοι για το σύνολο εκπαίδευσης.
- y_test* : Οι στόχοι για το σύνολο δοκιμής.

8. Ορισμός Μοντέλου

```
model = Sequential([  
    Dense(16, activation='relu', input_shape=(X_train.shape[1],)),  
    Dense(8, activation='relu'),  
    Dense(len(np.unique(y)), activation='softmax') # Multi-class classification  
)
```

- Δημιουργεί ένα νευρωνικό δίκτυο χρησιμοποιώντας την βιβλιοθήκη υψηλού επιπέδου **Keras**, που είναι ενσωματωμένη στην **TensorFlow**, με τρία επίπεδα: δύο κρυφά επίπεδα με 16 και 8 νευρώνες αντίστοιχα και ένα επίπεδο εξόδου για πολλαπλή κατηγοριοποίηση. Εδώ χρησιμοποιούμε την κλάση **“Sequential”** για την δημιουργία του νευρωνικού δικτύου η οποία μας επιτρέπει να προσθέσουμε διαδοχικά Layers. Ο τύπος των Layers είναι “Dense” που σημαίνει ότι κάθε νευρώνας (στήλη) του επιπέδου είναι συνδεδεμένος με κάθε νευρώνα του προηγούμενου και του επόμενου επιπέδου. Έτσι η πληροφορία από τον κάθε νευρώνα του προηγούμενου επιπέδου θα επηρεάζει τον κάθε νευρώνα του επόμενου επιπέδου. Έχουμε λοιπόν: **Είσοδοι** Κάθε νευρώνας σε ένα επίπεδο dense παίρνει όλες τις τιμές (features) από τους νευρώνες του προηγούμενου επιπέδου. **Εξόδοι** : Οι εξόδοι του επιπέδου dense υπολογίζονται εφαρμόζοντας μια γραμμική (linear) συνδυαστική συνάρτηση καθώς και μια λειτουργία ενεργοποίησης (activation) για να προσδώσουν μη γραμμικότητα (Διαφορετικά βάρη)

9. Συγκέντρωση Μοντέλου

```
model.compile(optimizer='adam', loss='sparse_categorical_crossentropy',  
metrics=['accuracy'])
```

- Καθορίζει τον βελτιστοποιητή (Adam) και την συνάρτηση απώλειας (sparse categorical crossentropy) που θα χρησιμοποιηθούν κατά την εκπαίδευση
- optimizer='adam'** : Ορίζουμε τον αλγόριθμο που θα χρησιμοποιήσουμε για την βελτιστοποίηση των βαρών του μοντέλου κατά την διαδικασία της εκπαίδευσης. Εδώ επιλέξαμε τον αλγόριθμο **adam (Adaptive Moment Estimation)** έναν δημοφιλή αλγόριθμο ο οποίος χρησιμοποιεί προσαρμοσμένα ρυθμιστικά βήματα

(learning rates) για κάθε παράμετρο, επιτρέποντας έτσι πιο γρήγορη και πιο αποτελεσματική εκπαίδευση.

- ***loss='sparse_categorical_crossentropy'*** : Καθορίζουμε την συνάρτηση απώλειας (loss Function), κατά την εκπαίδευση του μοντέλου, η οποία «βλέπει» την απόκλιση μεταξύ των προβλεπόμενων τιμών του μοντέλου και των πραγματικών τιμών. Η συνάρτηση που θα χρησιμοποιήσουμε είναι η ***'sparse_categorical_crossentropy'*** η οποία χρησιμοποιείται όταν έχουμε περισσότερες από δύο κλάσεις και οι ετικέτες στόχου είναι αριθμητικές και όχι one-hot κωδικοποιημένες (Κάθε τιμή της κατηγορίας είναι ένα διάνυσμα) που θα ήταν καταλληλότερη η ***categorical_crossentropy***.
- ***metrics=['accuracy']*** : Ορίζουμε ποιες μετρικές θα παρακολουθούνται κατά την εκπαίδευση και την αξιολόγηση του μοντέλου επιλέγοντας την πιο κοινώς χρησιμοποιούμενη μετρική (*accuracy=ακρίβεια και όχι ευαισθησία, Απώλεια κλπ*) για την αξιολόγηση της απόδοσης του μοντέλου ταξινόμησης η οποία και υπολογίζει το ποσοστό των σωστών προβλέψεων σε σχέση με το σύνολο των παρατηρήσεων.

10. Εκπαίδευση Μοντέλου

model.fit(X_train, y_train, epochs=100, batch_size=8, verbose=1)

- Αυτή η γραμμή εκπαιδεύει το μοντέλο χρησιμοποιώντας τα εκπαιδευτικά δεδομένα (X_train, y_train).
- ***epochs=100*** : Το μοντέλο εκπαιδεύεται για 100 εποχές (epochs) (Ο αριθμός 100 είναι ο ελάχιστος αριθμός για την επαρκή εκπαίδευση ενός μοντέλου), δηλ το μοντέλο θα εκπαιδευτεί 100 φορές με τα ίδια δεδομένα, κάθε φορά προσπαθώντας να βελτιώσει τα βάρη του με βάση την απώλεια ώστε να επιτύχει καλύτερες προβλέψεις και λιγότερες απώλειες.
- ***batch_size=8*** : Ορίζεται και το μέγεθος παρτίδας (batch_size) σε 8, που σημαίνει ότι το μοντέλο θα επεξεργάζεται 8 δείγματα σε κάθε βήμα εκπαίδευσης. Μια επιλογή μέχρι 32 συνήθως προσφέρει πιο ακριβείς εκτιμήσεις των παραγώγων λόγω περισσότερων ενημερώσεων των βαρών
- ***verbose=1*** : Σημαίνει ότι θα εκτυπώνονται μηνύματα σχετικά με την πορεία της εκπαίδευσης. Επιλογές 0 = Σιωπηλή Εκπαίδευση

1= Εκπαίδευση με πληροφορίες

2= Αναλυτική Εκπαίδευση

11. Αξιολόγηση του Μοντέλου

```
loss, accuracy = model.evaluate(X_test, y_test)
```

```
print(f"Test Accuracy: {accuracy:.2f}")
```

- Αξιολογεί την απόδοση του μοντέλου στα δεδομένα δοκιμής (X_test, y_test).
- Η μέθοδος evaluate επιστρέφει την απώλεια (loss) και την ακρίβεια (accuracy) του μοντέλου και τις αποθηκεύει στις μεταβλητές loss και accuracy.
- Εκτυπώνει την ακρίβεια των δοκιμών με μορφοποίηση (μέχρι δύο δεκαδικά ψηφία).

12. Προβλέψεις

```
y_pred = model.predict(X_test).argmax(axis=1)
```

- ***model.predict(X_test)*** : Η μέθοδος αυτή χρησιμοποιείται για να κάνει προβλέψεις με το εκπαιδευμένο μοντέλο για τα δεδομένα εισόδου στο σύνολο δοκιμής (X_test) Η έξοδος της ***.predict*** είναι γενικά ένας πίνακας με πιθανότητες που αν το μοντέλο είναι εκπαιδευμένο σε περισσότερες από δύο κατηγορίες οι προβλέψεις θα είναι ένα διάνυσμα με πιθανότητες για κάθε κατηγορία.
- ***argmax(axis=1)*** : Μας επιστέφει την κατηγορία με την υψηλότερη πιθανότητα για κάθε δείγμα στο σύνολο δοκιμής υποδεικνύοντας ότι (***axis=1***) θέλουμε να λάβουμε τον δείκτη της κατηγορίας με την μεγαλύτερη πιθανότητα.

13. Πίνακας Σύγχυσης (Confusion Matrix)

```
cm = confusion_matrix(y_test, y_pred)
```

- Δημιουργεί έναν πίνακα σύγχυσης (confusion matrix) που απεικονίζει τον αριθμό των σωστών και λανθασμένων προβλέψεων για κάθε κατηγορία.

14. Υπολογισμός Μετρικών

```
accuracy = accuracy_score(y_test, y_pred)
precision = precision_score(y_test, y_pred, average='weighted')
recall = recall_score(y_test, y_pred, average='weighted')
f1 = f1_score(y_test, y_pred, average='weighted')
```

- Υπολογίζει διάφορες μετρικές για την αξιολόγηση της απόδοσης του μοντέλου:

Accuracy: Το ποσοστό των σωστών προβλέψεων.

Precision: Ο λόγος σωστών θετικών προβλέψεων προς το σύνολο των θετικών προβλέψεων.

Recall: Ο λόγος σωστών θετικών προβλέψεων προς το σύνολο των πραγματικών θετικών παρατηρήσεων.

F1 Score: Ο μέσος αρμονικός του precision και recall.

Η παράμετρος **Weighted Average** δηλώνει ότι οι κατηγορίες με περισσότερα παραδείγματα θα έχουν μεγαλύτερο βάρος-επιροή στο τελικό αποτέλεσμα

15. Εκτύπωση Μετρικών

```
print("\n📊 Model Evaluation Metrics:")
print(f"✅ Accuracy: {accuracy:.2f}")
print(f"✅ Precision: {precision:.2f}")
print(f"✅ Recall: {recall:.2f}")
print(f"✅ F1 Score: {f1:.2f}")
```

- Εκτυπώνει συνοπτικά τις μετρικές που υπολογίστηκαν προηγουμένως με κατάλληλη μορφοποίηση.

16. Αναφορά Κατηγοριοποίησης (Classification Report)

```
print("\n🔍 Classification Report:")  
print(classification_report(y_test, y_pred))
```

- Εκτυπώνει μια λεπτομερή αναφορά κατηγοριοποίησης που περιέχει precision, recall και F1 score για κάθε κατηγορία, καθώς και υπολογισμούς για το συνολικό accuracy.

17. Οπτικοποίηση Πίνακα Σύγχυσης

```
plt.figure(figsize=(6, 4))  
sns.heatmap(cm, annot=True, fmt="d", cmap="Blues",  
xticklabels=np.unique(y), yticklabels=np.unique(y))  
plt.xlabel("Predicted")  
plt.ylabel("Actual")  
plt.title("Confusion Matrix")  
plt.show()
```

- **Δημιουργία Σχήματος:** Ορίζει το μέγεθος του σχήματος με `figsize=(6, 4)` για να είναι ευανάγνωστο.
- **Heatmap:** Χρησιμοποιεί τη συνάρτηση `heatmap` της Seaborn για να δημιουργήσει μια θερμική απεικόνιση του πίνακα σύγχυσης (`cm`).
 - `annot=True`: Ενεργοποιεί την απεικόνιση των αριθμητικών τιμών μέσα στα τετράγωνα.
 - `fmt="d"`: Δηλώνει ότι οι αριθμοί θα είναι ακέραιοι (whole numbers).
 - `cmap="Blues"`: Ορίζει το χρωματικό σχήμα που θα χρησιμοποιηθεί.
 - `xticklabels` και `yticklabels`: Χρησιμοποιούν τις μοναδικές τιμές του `y` για την ανάθεση ετικετών στους άξονες `x` και `y`.
- **Ετικέτες και Τίτλος:** Καθορίζει ετικέτες για τους άξονες `x` και `y` (`Predicted` και `Actual` αντίστοιχα) και προσθέτει έναν τίτλο (`Confusion Matrix`) στο σχήμα.
- **Εμφάνιση Σχήματος:** Η συνάρτηση `plt.show()` εμφανίζει το γράφημα.

Παράρτημα Γ: «Συντομογραφίες Λιμανιών»

Code	Greek name	Latin name
AGN	Αγαθονήσι	Agathonisi
PIL	Αγία Γαλήνη Ρεθύμνου	Agia Galini Rethimnou
701	ΑΓΙΑ ΓΑΛΗΝΗ ΡΕΘΥΜΝΟΥ (ΠΑΛΙΟ LOC CODE)	AGIA GALINI RETHIMNOU (OLD LOC CODE)
AEF	Αγία Ευφημία Κεφαλληνίας	Agia Efimia
AKM	Αγία Κυριακή (Τρίκερι Μαγνησίας)	Agia Kyriaki (Trikeri Magnisia)
AMR	Αγία Μαρίνα Αίγινας	Agia Marina Aiginas
GMA	Αγία Μαρίνα Γραμματικού Αττικής	Agia Marina Grammatikou Attikis
AML	Αγία Μαρίνα Λέρου	Agia Marina Lerou
AGM	Αγία Μαρίνα Φθιώτιδας	Agia Marina Fthiotidas
APE	Αγία Πελαγία Κυθήρων	Agia Pelagia
ROU	Αγία Ρουμέλη Χανίων	Agia Roumeli
AGT	Άγιοι Θεόδωροι	Agio Theodoroi
AOR	ΑΓΙΟΚΑΜΠΟΣ ΕΥΒΟΙΑΣ (ΩΡΕΩΝ)	
AGK	Αγιόκαμπος Λάρισας	Agiokampos Larissas
SGB	Άγιος Γεώργιος	Saint George
AGE	Άγιος Γεώργιος Εύβοιας	Agios Georgios
AGO	Άγιος Ευστράτιος	Agios Efstratios
705	ΑΓΙΟΣ ΙΩΑΝΝΗΣ ΜΑΓΝΗΣΙΑΣ	
AKI	Άγιος Κήρυκος	Agios Kirykos
AKO	Άγιος Κωνσταντίνος	Agios Konstantinos
ANI	Άγιος Νικόλαος Κρήτης	Agios Nikolaos, Kriti
AGF	Άγιος Νικόλαος Φωκίδας	Agios Nikolaos Fokidas
STN	Άγιος Νικόλαος Φωκίδας	Saint Nicolas
706	ΑΓΙΟΣ ΝΙΚΟΛΑΟΣ ΧΑΛΚΙΔΙΚΗΣ	
ASF	Άγιος Στέφανος Κέρκυρας	Agios Stefanos Corfu
AGG	Αγκίστρι Αίγινας	Agistri Aiginas
ANL	Αγνώντας Σκοπέλου	Agnontas Skopelos
AGA	Αγριά Μαγνησίας	Agria
ADL	Αδάμαντας Μήλου	Adamas Milos
AIG	Αιγιάλη Αμοργού	Aigiali Amorgou
AEG	Αίγινα	Aegina
EGI	Αιγίνιο Πιερίας	Eginion
AEN	Αίγιο	Aigion
EDI	Αιδηψός Εύβοιας	Edipsos
AKT	Άκτιο Βόνιτσας	Aktio Vonitsas
AXD	Αλεξανδρούπολη	Alexandroupolis
LVR	Αλιβέρι	Aliverio
TRBXN	Αλικαρνασός	Halikarnassus (Bodrum)
063	ΑΛΜΥΡΟΠΟΤΑΜΟΣ	
ALO	Αλόνησος	Alonnisos

ATS	Αλτσι Σητείας	Altsi
711	ΑΛΥΚΗ ΒΟΛΟΥ	
APL	Αμαλιάπολη Μαγνησίας	Amaliapolis
AMI	Αμμουλιανή	Amoliani
AMF	Αμφιλοχία	Amfilochia
ANA	Ανάφη Κυκλάδων	Anafi
AND	Άνδρος	Andros
ITAOI	Ανκόνα Ιταλίας	Ancona
ATK	Αντίκυρα	Antikyra
ANP	Αντίπαρος	Antiparos
ANT	Αντίρριο	Andirrion
ARX	Άραξος Αχαΐας	Araxos
ARM	Αργοστόλι Κεφαλληνίας	Argostolion
ARK	Αρκίτσα Φθιώτιδας	Arkitsa
ARD	Αρκοί Δωδεκανήσου	Arkioi Dodekanissou
718	ΑΡΜΕΝΙΣΤΗΣ ΡΟΔΟΥ	
TEM	Αρτέμιδα (Λούτσα)	Artemis
SOS	Άσος Κεφαλληνίας	Assos
APV	Ασπροβάλτα	Asprovalta
ASS	Ασπρόπυργος	Aspropirgos
AST	Αστακός Αιτωλοακαρνανίας	Astakos
ARS	Άστρος Κυνουρίας	Astros
JTY	Αστυπάλεια	Astypalaia
ATA	Αταλάντη Φθιώτιδας	Atalandi
AKH	Αχίλλειο Μαγνησίας	Akhillion
ACL	Αχλάδι	Achladi
721	ΒΑΘΥ ΚΑΛΥΜΝΟΥ	
VTH	Βαθύ Σάμου	Vathy (Port Vathy)
VSS	Βασιλική Λευκάδας	Vassiliki Leykadas
ITVCE	Βενετία Ιταλίας	Venice
VLC	Βολισσός Χίου	Volissos (Limnia) Chios
VOL	Βόλος	Volos
VLM	Βουλιαγμένη	Vouliagmeni
VRI	Βρυσάκια Εύβοιας	Vrisakia
GLX	Γαλαξίδι	Galaxidi
GAL	Γαλατάς Τροιζηνίας	Galatas Troizinas
GAD	Γαύδος Χανίων	Gavdos Chanea
GAV	Γάυριο Άνδρου	Gavrion
YER	Γερακινή Χαλκιδικής	Yerakini
YAL	Γιάλτρα	Yaltra
GLY	Γλύφα Φθιώτιδας	Glyfa Fthiotidas
GFD	Γλυφάδα Σαρωνικού	Glyfada Saronikou
GLO	Γλώσσα Σκοπέλου	Glossa
YLI	Γυαλί Δωδεκανήσου	Yali Island

GYT	Γύθειο	Gythion
TR-CKZ	Δαρδανέλλια	Canakkale
DAF	Δάφνη Αγίου Όρους	Dafni
730	ΔΕΡΒΕΝΙ ΚΟΡΙΝΘΙΑΣ	
DOS	Δήλος	Delos
DIK	Διακόφτι Κυθήρων	Diakofti Kythiron
DIA	Διαφάνι Καρπάθου	Diafanion
DMB	Δομβραΐνα Βοιωτίας	Dombraina Boiotias
DON	Δονούσα Κυκλάδων	Donousa Kykladon
DPA	Δραπετσώνα	Drapetzona
DRE	Δρέπανο Ρίου	Drepano Riou
ALDRZ	Δυρράχιο Αλβανίας	Durres Albania
XZX	Ελαιοχώρια Χαλκιδικής	Elaiochoria
ELN	Ελαφόνησος Λακωνίας	Elafonissos Lakonias
ELT	Ελευθερές Καβάλας	Eleftheres
EEU	Ελευσίνα	Eleusina
735	ΕΠΑΝΟΜΗ ΘΕΣΣΑΛΟΝΙΚΗΣ	
EPI	Επίδαυρος	Epidavros
ETK	Ερατεινή Φωκίδας	Eratini
ERK	Ερεϊκούσα	Ereikousses Diapontioi Islands
737	ΕΡΕΣΣΟΣ ΛΕΣΒΟΥ	
ERE	Ερέτρια Εύβοιας	Eretria
ERM	Ερμιόνη	Ermioni
EYD	Εύδηλος Ικαρίας	Eydilos
ZTH	Ζάκυνθος	Zakynthos
IGO	Ηγουμενίτσα	Igoumenitsa
HRK	Ηρακλεία Κυκλάδων	Herakleia Kykladon
HER	Ηράκλειο	Heraklion (Iraklion)
TSO	Θάσος	Thasos
TKO	Θέση Κύριλλο	Thessi Kyrillo
SKG	Θεσσαλονίκη	Thessaloniki
SNK	Θεσσαλονίκη	Salonika
JTR	Θήρα	Thira
TRS	Θηρασιά	Thirasia
THM	Θύμεια	Thymaina Island
739	ΙΕΡΑΞ (ΓΕΡΑΚΑΣ) ΛΑΚΩΝΙΑΣ	IERAX (GERAKAS) LAKONIAS
IER	Ιεράπετρα	Ierapetra
IES	Ιερισσός	Ierissos
ITH	Ιθάκη	Ithaki (Ithaca)
IOS	Ίος	Ios
ITM	Ίσθμια	Isthmia
ITA	Ιτέα	Itea
KVA	Καβάλα	Kavala
KLD	Κάλαθος Ρόδου	Kalathos Rodou

KLM	Καλαμάκι Ισθμίων	Kalamaki
KLX	Καλαμάτα	Kalamata
KAI	Κάλαμος (Λευκάδας)	Kalamos Island (Lefkada)
743	ΚΑΛΛΟΝΗ ΛΕΣΒΟΥ	
KLL	Καλοί Λιμένες Ηρακλείου	Kalilimenes
KMI	Κάλυμνος	Kalymnos
KRE	Καμάρες Σίφνου	Kamares Sifnos
KSK	ΚΑΜΕΙΡΟΣ ΣΚΑΛΑ ΡΟΔΟΥ	KAMEIROS SKALA RHODES
KAG	Καργκινάρι Ικαρίας	Karkinagri Ikarias
KRM	Καρδάμaina Κω	Kardamaina Kos
KDM	Καρδάμυλα Χίου	Kardamyla
KAR	Καρλόβασι Σάμου	Neon Karlovasion
AOK	Κάρπαθος	Karpathos
KST	Κάρυστος Εύβοιας	Karystos
KSJ	Κάσος	Kassos
749	ΚΑΣΣΑΝΔΡΑ ΧΑΛΚΙΔΙΚΗΣ	
KSP	Κασσιόπη Κέρκυρας	Kassiopi Corfu
KIS	Καστέλι Κισσάμου Χανίων	Kastelli Kissamou
KAS	Καστελόριζο (Μεγίστη)	Kastellorizou (ex Megisti)
KSI	Καστός	Kastos Island
KAK	Κατάκολο	Katakolon
KTP	Κατάπολα Αμοργού	Katapola
KEI	Κατερίνη	Katerini
KAH	Κάτω Αχαΐα	Kato Achaia
KAP	Καψάλι Κυθήρων	Kapsalion (Kythira)
KEA	Κέα	Kea (Tzia)
751	ΚΕΡΑΜΕΙΑ ΛΕΣΒΟΥ	KERAMEIA LESVOS
KER	Κεραμωτή	Keramoti
KTS	Κερατσίνι	Keratsinion
CFU	Κέρκυρα	Kerkira (Corfu)
KEF	Κέφαλος Κω	Kefalos Ko
KIO	Κιάτο	Kiato
KMS	Κίμωλος	Kimolos
753	ΚΙΤΡΟΣ ΠΙΕΡΙΑΣ	
KLI	Κλήμα Σκοπέλου	Klima
KIN	Κοιλάδα Ερμιονίδας	Koilas Ermionidos
VAB	Κόλπος Νεάπολης Βοιών	Vatika Bay
KNS	Κοντιάς Λήμνου	Kontias Limnou
755	ΚΟΡΘΙΟΝ ΑΝΔΡΟΥ	
KRT	Κόρινθος	Korinthos
756	ΚΟΡΩΝΗ ΜΕΣΣΗΝΙΑΣ	
KEM	Κόστα Ερμιονίδας	Kosta Ermionidas
757	ΚΟΤΡΩΝΑΣ ΛΑΚΩΝΙΑΣ	
758	ΚΟΤΣΙΝΑΣ ΛΗΜΝΟΥ	

760	ΚΟΥΛΟΥΡΑ ΑΙΓΙΟΥ	
TRKUS	Κουσάντασι Τουρκίας	Kusadasi Turkey
KOF	Κουφονήσια Κυκλάδων	Koufonissia Kykladon
KYT	Κύθνος	Kythnos
KYL	Κυλλήνη	Kyllini
KYM	Κυμάσι Ευβοίας	Kimasi
KIM	Κύμη Εύβοιας	Kimi (Kymi)
762	ΚΥΠΑΡΙΣΣΙ ΛΑΚΩΝΙΑΣ	
KIP	Κυπαρισσία	Kiparissia
TR-IST	Κωνσταντινούπολη	Istanbul
KGS	Κως	Kos
PKK	Λακκί Λέρου	Lakkion
LRY	Λάρυμνα Φθιώτιδας	Larymna
LAV	Λαύριο	Lavrion (Laurium)
LIP	Λειψοί	Lipsoi
LEV	Λευκάδα	Lefkada (Lefkas)
LEF	Λευκαντί Εύβοιας	Lefkandi
LFK	Λευκίμμη	Lefkimmi
LND	Λεωνίδιο	Leonidion
LIX	Ληξούρι Κεφαλληνίας	Lixuri
767	ΛΙΜΕΝΑΡΙΑ ΘΑΣΟΥ	
LII	Λίμνη Ευβοίας	Limni
LIA	Λιναριά Σκύρου	Linaria
LDR	Λίνδος Ρόδου	Lindos Rodou
LIN	Λινοπεράματα Ηρακλείου	Linoperamata Hrakliou
769	ΛΟΥΤΡΑΚΙ ΚΟΡΙΝΘΙΑΣ	
LTR	Λουτρό Χανίων	Loutro Chanion
MTR	Μαθράκι	Mathraki Diapontioi Island
770	ΜΑΚΡΗ ΕΒΡΟΥ	
856	ΜΑΚΡΟΝΗΣΟΣ	MAKRONISOS
MSA	Μαλεσίνα Φθιώτιδας	Malessina
MKP	ΜΑΡΑΘΟΚΑΜΠΟΣ ΣΑΜΟΥ	
772	ΜΑΡΑΘΟΥΠΟΛΗ ΜΕΣΣΗΝΙΑΣ	
MRM	Μαρμάρι Εύβοιας	Marmari
773	ΜΑΡΠΗΣΣΑ ΠΑΡΟΥ	MARPISSA PAROS
099	ΜΑΡΩΝΕΙΑ ΑΓ. ΧΑΡΑΛΑΜΠΟΥ	
774	ΜΑΡΩΝΕΙΑ ΣΗΤΕΙΑΣ	
MXK	Μαστιχάρι Κω	Mastichari Kos
MGN	Μεγανήσι	Meganisi
MEA	Μέθανα	Methana
MET	Μεθώνη Μεσσηνίας	Methoni
MEN	Μενίδι Αιτωλοακαρνανίας	Menidi
MEL	Μεσολόγγι	Mesolongion
MHI	Μεστά Χίου	Mesta Hiou

MYL	Μηλάκι Ευβοίας	Mylaki
777	ΜΗΛΙΝΗ ΒΟΛΟΥ	
MOL	Μόλυβος	Molyvos
MON	Μονεμβασιά	Monemvasia
MVO	Μονή Βατοπεδίου	Moni Vatopediou
780	ΜΟΝΗ ΙΒΗΡΩΝ ΑΓΙΟΥ ΟΡΟΥΣ	
MDR	Μούδρος	Moudhros
781	ΜΟΥΡΤΟΣ ΘΕΣΠΡΩΤΙΑΣ	
MOU	Μουτσούνα Νάξου	Moutsouna
782	ΜΟΧΛΟΣ ΣΗΤΕΙΑΣ	
ITBRI	Μπάρι Ιταλίας	Bari
783	ΜΠΑΤΣΙ ΑΝΔΡΟΥ	
JMK	Μύκονος	Mykonos
784	ΜΥΛΟΙ ΑΡΓΟΥΣ	
MYR	Μύρινα	Mirina
MRS	Μυρτιές Καλύμνου	Mirties Kalimnou
MYB	Μύρτος Κεφαλληνίας	Myrtos Beach
785	ΜΥΤΙΚΑΣ ΑΙΤΩΛ/ΝΙΑΣ	
MJT	Μυτιλήνη	Mytilene
814	ΝΑ ΜΗ ΔΟΘΕΙ ΣΕ ΛΙΜΑΝΙ Ο ΚΩΔΙΚΟΣ ΑΥΤΟΣ, ΝΑ ΔΙΑΓΡΑΦΕΙ	ΕΙΝΑΙ Ο ΠΑΛΙΟΣ ΚΩΔΙΚΟΣ ΓΙΑ ΤΟ ΠΕΥΚΙ ΕΥΒΟΙΑΣ
JNX	Νάξος	Naxos
NAS	Νάουσα Πάρου	Naoussa Parou
NPK	Ναύπακτος	Navpaktos
NAF	Ναύπλιο	Nafplion
NAR	Νέα Αρτάκη Εύβοιας	Nea Artaki
NKA	Νέα Καλλικράτεια Χαλκιδικής	Nea Kallikratia
NKV	Νέα Καρβάλη Καβάλας	Nea Karvali
791	ΝΕΑ ΜΗΧΑΝΙΩΝΑ ΘΕΣ/ΝΙΚΗΣ	
NMA	Νέα Μουδανιά	Nea Moudhania
902	ΝΕΑ ΠΕΡΑΜΟΣ	
NPY	Νέα Πλάγια Χαλκιδικής	Nea Plagia Chalkidiki
792	ΝΕΑ ΠΟΤΙΔΑΙΑ ΧΑΛΚΙΔΙΚΗΣ	
NST	Νέα Στύρα Εύβοιας	Nea Stira
NEA	Νεάπολη Βοιών	Neapolis
793	ΝΕΟΣ ΜΑΡΜΑΡΑΣ ΧΑΛΚΙΔΙΚΗΣ	
796	ΝΙΚΗΤΗ ΧΑΛΚΙΔΙΚΗΣ	
NIS	Νίσυρος	Nisiros
797	ΝΤΙΠΙΟΝ ΛΕΣΒΟΥ	
NID	Νυδρί	Nidhrion
XKB	Ξηροκάμπι Λακωνίας	Xirokambion
798	ΞΥΛΟΚΑΣΤΡΟ ΚΟΡΙΝΘΙΑΣ	
OTH	Οθωνοί	Othonoi
OIK	Οία Θήρας	Oia Kyklades
INO	Οινούσσες	Inousses

800	ΟΛΥΜΠΙΑΔΑ ΧΑΛΚΙΔΙΚΗΣ	
PNT	Όρμος Πανόρμου Τήνου	Ormos Panormou
OUR	Ουρανούπολη	Ouranopolis
PEP	Παλαιά Επίδαυρος	Palaia Epidavros
PFA	Παλαιά Φώκαια	Palaia Fokaia
802	ΠΑΛΑΙΟΚΑΣΤΡΙΤΣΑ ΚΕΡΚΥΡΑΣ	
PSF	Παλαιοχώρα Σφακίων	Paleohora Sfakion
PCH	Παλαιόχωρα Χανίων	Paleochora
803	ΠΑΛΛΑΙΡΟΣ ΑΙΤΩΛΟΑΚΑΡΝΑΝΙΑΣ	
805	ΠΑΛΙΟΥΡΙ ΧΑΛΚΙΔΙΚΗΣ	
PAO	Παλούκια Σαλαμίνας	Paloukia
806	ΠΑΜΦΙΛΑ ΛΕΣΒΟΥ	
807	ΠΑΝΑΓΙΟΥΔΑ ΛΕΣΒΟΥ	
088	ΠΑΝΟΡΜΙΤΗΣ ΣΥΜΗΣ	
PAX	Παξοί	Paxi
PAR	Πάργα	Parga
PAS	Πάρος	Paros
PMS	Πάτμος	Patmos
GPA	Πάτρα	Patras
810	ΠΑΧΕΙΑ ΑΜΜΟΣ ΙΕΡΑΠΕΤΡΑ	
PME	Πάχη Μεγάρων	Pachi
PIR	Πειραιάς	Piraeus
PEL	Πελασγία	Pelasyia
PRR	Περαία Θεσσαλονίκης	Perea
PER	Πέραμα Αττικής	Perama
PGM	Πέραμα Γέρας Μυτιλήνης	Perama Geras Mytilinis
PSK	Πεσάδα Κεφαλληνίας	Pessada Kefallinnias
PEB	Πετανοί Κεφαλληνίας	Petani Beach
PLV	Πέτρα Λέσβου	Petra
PRE	Πετρίες Εύβοιας	Petries Evvoias
813	ΠΕΤΡΙΤΗΣ ΚΕΡΚΥΡΑΣ	
PYV	Πευκί Εύβοιας	Peyki Evvoias
851	ΠΕΥΚΟΧΩΡΙ ΧΑΛΚΙΔΙΚΗΣ	
PIS	Πισαετός Ιθάκης	Pissaetos Ithakis
857	ΠΙΣΩ ΛΙΒΑΔΙ ΠΑΡΟΥ	PISO LIVADI PAROS
815	ΠΛΑΚΑ ΛΗΜΝΟΥ	
ALG	Πλακιάς Ρεθύμνου	Plakias Rethimnou
855	ΠΛΑΚΙΑΣ ΡΕΘΥΜΝΟΥ (ΠΑΛΙΟ LOCODE)	PLAKIAS RETHIMNOU (OLD LOCODE)
816	ΠΛΑΤΑΜΩΝΑΣ ΠΙΕΡΙΑΣ	
PIA	Πλατανιά Μαγνησίας	Platania
PTI	Πλαταριά Θεσπρωτίας	Plataria
PLG	Πλατυγιάλι Αιτωλοακαρνανίας	Platygiali Etoloakarnanias
PIT	Πλύτρα Λακωνίας	Pitra
PLM	Πλωμάρι Λέσβου	Plomarion

PTK	Πολιτικά Ευβοίας	Politika
PLC	Πολιχνίτος Λέσβου	Polychnitos Lesvou
PKE	Πόρος Κεφαλληνίας	Poros Kefallinias
PTR	Πόρος Τροιζηνίας	Poros Trizinias
820	ΠΟΡΤΟ ΚΑΓΙΟ ΛΑΚΩΝΙΑΣ	
854	ΠΟΡΤΟ ΚΑΡΡΑ ΧΑΛΚΙΔΙΚΗΣ	
PTL	Πόρτο Λάγος Ξάνθης	Lagos (Porto-Lago)
821	ΠΟΡΤΟ ΛΑΦΙΑ ΕΥΒΟΙΑΣ	
PHE	Πορτοχέλι	Porto Heli
POA	Ποταμός Αντικυθήρων	Potamos Antikithera
PVK	Πρέβεζα	Preveza/Lefkas
PPI	Πρίνος Θάσου	Prinos Thassou
ITBDS	Πρίντεζι Ιταλίας	Brindisi
PTE	Πτελεός Μαγνησίας	Pteleos
823	ΠΥΔΝΑ ΠΙΕΡΙΑΣ	
PYT	Πυθαγόρειο Σάμου	Pythagoreion
PYL	Πύλος Μεσσηνίας	Pylos
PYR	Πύργος	Pyrgos
824	ΠΥΡΓΟΣ ΚΑΡΥΣΤΙΑΣ ΕΥΒΟΙΑΣ	
RAF	Ραφήνα	Rafina
825	ΡΑΧΑΙ ΦΘΙΩΤΙΔΑΣ	
REV	Ρεβυθούσα	Revithoussa
RET	Ρέθυμνο	Rethymnon
RIO	Ρίο	Rio
RHO	Ρόδος	Rhodes
SGT	Σαγιάδα Θεσπρωτίας	Sagiada Thesprotias
SAL	Σαλαμίνα	Salamis
SAI	Σάμη Κεφαλληνίας	Sami
SAM	Σαμοθράκη	Samothraki
SMI	Σάμος	Samos
SER	Σέριφος	Seriphos
JSH	Σητεία	Siteia
SGR	Σίγρι Λέσβου	Sigri Lesvou
SII	Σίκινος	Sikinos
SIK	Σικωνία	Sikionia
SKM	Σκάλωμα Θεσπρωτίας	Skaloma
SKA	Σκαραμαγκάς	Skaramangas
SKD	Σκαφιδιά Ηλείας	Skafidia
JSI	Σκιάθος	Skiathos
SKO	Σκόπελος	Skopelos
SKU	Σκύρος	Skiros
SLA	Σουβάλα Αίγινας	Souvala
SOG	Σούγια Χανίων	Sougia Chanion
SUD	Σούδα Χανίων	Suda Bay

SOR	Σούρπη	Soupri
SPE	Σπέτσες	Spetsai
SHA	Σταυρός Χαλκιδικής	Stavros Halkidikis
SCR	Σταυροχώρι	Stavrochorion
833	ΣΤΟΜΙΟ ΚΟΡΙΝΘΙΑΣ	
STI	Στρατών Χαλκιδικής	Stratoni
SYS	Στυλίδα	Stylida (Stylis)
834	ΣΥΚΑΜΙΝΕΑ ΛΕΣΒΟΥ	
SYM	Σύμη	Symi
JSY	Σύρος	Syros (Syrá)
SHI	Σχοινάρι Ζακύνθου	Shinari Zakynthou
SHC	Σχοινούσα Κυκλάδων	Schinousa Kykladon
TIL	Τήλος	Tilos
TIN	Τήνος	Tinos
TLA	Τολό Αργολίδας	Tolo Argolidos
TR901	ΤΟΥΡΚΟΥΤ ΡΕΪΣ	TURGUTREIS
TRK	Τρίκερι Μαγνησίας	Trikeri
TRY	Τρυπητή Χαλκιδικής	Trypiti Chalkidikis
TSI	Τσιγγέλι Βόλου	Tsingeli
840	ΤΥΡΟΣ ΑΡΚΑΔΙΑΣ	
HYD	Ύδρα	Hydra (Idhra)
FNR	Φανερωμένη Σαλαμίνας	Faneromeni Salaminas
FAM	Φαρμακονήσι Δωδεκανήσου	Farmakonissi Dodekanissos
FIS	Φισκάρδο Κεφαλληνίας	Fiskardo
FOL	Φολέγανδρος	Folegandros
FOU	Φούρνοι Κορσεών	Fournoi
SAT	Φρίκες Ιθάκης	Frikes Ithakis
HAL	Χάλκη Δωδεκανήσου	Halki Dodekanison
CLK	Χαλκίδα	Chalkida
CHQ	Χανιά	Chania
JKH	Χίος	Chios
847	ΧΟΡΕΥΤΟ ΒΟΛΟΥ	
905	ΧΡΥΣΟΜΗΛΙΑ (ΦΟΥΡΝΟΙ)	
CSF	Χώρα Σφακιών	Khora Sfakion
848	ΨΑΘΟΠΥΡΓΟΣ ΑΧΑΪΑΣ	
PAA	Ψαρά	Psara
PSA	Ψαχνά Εύβοιας	Psakhna
PSE	Ψέριμος Δωδεκανήσου	Phserimos Dodekanisou
ORE	Ωρεοί Εύβοιας	Orei
ORO	Ωρωπός	Oropos

Παράρτημα Δ: «Αυτοματοποιημένο σενάριο (automation script) σε γλώσσα προγραμματισμού Python»

```
from playwright.sync_api import sync_playwright
from datetime import date
import time

def page_scrot():
    with sync_playwright() as playwright:
        browser = playwright.chromium.launch(headless=True)

        current_date = date.today()

        # Creating a new page with ignore_https_errors set to True
        page = browser.new_page(ignore_https_errors=True)

        # Ariadne
        page.goto("https://marinetraffic.live/vessels.php/ariadne-
position/240580000/#9135262")
        time.sleep(15)
        page.screenshot(path=f"ariadne_{current_date}.png", full_page=True)

        # Chios
        page.goto("https://marinetraffic.live/vessels.php/blue-star-chios-
position/240672000/#9215555")
        time.sleep(15)
        page.screenshot(path=f"chios_{current_date}.png", full_page=True)

        # Delos
        page.goto("https://marinetraffic.live/vessels.php/blue-star-delos-
position/241087000/#9565039")
        time.sleep(15)
        page.screenshot(path=f"delos_{current_date}.png", full_page=True)

        # El.Venizelos
        page.goto("https://marinetraffic.live/vessels.php/el-venizelos-
position/237628000/#7907673")
        time.sleep(15)
        page.screenshot(path=f"el-venizelos_{current_date}.png", full_page=True)

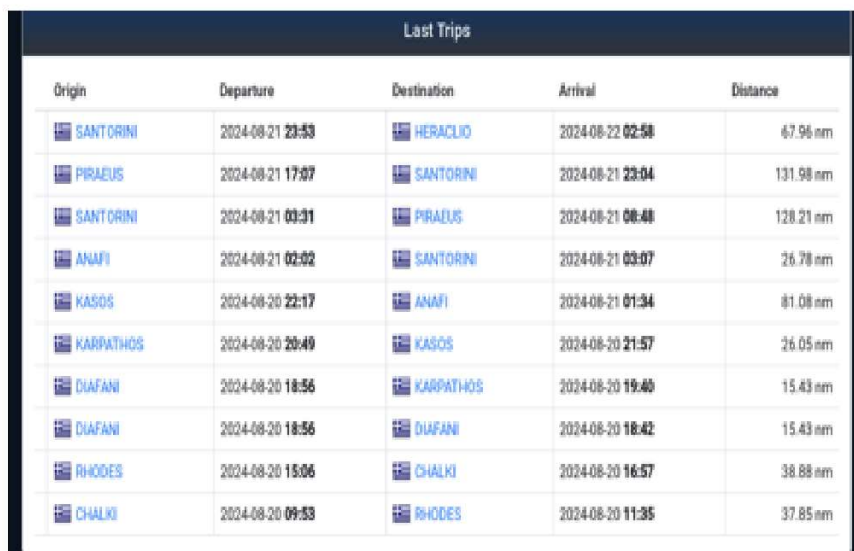
        # Elyros
        page.goto("https://marinetraffic.live/vessels.php/elyros-
position/240685000/#9178599")
        time.sleep(15)
        page.screenshot(path=f"elyros_{current_date}.png", full_page=True)

        # Kissamos
```

```
page.goto("https://marinetraffic.live/vessels.php/kissamos-  
position/239672000/#9035876")  
time.sleep(15)  
page.screenshot(path=f"kissamos_{current_date}.png", full_page=True)  
  
# Myconos  
page.goto("https://marinetraffic.live/vessels.php/blue-star-myconos-  
position/240389000/#9208679")  
time.sleep(15)  
page.screenshot(path=f"myconos_{current_date}.png", full_page=True)  
  
# Paros  
page.goto("https://marinetraffic.live/vessels.php/blue-star-paros-  
position/239924000/#9241774")  
time.sleep(15)  
page.screenshot(path=f"paros_{current_date}.png", full_page=True)  
  
# Dodekanisos Express  
page.goto("https://marinetraffic.live/vessels.php/dodekanisos-express-  
position/237176400/#0")  
time.sleep(15)  
page.screenshot(path=f"dodekanisos-express_{current_date}.png", full_page=True)  
  
browser.close()  
  
if __name__ == "__main__":  
    page_scrot()
```

Στην συνέχεια αναλύουμε τον κώδικα python που αναφέρθηκε παραπάνω

Ο κώδικας χρησιμοποιεί τη βιβλιοθήκη Playwright για να τραβήξει στιγμιότυπα οθόνης διαφορετικών πλοίων από μία συγκεκριμένη ιστοσελίδα, με τις εικόνες να αποθηκεύονται με την τρέχουσα ημερομηνία. Ένα δείγμα μιας εικόνας βλέπουμε παρακάτω



Εικόνα 17

Δείγμα
στιγμιότυπου
οθόνης από
αυτοματοποιημένο
σύστημα σε Linux

1.Εισαγωγές

```
from playwright.sync_api import sync_playwright
from datetime import date
import time
```

- **from playwright.sync_api import sync_playwright:** Αυτή η γραμμή εισάγει το sync_playwright από την βιβλιοθήκη Playwright. Έτσι έχουμε την δυνατότητα συγγραφής κώδικα για την εκτέλεση περιήγησης σε διάφορες μηχανές αναζήτησης προγράμματος περιήγησης, όπως Chrome, Firefox και WebKit. Η sync μορφή επιτρέπει συγχρονισμένη χρήση, κάτι που σημαίνει ότι ο κώδικας εκτελείται βήμα προς βήμα, διευκολύνοντας την κατανόηση και τη διαχείριση.
- **datetime.date:** Χρησιμοποιείται για να πάρει την τρέχουσα ημερομηνία.
- **time:** Βιβλιοθήκη διαχείρισης χρόνου η οποία παρέχει συναρτήσεις διαχείρισης χρόνου αναμονής.

2.Συνάρτηση page_screenshot

```
def page_screenshot():
```

Αυτή η συνάρτηση θα εκτελέσει όλες τις ενέργειες για τη λήψη των στιγμιότυπων οθόνης.

3.Αρχειοποίηση του Playwright και του Πλοηγού

```
with sync_playwright() as playwright:
    browser = playwright.chromium.launch(headless=True)
```

- **sync_playwright():** Αρχίζει μια νέα συνεδρία του Playwright χρησιμοποιώντας τον «Διαχειριστή περιβάλλοντος» **with** εξασφαλίζοντας ότι ο Playwright θα κλείσει σωστά όταν ολοκληρωθεί η εκτέλεση του μπλοκ.
- **browser = playwright.chromium.launch(headless=True):** Εκκινεί τον περιηγητή Chromium σε "headless" λειτουργία, δηλαδή χωρίς να εμφανίζεται γραφικό περιβάλλον (ιδανικό για αυτόματες διαδικασίες).

4. Τρέχουσα Ημερομηνία

```
current_date = date.today()
```

Αποθηκεύει την τρέχουσα ημερομηνία.

5. Δημιουργία Σελίδας και Λήψη Στιγμιότυπων

Ο κώδικας περιλαμβάνει μια σειρά από βήματα που ακολουθούν την εξής λογική:

1. Δημιουργία μιας νέας σελίδας με παράμετρο ***ignore_https_errors=True***, που επιτρέπει την περιήγηση σε ιστοσελίδες με σφάλματα HTTPS, όπως ληγμένα πιστοποιητικά ή άλλες προειδοποιήσεις ασφαλείας. Αυτό είναι χρήσιμο για περιεχόμενο που μπορεί να μην είναι σωστά ασφαλές αλλά είναι απαραίτητο για τον σκοπό της αυτοματοποίησης..
2. Πλοήγηση σε διάφορες διευθύνσεις URL και αναμονή για μία συγκεκριμένη διάρκεια προτού τραβηχτεί το στιγμιότυπο οθόνης.

Για παράδειγμα, η διαδικασία λήψης στιγμιότυπου της "Ariadne":

```
page.goto("https://marinetraffic.live/vessels.php/ariadne-position/240580000/#9135262")  
time.sleep(15)  
page.screenshot(path=f"ariadne_{current_date}.png", full_page=True)
```

- **page.goto(url)**: Πλοηγεί τη σελίδα στη διεύθυνση του πλοίου.
- **time.sleep(15)**: Περιμένει 15 δευτερόλεπτα για να εξασφαλιστεί ότι η σελίδα έχει φορτωθεί πλήρως.
- **page.screenshot(...)**: Τραβάει στιγμιότυπο της σελίδας και το αποθηκεύει περιλαμβάνοντας την ημερομηνία στο όνομα του αρχείου για μεταγενέστερη χρήση. Η παράμετρος ***path*** ορίζει το όνομα του αρχείου με βάση την τρέχουσα ημερομηνία (***current_date*** που είναι μια μεταβλητή που περιέχει την ημερομηνία). Η

παράμετρος `full_page=True` υποδηλώνει ότι το στιγμιότυπο θα περιλαμβάνει ολόκληρη τη σελίδα, όχι μόνο το ορατό τμήμα της. Αυτό είναι χρήσιμο όταν η σελίδα περιέχει μεγάλα ή δυναμικά περιεχόμενα που επεκτείνονται περισσότερο από την αρχική οθόνη.

6.Κλείσιμο του Περιηγητή

```
browser.close()
```

Κλείνει τον περιηγητή μόλις ολοκληρωθούν όλες οι διαδικασίες.

7.Εκτέλεση της Συνάρτησης

```
if __name__ == "__main__":  
    page_scrot()
```

Αυτή η ενότητα ελέγχει αν το σενάριο εκτελείται ως κύριο πρόγραμμα και καλεί τη συνάρτηση `page_scrot()`.

Ο κώδικας αποτελεί μια βασική υλοποίηση αυτοματοποίησης της διαδικασίας λήψης στιγμιότυπων οθόνης για να παρακολουθεί την κατάσταση ορισμένων πλοίων από συγκεκριμένες σελίδες στην ιστοσελίδα του **MarineTraffic**, επιτρέποντας τη δημιουργία εικόνων, που μπορούν να χρησιμοποιηθούν για την καταγραφή των συμβάντων και της πραγματικής κατάστασης των πλοίων.

Υπεύθυνη Δήλωση Συγγραφέα:

Δηλώνω ρητά ότι, σύμφωνα με το άρθρο 8 του Ν.1599/1986, η παρούσα εργασία αποτελεί αποκλειστικά προϊόν προσωπικής μου εργασίας, δεν προσβάλλει κάθε μορφής δικαιώματα διανοητικής ιδιοκτησίας, προσωπικότητας και προσωπικών δεδομένων τρίτων, δεν περιέχει έργα/εισφορές τρίτων για τα οποία απαιτείται άδεια των δημιουργών/δικαιούχων και δεν είναι προϊόν μερικής ή ολικής αντιγραφής, οι πηγές δε που χρησιμοποιήθηκαν περιορίζονται στις βιβλιογραφικές αναφορές και πληρούν τους κανόνες της επιστημονικής παράθεσης.