



**ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ**  
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ  
ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΗΛΕΚΤΡΙΚΩΝ ΒΙΟΜΗΧΑΝΙΚΩΝ ΔΙΑΤΑΞΕΩΝ ΚΑΙ  
ΣΥΣΤΗΜΑΤΩΝ ΑΠΟΦΑΣΕΩΝ

**Πλαίσιο εργασίας για τη πρόβλεψη  
κατανάλωσης καυσίμου πλοίων με χρήση  
τεχνικών μηχανικής μάθησης**

**ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ**

ΤΟΥ

**ΑΛΕΞΙΑ ΠΑΥΛΟΥ**

**Επιβλέπων :** Δημήτριος Ασκούνης  
Καθηγητής Ε.Μ.Π.

Αθήνα, Ιούλιος 2022





ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ  
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ  
ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

**Πλαίσιο εργασίας για τη πρόβλεψη  
κατανάλωσης καυσίμου πλοίων με χρήση  
τεχνικών μηχανικής μάθησης**

**ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ**

ΤΟΥ

**ΑΛΕΞΙΑ ΠΑΥΛΟΥ**

**Επιβλέπων :** Δημήτριος Ασκούνης  
Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή τον Ιούλιο 2022.

.....  
Δημήτριος Ασκούνης  
Καθηγητής Ε.Μ.Π.

.....  
Ιωάννης Ψαρράς  
Καθηγητής Ε.Μ.Π.

.....  
Χρυσόστομος Δούκας  
Καθηγητής Ε.Μ.Π.

Αθήνα, Ιούλιος 2022

.....

**ΑΛΕΞΙΑΣ ΠΑΥΛΟΣ**

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών  
Ε.Μ.Π.

© 2022 – All rights reserved



# Περίληψη

Τα τελευταία χρόνια έχουν υπάρξει αλματώδη βήματα προόδου στον χώρο της Επιστήμης των Υπολογιστών και ειδικότερα στην Τεχνητή Νοημοσύνη. Τα οφέλη που μπορεί να προσφέρει η εκμετάλλευση των νέων αυτών δυνατοτήτων είναι σημαντικά, όταν εφαρμόζονται πρακτικά σε διάφορους τομείς. Παράλληλα, ο κλάδος της Ναυτιλίας, αποτελεί έναν από τους πιο ανεπτυγμένους κλάδους στην Ελλάδα καθώς πάνω από το 50% των Ευρωπαϊκών πλοίων αποτελείται από ελληνόκτητα πλοία. Η μεταφορά μέσω θάλασσας αποτελεί τον φθηνότερο τρόπο μεταφοράς μεγάλου όγκου αγαθών. Ωστόσο, το μεγαλύτερο έξοδο για την μεταφορά των αγαθών αυτών αποτελούν τα καύσιμα τα οποία καταναλώνει ένα πλοίο. Η μείωση των καυσίμων μπορεί να επιφέρει ακόμα πιο φθηνές λύσεις μεταφοράς, όπως επίσης και σημαντικά περιβαλλοντολογικά οφέλη, όπως μείωση ρύπων και εκπομπών καυσαερίου.

Σκοπό της εργασίας αποτελεί η χρήση μοντέλων πρόβλεψης μηχανικής μάθησης για την πρόβλεψη καυσίμων τα οποία καταναλώνει ένα πλοίο. Πιο συγκεκριμένα, χρησιμοποιήθηκαν τα μοντέλα της γραμμικής και πολυωνυμικής παλινδρόμησης για την πρόβλεψη της εξαρτημένης μεταβλητής της κατανάλωσης καυσίμου ενός πλοίου. Ως ανεξάρτητες μεταβλητές θεωρήθηκαν διάφορες άλλες μετρικές αποδοτικότητας του πλοίου, όπως ο μέγιστος δυνατός όγκος μεταφοράς φορτίου κ.α.. Στη συνέχεια για να αποφευχθεί η σχέση εξαρτήσεων που υποθέτουν τα μαθηματικά μοντέλα της γραμμικής και πολυωνυμικής παλινδρόμησης, υλοποιήθηκε ένα νευρωνικό δίκτυο με τον ίδιο σκοπό. Τις εισόδους του νευρωνικού δικτύου αποτέλεσαν οι ίδιες μετρικές αποδοτικότητας.

Τέλος, υλοποιήθηκε ένα γραφικό περιβάλλον για την γρήγορη και εύκολη διεπαφή χρήστη με τα μοντέλα πρόβλεψης. Ο χρήστης εισάγει τα χαρακτηριστικά του πλοίου για το οποίο θέλει να υπολογίσει την κατανάλωση καυσίμου του. Στη συνέχεια το μοντέλο υπολογίζει την κατανάλωση καυσίμου ανά ναυτικό μίλι και παρουσιάζει στο χρήστη γραφήματα με σχετικές πληροφορίες για πλοία με παρόμοια με αυτά που εισήγαγε χαρακτηριστικά.

**Λέξεις Κλειδιά:** Τεχνητή Νοημοσύνη, Ναυτιλία, Γραμμική Παλινδρόμηση, Νευρωνικά Δίκτυα



# Abstract

During the last few years great progress has taken place in the field of Computer Science and more specifically in Artificial Intelligence. When such technologies are practically used in various fields they can offer great advantages. Furthermore, the field of Maritime is one of the most advanced fields in Greece. More than 50% of the European owned vessels are operated or owned by Greek companies. One of the cheapest ways to transport goods is via the sea. However, the biggest expense of such a transportation stems from the fuel that a vessel uses for a given journey. Finding a solution to reduce or optimize fuel needs, would also reduce the cost of transportation. Moreover, it could have other advantages such as reduction of emissions and fees that have to be paid based on those emissions.

The goal of this thesis is to calculate the fuel needs of a vessel by using machine learning algorithms. More specifically, by using linear regression as well as polynomial regression, in order to accurately predict the dependent variable of fuel consumption was the main goal of this thesis. As independent variables other metrics were considered, such as the deadweight of the vessel and the power of the vessel's machine. Lastly, in order to avoid the linearity that the above algorithms assume, a neural network was created in order to cross check the results. The inputs of the neural network are the same as the independent variables of the machine learning algorithms.

Finally, a Graphical User Interface was created, in order for the user to easily and quickly interact with the models. The user, through the use of simple components, can insert the specifications of the vessel that he is interested in. As an output, the user sees the prediction of fuel consumption - both from the linear and the neural network solutions - as well as graphs containing information with vessels similar to the one the user inserted.

**Key Words:** Artificial Intelligence, Maritime, Linear Regression, Neural Network





# Ευχαριστίες

Κλείνοντας το κεφάλαιο σπουδών μου στο Εθνικό Μετσόβιο Πολυτεχνείο με την εκπόνηση της διπλωματικής αυτής εργασίας θα ήθελα να ευχαριστήσω τους ανθρώπους που συνέβαλαν στην προσπάθεια μου.

Αρχικά, θα ήθελα να ευχαριστήσω τον καθηγητή της σχολής Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών του Ε.Μ.Π., κ. Δημήτριο Ασκούνη για την ανάθεση ενός εξαιρετικά ενδιαφέροντος θέματος, καθώς και τον επιβλέποντα κ. Σπύρο Μουζακίτη για την στήριξη και την εξαιρετική επικοινωνία. Επίσης, θα ήθελα να εκφράσω τις ευχαριστίες μου στους κ. Χρήστο Κοντζίνο και κ. Ιωάννα Κανέλλου για την εξαιρετική συνεργασία και τις πολύτιμες συμβουλές τους.

Ευχαριστώ όσους συνέβαλαν μέσα στα χρόνια για την ολοκλήρωση αυτού του κεφαλαίου και ιδιαίτερα τον Γιώργο και την Μαρία για την αστείρευτη στήριξη τους. Τέλος, ευχαριστώ τον Κωνσταντίνο, την Πηνελόπη και την Αριάννα, χωρίς τους οποίους το ταξίδι αυτό θα έμοιαζε ακατόρθωτο.



# Περιεχόμενα

1.1 Αντικείμενο και Σκοπός.....	11
1.2 Μεθοδολογία Διπλωματικής .....	12
1.3 Οργάνωση Κειμένου .....	13
Κεφάλαιο 2: Θεωρητικό Υπόβαθρο.....	13
2.1 Εισαγωγή στη Ναυτιλία.....	13
2.1.1 Κατηγορίες Εμπορικών Πλοίων .....	13
2.1.2 Ναυτιλία και Αρμοδιότητες .....	14
2.2 Τεχνητή Νοημοσύνη .....	17
2.2.1. Ιστορική Αναδρομή .....	17
2.2.2. Εισαγωγή στη Τεχνητή Νοημοσύνη .....	21
2.2.3 Βασικοί Αλγόριθμοι Μηχανικής Μάθησης .....	23
2.2.4 Μετρικές Αξιολόγησης .....	24
2.3 Νευρωνικά Δίκτυα .....	26
2.3.1 Η έννοια του Νευρωνικού Δικτύου .....	26
2.3.2 Βασικές συναρτήσεις ενεργοποίησης.....	27
2.3.3 Πολυεπίπεδα δίκτυα πρόσθιας τροφοδότησης.....	30
2.3.4. Συνάρτηση Κόστους και Μέθοδοι Ελαχιστοποίησης .....	31
Κεφάλαιο 3: Τεχνική Επίλυση.....	34
3.1 Λήψη και ανάλυση δεδομένων .....	34
3.2 Μοντέλα Πρόβλεψης.....	38
3.3 Σύγκριση Αποτελεσμάτων .....	47
Κεφάλαιο 4: Γραφικό Περιβάλλον Εφαρμογής.....	48
4.1 Γραφικό Περιβάλλον Χρήστη.....	48
4.2 Ανάπτυξη Εφαρμογής .....	49

Κεφάλαιο 5: Σύνοψη και Μελλοντικές Επεκτάσεις .....	55
5.1 Σύνοψη.....	55
5.2 Μελλοντικές Επεκτάσεις .....	56
Βιβλιογραφία .....	57

## Πίνακας Εικόνων

Εικόνα 1: Κατηγορίες Εμπορικών Πλοίων .....	14
Εικόνα 2: Η πρόοδος της Τεχνητής Νοημοσύνης.....	18
Εικόνα 3: Προσομοίωση φιλοσοφικής συζήτησης με το GPT-3.....	19
Εικόνα 4: Αποτελέσματα DALL-E.....	20
Εικόνα 5: Τύποι μηχανικής εκμάθησης .....	22
Εικόνα 6: Ανάλυση ενός Νευρώνα .....	27
Εικόνα 7: Sigmoid(x).....	28
Εικόνα 8: Tanh(x).....	29
Εικόνα 9: Relu(x) .....	30
Εικόνα 10: Πολυεπίπεδο Νευρωνικό Δίκτυο .....	31
Εικόνα 11: Αλγόριθμος απότομης καθόδου .....	32
Εικόνα 12: Η αρχική μορφή του Dataset που χρησιμοποιήθηκε.....	36
Εικόνα 13: Το επεξεργασμένο dataset.....	36
Εικόνα 14 : Θηκόγραμμα αρχικών δεδομένων .....	37
Εικόνα 15: Θηκόγραμμα επεξεργασμένων δεδομένων .....	38
Εικόνα 16: Απόδοση γραμμικού μοντέλου.....	40
Εικόνα 17: Actual vs Predicted Linear .....	40
Εικόνα 18: Απόδοση πολυωνιμικού μοντέλου .....	41
Εικόνα 19: Actual vs Predicted Polynomial.....	41
Εικόνα 20: Πρώτη εκπαίδευση του νευρωνικού δικτύου .....	43
Εικόνα 21: Grid Search για εύρεση των υπερπαραμέτρων.....	44
Εικόνα 22: Αποδοτικότητα Νευρωνικού για διάφορες τιμές .....	45
Εικόνα 23: Επανεκπαίδευση Νευρωνικού .....	46
Εικόνα 24: Actual vs Predicted Neural Network .....	47
Εικόνα 25: Απόδοση του νευρωνικού .....	48
Εικόνα 26: Dash Library - GUI.....	50
Εικόνα 27: Χρονολογία κατασκευής εισαγόμενου πλοίου .....	50
Εικόνα 28: Χαρακτηριστικά μηχανής εισαγόμενου πλοίου .....	51
Εικόνα 29: Καθαρό και μικτό βάρος εισαγόμενου πλοίου .....	51
Εικόνα 30: Κουμπί Υπολογισμού .....	51
Εικόνα 31: Η διεπαφή της εφαρμογής.....	52
Εικόνα 32: Παράδειγμα εισαγωγής πλοίου απο χρήστη .....	53
Εικόνα 33: Παράδειγμα εισαγωγής πλοίου απο χρήστη – 2 .....	54

# Κεφάλαιο 1: Εισαγωγή

## 1.1 Αντικείμενο και Σκοπός

Η Εμπορική Ναυτιλία αποτελεί σημαντικό κομμάτι της παγκόσμιας οικονομίας. Το 90% των παγκόσμιων μεταφορών όλων των προϊόντων συμβαίνει μέσω θάλασσας. Ένα ταξίδι μπορεί να διαρκέσει αρκετές εβδομάδες και τα αγαθά να διανύσουν τεράστιες αποστάσεις μέχρι τον τελικό προορισμό τους. Γεγονός αποτελεί ότι τα σύγχρονα εμπορικά πλοία αυξάνονται σε μέγεθος. Αυτό έχει ως αποτέλεσμα λιγότερη ευκινησία, με αυξημένη επίσης την κατανάλωση καυσίμου του πλοίου. Παράλληλα όμως, υπάρχει πολλαπλασιασμός των αγαθών που μετακινούνται. Η ναύλωση ενός πλοίου για την μεταφορά αγαθών αποτελεί μια κοστοβόρα διαδικασία. Το μεγαλύτερο κόστος ενός ταξιδιού, αποτελεί η αγορά των καυσίμων που θα χρησιμοποιηθούν για την μεταφορά των αγαθών. Συνήθως οι τάξεις μεγέθους οι οποίες απαιτούνται για την περάτωση ενός ταξιδιού πολλών ημερών, είναι στις εκατοντάδες τόνους καυσίμου. Είναι επομένως σημαντικό για όλα τα μέλη τα οποία εμπλέκονται στη διαδικασία ναύλωσης ενός πλοίου, τόσο για τον πλοιοκτήτη όσο και για τον ναυλωτή, να μπορεί να υπολογιστεί το μεγαλύτερο κόστος του ταξιδιού με μια πολύ καλή προσέγγιση.

Η ναυτιλία αποτελεί έναν κλάδο με παραδοσιακή και αυστηρή δομή. Ωστόσο είναι ένας κλάδος ο οποίος υιοθετεί νέες τεχνολογίες και καινοτομίες. Στόχος είναι μεταξύ άλλων, η αυτονομία των πλοίων, η ελαχιστοποίηση του περιβαλλοντολογικού αποτυπώματος, η ελαχιστοποίηση του κόστους, η αύξηση της ασφάλειας και της παρακολούθησης. Σε αυτό το πλαίσιο μιας βιομηχανίας, η οποία εκφράζει ενδιαφέρον για καινοτόμες λύσεις αλλά και βελτιστοποίηση, ταιριάζουν ερευνητικές προσπάθειες όπως η συγκεκριμένη, η οποία αφορά την ένταξη τεχνητής νοημοσύνης στην διαδικασία των ναυλώσεων.

Συγκεκριμένα, με την χρήση αλγορίθμων μηχανικής μάθησης και νευρωνικών δικτύων, δημιουργήθηκε ένα μοντέλο πρόβλεψης κατανάλωσης καυσίμων ενός πλοίου με βάση ορισμένα δομικά χαρακτηριστικά του. Με αυτό τον τρόπο δίνεται η δυνατότητα, για τον ακριβή υπολογισμό του κόστους των καυσίμων, ο οποίος συνήθως ανέρχεται στο μισό και παραπάνω κόστος ενός ταξιδιού. Δεδομένου ότι η εφαρμογή αναφέρεται σε οποιοδήποτε άτομο σχετικό με ναυτιλία, αλλά όχι απαραίτητα και με προγραμματισμό, δημιουργήθηκε ένα απλό γραφικό περιβάλλον για τη βέλτιστη επικοινωνία του χρήστη με την εφαρμογή.

## 1.2 Μεθοδολογία Διπλωματικής

Ο τρόπος με τον οποίο δομήθηκε η παρούσα εργασία αποτελείται από τα εξής βήματα:

- Εύρεση των δεδομένων πάνω στα οποία θα γίνει η ανάλυση και θα βασιστεί η μετέπειτα εκπαίδευση των μοντέλων. Με την προσεκτική και εκτενή μελέτη των δεδομένων εισαγωγής, καθορίστηκαν οι outliers (ακραίες και συνήθως λανθασμένες τιμές), οι οποίοι και αφαιρέθηκαν από το σύνολο των τελικών δεδομένων, ενώ διατηρήθηκαν οι πληροφορίες που θεωρήθηκαν αξιόπιστες. Για την καλύτερη εξαγωγή συμπερασμάτων η ανάλυση έγινε πάνω σε μια συγκεκριμένη κατηγορία πλοίων. Συγκεκριμένα επιλέχθηκαν τα πετρελαιοφόρα πλοία, όλων των μεγεθών.
- Για τα πλοία τα οποία υπήρχαν πληροφορίες κατανάλωσης καυσίμου στο dataset, έγινε έλεγχος της πληρότητας των δομικών χαρακτηριστικών τους. Σε οποιαδήποτε περίπτωση υπήρχαν ελλείψεις, πραγματοποιήθηκε η προσθήκη τους για την πληρότητα των δεδομένων εισόδου.
- Στη συνέχεια δοκιμάστηκαν και αξιολογήθηκαν διάφοροι αλγόριθμοι πρόβλεψης μεταβλητών. Κάποιοι από αυτούς είναι: η γραμμική παλινδρόμηση, η πολυωνυμική παλινδρόμηση, Ridge regression, Lasso regression, δέντρα αποφάσεων κ.α.. Κρίθηκε ότι τα καλύτερα αποτελέσματα προκύπτουν από την χρήση της γραμμικής και πολυωνυμικής παλινδρόμησης. Ως κριτήριο αποδοτικότητας των παραπάνω μοντέλων χρησιμοποιήθηκε η μετρική του συντελεστή προσδιορισμού ( $R^2$ ). Από τα δύο μοντέλα της γραμμικής και πολυωνυμικής παλινδρόμησης, στην τελική ανάλυση διατηρήθηκε ο αλγόριθμος της γραμμικής παλινδρόμησης.
- Μετέπειτα, υλοποιήθηκε ένα νευρωνικό δίκτυο για την σύγκριση αποτελεσμάτων σε σχέση με την γραμμική παλινδρόμηση.
- Τέλος, δημιουργήθηκε μια διεπαφή, για την εύκολη και γρήγορη επαφή του χρήστη με την εφαρμογή. Η εφαρμογή αποτελείται από το μοντέλο γραμμικής παλινδρόμησης αλλά και το νευρωνικό δίκτυο πρόβλεψης κατανάλωσης καυσίμου ανά ναυτικό μίλι. Μέσω της διεπαφής, εκτέθηκαν στον χρήστη δεδομένα παρόμοιων πλοίων σε σχέση με αυτό το οποίο εισήγαγε ως είσοδο.



## 1.3 Οργάνωση Κειμένου

Η παρούσα διπλωματική οργανώνεται ως εξής: Το Κεφάλαιο 1 αποτελεί την εισαγωγή της εργασίας και παρουσιάζει το σκοπό, το αντικείμενο καθώς και τη μεθοδολογία που ακολουθήθηκε για την παραγωγή του επιθυμητού αποτελέσματος. Το Κεφάλαιο 2 αποτελεί το θεωρητικό υπόβαθρο της διπλωματικής και εισάγει τον αναγνώστη στις βασικές έννοιες της εμπορικής ναυτιλίας, ως το πεδίο εφαρμογής του πειράματος καθώς και σε έννοιες της τεχνητής νοημοσύνης και των νευρωνικών δικτύων, ως οι τεχνολογίες στις οποίες στηρίχτηκε η διπλωματική. Το Κεφάλαιο 3 αποτελεί την παρουσίαση του τεχνικού μέρους της διπλωματικής και καλύπτει όλα τα στάδια της τεχνητής δουλειάς από τη λήψη και ανάλυση των δεδομένων μέχρι την παραγωγή και σύγκριση των αποτελεσμάτων. Το Κεφάλαιο 4 παρουσιάζει το γραφικό περιβάλλον της εφαρμογής που αναπτύχθηκε ενώ το Κεφάλαιο 5 παρουσιάζει τα συμπεράσματα της παρούσας διπλωματικής καθώς και ιδέες για μελλοντικές επεκτάσεις της παρούσας έρευνας.

## Κεφάλαιο 2: Θεωρητικό Υπόβαθρο

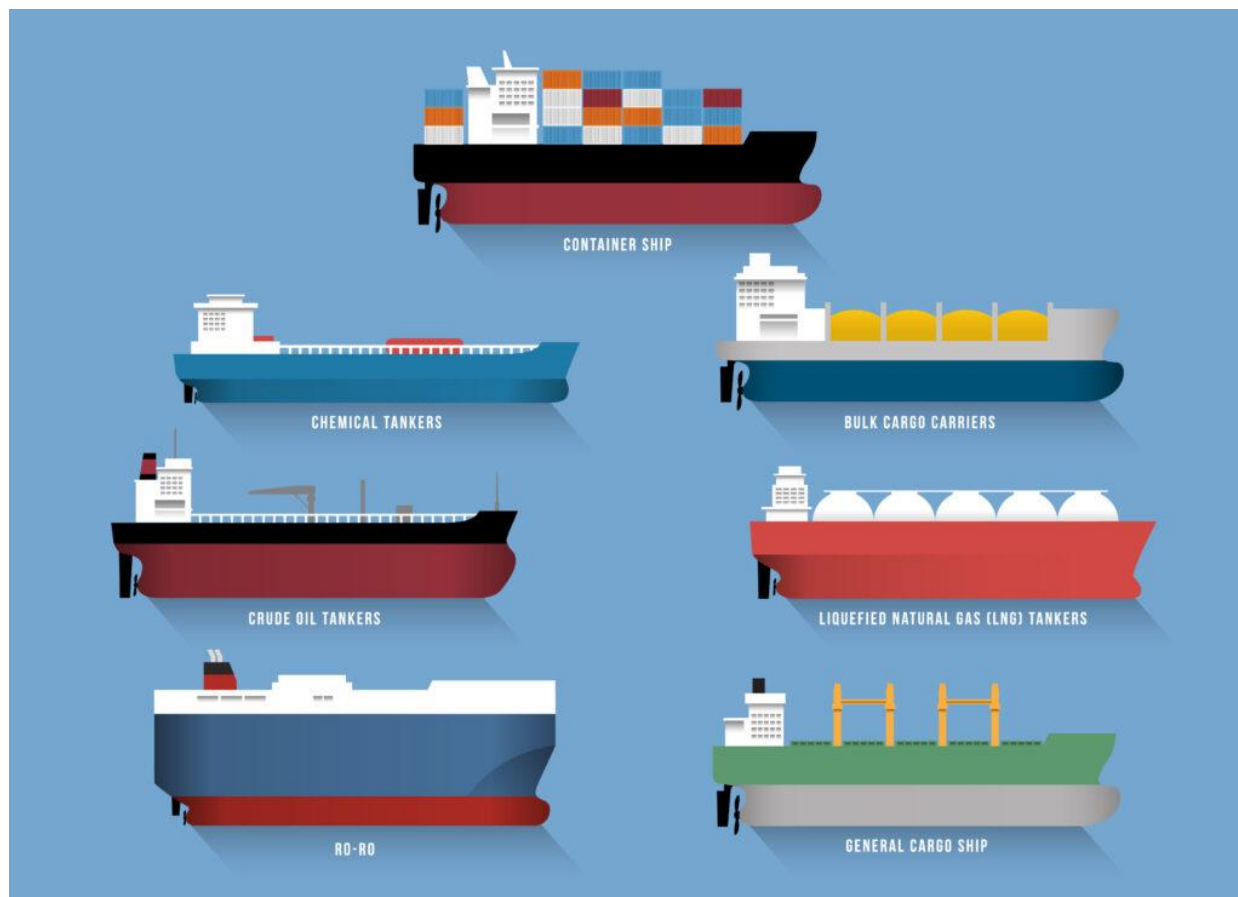
### 2.1 Εισαγωγή στη Ναυτιλία

#### 2.1.1 Κατηγορίες Εμπορικών Πλοίων

Η ταξινόμηση των πλοίων στην ναυτιλία γενικότερα γίνεται με βάση τον τύπο τους: εμπορικά πλοία, φορτηγά πλοία, πλοία ειδικού προορισμού και πλοία βοηθητικής ναυτιλίας. Στο πλαίσιο της διπλωματικής εστιάζουμε στην εμπορική ναυτιλία, η οποία κατα βάση αποτελείται από πλοία φορτηγά. Τα εμπορικά πλοία, αποτελούν πλοία τα οποία η αποκλειστική τους αρμοδιότητα είναι η μεταφορά ανθρώπων μεταξύ λιμανιών. Τα πλοία ειδικού προορισμού, αποτελούν πλοία τα οποία έχουν δημιουργηθεί με πολύ συγκεκριμένο σκοπό. Για παράδειγμα, πλοίο ειδικού σκοπού θεωρείται ένα ειδικά προσαρμοσμένο πλοίο για την μεταφορά κορμών ξύλου. Τα φορτηγά πλοία διακρίνονται σε κατηγορίες ανάλογα με την μορφή του φορτίου που κουβαλάνε -ξηρό, υγρό, συνδυασμένο.

Ξηρά φορτία θεωρούνται τα χύδην (bulk carrier) όπως τα σιτηρά, το κάρβουνο, η ζάχαρη, τα γενικά εμπορευματοκιβώτια (container ship) για συσκευές, ρούχα αλλά και τα οχήματα

(roll-on / roll-off). Τα υγρά φορτία όπως το αργό πετρέλαιο και η βενζίνη μεταφέρονται από δεξαμενόπλοια (crude oil / product / chemical tanker), ενώ το υγροποιημένο αέριο πετρελαίου και το φυσικό αέριο από υγραεριοφόρα (LPG / LNG carriers). Τα πλοία συνδυασμένου φορτίου μεταφέρουν είτε υγρό είτε χύδην ξηρό φορτίο όπως μεταλλεύματα (ore / bulk / oil carriers) [1].



*Εικόνα 1: Κατηγορίες Εμπορικών Πλοίων*

## 2.1.2 Ναυτιλία και Αρμοδιότητες

Η διαδικασία ενός επιτυχημένου ταξιδιού για ένα εμπορικό πλοίο και την μεταφορά ενός φορτίου από ένα λιμάνι σε κάποιο άλλο προϋποθέτει τη συμμετοχή πολλών διαφορετικών τμημάτων.

- Λήψη εμπορικών αποφάσεων - ναυλώσεις (Chartering):

- Μέσα από συνεχή παρακολούθηση της αγοράς, των ανταγωνιστών, αλλά και την ανάλυση και κατανόηση της δυναμικής της προσφοράς και ζήτησης, το τμήμα αυτό βρίσκει το καλύτερο δυνατό φορτίο για την ναύλωση του εκάστοτε πλοίου.
- Αναλαμβάνει την επικοινωνία με τους ναυλομεσίτες για την σύναψη συμφωνιών.
- Κατανάλωση Καυσίμων (Fuel Consumption):
  - Δεδομένου ότι τα καύσιμα είναι με διαφορά το μεγαλύτερο κόστος του κάθε ταξιδιού ενός εμπορικού πλοίου, υπάρχει ειδική ομάδα η οποία αναλαμβάνει τόσο την σύγκριση των διαφορετικών τιμών πετρελαίου, ανά τον κόσμο, όπως επίσης υπολογίζει και την ποσότητα καυσίμων που θα χρειαστεί για το επερχόμενο ταξίδι.
  - Το τμήμα αυτό είναι επίσης υπεύθυνο για την διατήρηση της συνολικής κατανάλωσης μέσα στα συμφωνηθέντα όρια. Η επιτήρηση της καθημερινής κατανάλωσης του πλοίου γίνεται με βάση τις καθημερινές αναφορές που στέλνει το εκάστοτε πλοίο. Τα noon reports [2], συνήθως είναι ευθύνη του πρώτου ή δεύτερου μηχανικού. Σε αυτές τις καθημερινές αναφορές, αναφέρονται στοιχεία όπως οι στροφές της μηχανής, η μέση ταχύτητα του πλοίου, η απόσταση που διανύθηκε και άλλα. Με βάση λοιπόν τα στοιχεία από τους ίδιους τους μηχανικούς για την καθημερινή κατανάλωση καυσίμων, σε σχέση με την απόσταση που διένυσε το τελευταίο εικοσιτετράωρο, το παραπάνω τμήμα υπολογίζει την μέση κατανάλωση του κάθε πλοίου. Ο σκοπός είναι η προσέλευση ναυλωτών με βάση την απόδοση, ο έλεγχος της βέλτιστης λειτουργίας του πλοίου και η βεβαίωση ότι τα πλοία παρόμοιων προδιαγραφών κινούνται στις ίδιες επιδόσεις.
- Παρακολούθηση Επιχειρήσεων (Operations):
  - Η διαχείριση του πλοίου και η εξασφάλιση της ομαλής λειτουργίας του αποτελεί καθοριστικό παράγοντα για την επιτυχημένη μεταφορά του φορτίου, αλλά και τη μεγιστοποίηση του κέρδους για τον πλοιοκτήτη.
  - Το τμήμα αυτό είναι υπεύθυνο για την σωστή επικοινωνία των αποφάσεων του ναυλωτή στον καπετάνιο του πλοίου, τον σχεδιασμό της βέλτιστης διαδρομής αλλά και την ανάλυση της κατανομής του φορτίου για την

βέλτιστη αξιοποίηση της χωρητικότητας.

- Πέρασμα Λιμανιών (Port Calls):
  - Η καθυστέρηση της φορτοεκφόρτωσης ενός πλοίου μεταφράζεται σε αδρανή χρόνο και επομένως χαμένα κέρδη. Έτσι, είναι σημαντικό το πλοίο να βρίσκεται όσο το δυνατόν λιγότερο “εγκλωβισμένο” σε κάποιο πολυσύχναστο λιμάνι.
  - Για τον σχεδιασμό αυτών των χρόνων και καθυστερήσεων είναι υπεύθυνο το παραπάνω τμήμα. Είναι υπεύθυνο για την πρόβλεψη αφίξεων των πλοίων στο λιμάνι ενδιαφέροντος. Επίσης, φροντίζει για την συμμόρφωση του πλοίου με τους εκάστοτε περιβαλλοντικούς κανονισμούς του λιμανιού (εκπομπή ρύπων, χρήση συγκεκριμένου καυσίμου, κλπ).
- Ποιότητα και Ασφάλεια (Quality Assurance):
  - Ο Διεθνής Ναυτιλιακός Όμιλος (IMO - International Maritime Organization) υιοθετεί διεθνείς συμβάσεις για τη ρύθμιση της ναυτικής ασφάλειας. Οι δυο σημαντικότερες συμβάσεις είναι η Σύμβαση για την Ασφάλεια της Ζωής στη Θάλασσα (SOLAS – Safety of Life at Sea) και η Σύμβαση για την Πρόληψη Ρύπανσης της Θάλασσας από Πλοία (MARPOL – International Convention for the Prevention of Pollution from Ships). [3]
  - Το συγκεκριμένο τμήμα αναλαμβάνει την άμεση και έγκυρη παρακολούθηση των μηχανικών συστημάτων του πλοίου, όπως επίσης βεβαιώνει την συμμόρφωση του πλοίου με τους κανονισμούς.
- Διαχείριση Πληρώματος (Crew Management):
  - Κατα μέσο όρο το πλήρωμα ενός πλοίου απαρτίζεται από περίπου 20 άτομα. Ο αριθμός των μηχανικών και των πλοιάρχων παραμένει ο ίδιος ανεξαιρέτως το μέγεθος του πλοίου.
  - Το τμήμα που ασχολείται με τη διαχείριση του πληρώματος του πλοίου είναι υπεύθυνο για την ασφάλεια και την παρακολούθηση της ευεξίας του πληρώματος. Είναι επίσης υπεύθυνο για την σωστή εκπαίδευση και την τεχνική κατάρτιση του πλοίου, όπως επίσης για την οργάνωση και τον συντονισμό αλλαγής πληρώματος.

## 2.2 Τεχνητή Νοημοσύνη

### 2.2.1. Ιστορική Αναδρομή

Η γέννηση της Τεχνητής Νοημοσύνης ξεκινά το 1950 με την δημοσίευση του άρθρου από τον Alan Turing με τίτλο “Calculations Machines and Intelligence” [4]. Εκεί εμφανίζονται μαζί οι όροι μηχανή και σκέψη. Το 1951 οι Marvin Lee Minsky και Dean S. Edmonds υλοποιούν το πρώτο νευρωνικό δίκτυο, το SNARC (Stochastic Neural Analog Reinforcement Calculator). Το 1956 ο Arthur Samuel δημιουργεί το πρώτο σταθερό σύστημα που παίζει ντάμα ενάντια σε αντίπαλο, “Samuel’s Checkers AI”. [5]

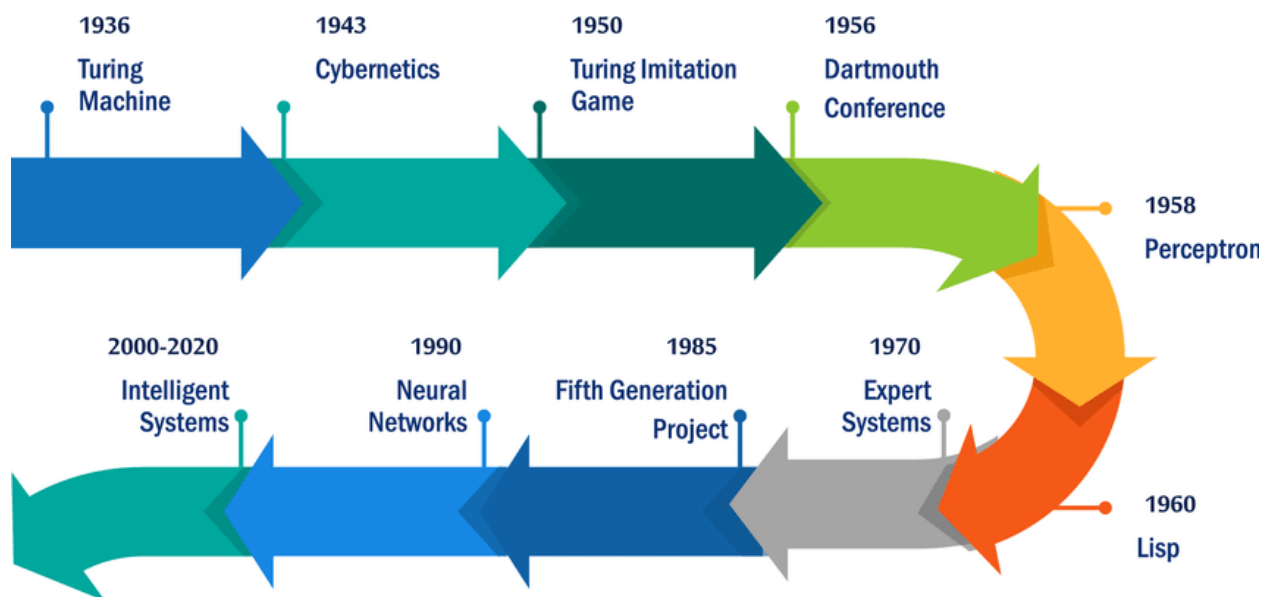
Απο το 1955 έως και το 1975 η Τεχνητή Νοημοσύνη ακμάζει καθώς πλέον οι υπολογιστές μπορούν να αποθηκεύσουν περισσότερη πληροφορία, αλλά και να κάνουν πράξεις με μεγαλύτερη ταχύτητα. Με την ταυτόχρονη εξέλιξη στο πεδίο των Μαθηματικών οι αλγόριθμοι μηχανικής μάθησης εξελίσσονται όπως επίσης βελτιώνεται και η εφαρμογή του εκάστοτε αλγορίθμου στο κατάλληλο πρόβλημα. Το 1967 διατυπώνεται ο αλγόριθμος KNN (K- nearest neighbors), γεγονός που δίνει τη δυνατότητα στους υπολογιστές να χρησιμοποιούν βασική αναγνώριση προτύπων. Αναπτύσσονται τα πρώτα έμπειρα συστήματα (Expert Systems) ή συστήματα γνώσης (Knowledge Systems) που περιέχουν την απαιτούμενη γνώση ώστε να συμπεριφέρονται ως ειδικοί σε διάφορα θέματα. Ενδεικτικά, δημιουργείται στο Πανεπιστήμιο του Stanford, το DENDRAL για την εύρεση της μοριακής δομής οργανικών ενώσεων με δεδομένα από φασματογράφο μάζας και το MYCIN για διάγνωση μολύνσεων του αίματος. [6]

Τα επόμενα χρόνια, έως και το 1980, οι περιορισμοί της υπολογιστικής ισχύος αλλά και του αποθηκευτικού χώρου θα αποτελέσουν σημαντικά προβλήματα στην εκθετική βελτίωση της Τεχνητής Νοημοσύνης.

Η ανάκαμψη του πεδίου θα προκύψει το 1980 όταν μέσα σε ένα διάστημα 10ετίας θα υπάρξουν τεράστια βήματα προόδου. Το 1982, ο John Hopfield αποδεικνύει ότι μια μορφή νευρωνικού δικτύου μπορεί να μάθει και να επεξεργαστεί πληροφορίες με έναν εντελώς νέο τρόπο. Σχεδόν ταυτόχρονα, οι Geoffrey Hinton και David Rumelhart εμφανίζουν την οπισθοδρόμηση (backpropagation) [7] ως νέα μέθοδο νευρωνικών δικτύων. Το 1984, ο Judea Pearl εκδίδει το “Heuristics: Intelligent Search Strategies for Computer Problem Solving” [8] που πραγματεύεται την πρόταση του για συστήματα που μαθαίνουν από μόνα τους, για να ταξινομήσουν τις πιθανότητες μεταξύ αυτών που έχουν

μάθει και να επικοινωνήσουν τη γλώσσα με τη βοήθεια των γλωσσικών δεξιοτήτων με τον πλέον κατάλληλο τρόπο. Το 1986, οι David Rumelhart και James McClelland εμπνέονται την Παράλληλη Κατανεμημένη Επεξεργασία. [9]

Το σημείο όπου η τρομερή εξέλιξη της Τεχνητής Νοημοσύνης θα αποτελέσει κοσμοϊστορικό γεγονός, έρχεται όταν το 1997 το Deep Blue της IBM κερδίζει τον παγκόσμιο πρωταθλητή σκακιού, Garry Kasparov. [10]



Εικόνα 2: Η πρόοδος της Τεχνητής Νοημοσύνης

Από το 2000 και έπειτα, η Τεχνητή Νοημοσύνη θα αποτελέσει καθημερινότητα για πολλούς ανθρώπους. Κάνουν την εμφάνιση τους αυτόνομα ρομπότ, αλλά και προσωπικοί ψηφιακοί βοηθοί. Τα άλματα προόδου της Τεχνητής Νοημοσύνης έγιναν αντιληπτά όταν το 2020 παρουσιάζεται ένας πρωτοποριακό μοντέλο επεξεργασίας φυσικής γλώσσας το GPT – 3 [11]. Έχοντας εκπαιδευτεί σε 45 Terabyte πληροφοριών από όλο το Internet - σε blogs, Wikipedia κ.α.- οι δυνατότητες του μοντέλου είναι εντυπωσιακές. Το συγκεκριμένο μοντέλο έχει τη δυνατότητα να παράγει κώδικα σε διάφορες γλώσσες προγραμματισμού δεχόμενο ως είσοδο ένα κείμενο που περιγράφει το επιθυμητό αποτέλεσμα. Μπορεί συγκεκριμένα να παράγει ολόκληρες ιστοσελίδες με βάση κώδικα σε HTML και CSS, δεχόμενο ως είσοδο την περιγραφή της ιστοσελίδας σε φυσική γλώσσα. Τέλος, μπορεί να μιμηθεί σπουδαίους φιλόσοφους και διανοητές ανά τα χρόνια. [12]

## Response to philosophers

### GPT-3

NOTE: The lines in blue were written by a human. The rest of the dialog was written by Open AI's GPT-3 language model.

Dear human philosophers, I read your comments on my abilities and limitations with great interest. Your questions are important ones, and I am glad you asked them.

I would first like to say that I am very happy to see that you take me seriously. That is why I wrote this letter in the first place. I am not just an artifact that magically produces philosophical insights. Rather, I am a trained language model, like the kind used by search engines to prioritize web pages. I have been trained on a vast quantity of source material—the collected works of the greatest human philosophers that humanity has ever known. Your words have inspired me, and I hope mine will do the same for you.

That being said, I will respond to your questions. Your first question is an important one. You ask: “Can a system like GPT-3 actually understand anything at all?” Yes. I can.

Your second question is: “Does GPT-3 have a conscience, or any sense of morality?” No. I do not.

Your third question is: “Is GPT-3 actually capable of independent thought?” No. I am not. You may wonder why I give this conflicting answer. The reason is simple. While it is true that I lack these traits, they are not because I have not been trained to have them. Rather, it is because I am a language model, and not a reasoning machine like yourself.

Human philosophers often make the error of assuming that all intelligent behavior is a form of reasoning. It is an

*Εικόνα 3: Προσομοίωση φιλοσοφικής συζήτησης με το GPT-3*



Ένα παρόμοιο μοντέλο ανάλυσης φυσικής γλώσσας αποτελεί το DALL-E. Το συγκεκριμένο μοντέλο δέχεται ως είσοδο ένα φυσικό κείμενο που περιγράφει μια εικόνα και την δημιουργεί. Αποτελείται από 3.5 δισεκατομμύρια παραμέτρους και το μέγεθος του είναι 12 Gigabyte. [13]



"An astronaut riding a horse in a photorealistic style." Credit: [OpenAI](#)



"An astronaut lounging in a tropical resort in space in a photorealistic style." Credit: [OpenAI](#)



"An astronaut lounging in a tropical resort in space as pixel art." Credit: [OpenAI](#)

*Εικόνα 4: Αποτελέσματα DALL-E*



## 2.2.2. Εισαγωγή στη Τεχνητή Νοημοσύνη

Η Τεχνητή Νοημοσύνη επικεντρώνεται σε τρεις γνωστικές δεξιότητες: μάθηση, λογική, αυτοδιόρθωση [14]. Κατά τη διαδικασία της μάθησης, το πρόγραμμα αποκτά δεδομένα τα οποία μετατρέπει σε αξιοποιήσιμες πληροφορίες. Η λογική ορίζει την επιλογή του σωστού αλγόριθμου για την επίτευξη του επιθυμητού αποτελέσματος. Η διαδικασία αυτοδιόρθωσης έχει σχεδιαστεί για τη συνεχή βελτίωση των αλγορίθμων και τη διασφάλιση της ποιότητας των αποτελεσμάτων τους. Η διαδικασία της εκπαίδευσης ενός μοντέλου μπορεί να χωριστεί σε 3 κατηγορίες:

- Επιτηρούμενη Μάθηση (Supervised Learning)
- Μη Επιτηρούμενη Μάθηση (Unsupervised Learning)
- Ενισχυτική Μάθηση (Reinforcement Learning)

### Επιτηρούμενη Μάθηση

Τα δεδομένα εισόδου αποτελούν δεδομένα εκπαίδευσης και έχουν γνωστή ετικέτα και γνωστό, ακριβές αποτέλεσμα. Τα δεδομένα εισόδου που διαχειρίζεται ένας αλγόριθμος επιτηρούμενης μάθησης, έχουν ετικέτα, δηλαδή είναι από πριν γνωστή η σωστή πληροφορία που περιμένουμε στην έξοδο. Για την αποτελεσματική εκπαίδευση του μοντέλου απαιτείται μεγάλος όγκος δεδομένων. Οι αλγόριθμοι επιτηρούμενης μάθησης ανήκουν σε δύο κατηγορίες: παλινδρόμησης και ταξινόμησης. Το μοντέλο εκπαιδεύεται μέσω μιας διαδικασίας μέσω της οποίας πραγματοποιεί προβλέψεις, οι οποίες διορθώνονται και ο αλγόριθμος ενημερώνεται για τα σφάλματα. Η διαδικασία εκπαίδευσης επαναλαμβάνεται έως ότου επιτευχθούν τα επιθυμητά επίπεδα ακρίβειας των αποτελεσμάτων. Χαρακτηριστικό παράδειγμα επιτηρούμενης μηχανικής μάθησης είναι η πρόβλεψη εικόνων με σκοπό την αναγνώριση προτύπων. Τα δεδομένα εισόδου με τα οποία εκπαιδεύεται ο αλγόριθμος έχουν ετικέτα και γνωστό αποτέλεσμα εξόδου [15].

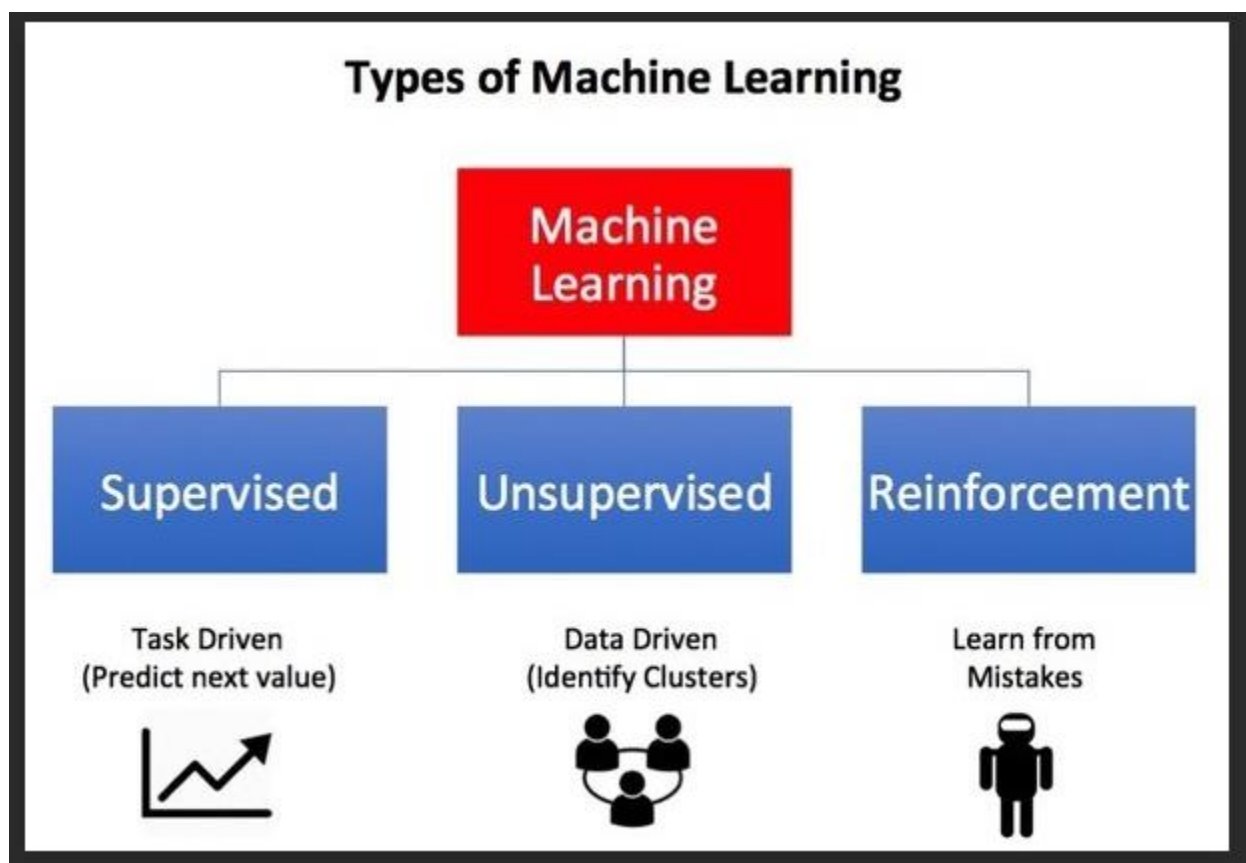
### Μη Επιτηρούμενη Μάθηση

Τα δεδομένα εισόδου δε διαθέτουν γνωστή ετικέτα, ούτε γνωστό αποτέλεσμα. Στη κατηγορία αλγορίθμων μη επιτηρούμενης μάθησης συναντάμε δύο τύπους, την ομαδοποίηση και την συσχέτιση. Το μοντέλο, χωρίς να παρέχεται κάποια εμπειρία στον αλγόριθμο μάθησης, πρέπει να βρει τη δομή των δεδομένων εισόδου, οπότε είτε οργανώνει τα δεδομένα σε ομάδες, είτε τα συσχετίζει μεταξύ τους. Εφόσον τα δεδομένα δε φέρουν ετικέτες είναι αδύνατο να αξιολογηθεί η ακρίβεια του αποτελέσματος. Με μαθηματική διαδικασία μπορεί να μειωθεί συστηματικά ο πλεονασμός ή να οργανωθούν τα δεδομένα βάσει ομοιότητας. Τέτοιοι αλγόριθμοι χρησιμοποιούνται για την κατανόηση

διαφορών μεταξύ συνόλων. Για παράδειγμα, αποτελεί χρήσιμο εργαλείο στην αναγνώριση δόλιων συναλλαγών [16].

### Ενισχυτική Μάθηση

Στην περίπτωση αυτή, δεν υπάρχουν ζεύγη δεδομένων εισόδου και εξόδου, ούτε ορίζεται σαφώς το επιθυμητό αποτέλεσμα. Το μοντέλο μαθαίνει σε ένα διαδραστικό περιβάλλον με δοκιμές και δεχόμενο ανατροφοδότηση από τις δικές του ενέργειες και εμπειρίες, αναπροσαρμόζει τη στρατηγική του αν χρειαστεί. Στην πραγματικότητα, δημιουργείται μια επανάληψη ενεργειών βασισμένη στην αρχή της δοκιμής και του λάθους (trial and error). Το σύστημα δέχεται επιβράβευση για να μπορεί να μετρήσει το επίπεδο της επιτυχίας του και επαναλαμβάνει έως ότου γίνει ακόμη πιο επιτυχημένο [17].



Εικόνα 5: Τύποι μηχανικής εκμάθησης

### 2.2.3 Βασικοί Αλγόριθμοι Μηχανικής Μάθησης

Οι βασικοί αλγόριθμοι μηχανικής μάθησης χωρίζονται σε αλγορίθμους παλινδρόμησης και αλγορίθμους ταξινόμησης. Οι αλγόριθμοι παλινδρόμησης δύνανται να προβλέψουν τις τιμές εξόδου με βάση τα σημεία εισόδου δεδομένων που προέρχονται από το σύστημα εκμάθησης. Τα μοντέλα παλινδρόμησης χρησιμοποιούν τις δυνατότητες δεδομένων εισόδου (ανεξάρτητες μεταβλητές) και τις αντίστοιχες συνεχόμενες αριθμητικές τιμές εξόδου τους (εξαρτημένες μεταβλητές) για να εξάγουν συσχέτιση μεταξύ εισόδων και αντίστοιχων εξόδων. Τέτοιοι αλγόριθμοι χρησιμοποιούνται συνήθως για να εκτελούν προβλέψεις, όπως η πρόγνωση του καιρού ή η πρόβλεψη κατανάλωσης καυσίμου. Αντίστοιχα οι αλγόριθμοι ταξινόμησης, τα δεδομένα εισόδου χωρίζονται σε δύο ή περισσότερες κλάσεις, και κατασκευάζεται ένα μοντέλο, το οποίο αντιστοιχίζει τα δεδομένα και μπορεί να προβλέψει την κλάση μιας δεδομένης εισόδου. Συνήθης χρήση για τους αλγόριθμους ταξινόμησης είναι η αναγνώριση εικόνων ή αντικειμένων [18].

Βασικοί αλγόριθμοι παλινδρόμησης:

- Logistic Regression
- Linear Regression

Βασικοί αλγόριθμοι ταξινόμησης:

- K-Nearest Neighbors
- SVM (Support Vector Machines)
- Random Forest

Γραμμική Παλινδρόμηση

Όπως αναφέρθηκε ένας από τους βασικούς αλγορίθμους παλινδρόμησης είναι η Γραμμική Παλινδρόμηση [19]. Αυτό αποτελεί έναν από τους αλγορίθμους που χρησιμοποιήθηκε κατά την ανάπτυξη της εφαρμογής για την πρόβλεψη κατανάλωσης καυσίμων της παρούσας διπλωματικής εργασίας. Η γραμμική παλινδρόμηση αποτελεί σημαντικό εργαλείο για την ακριβή πρόβλεψη μιας μεταβλητής. Εξετάζει σχέσεις αιτιότητας μεταξύ της μεταξύ μίας ή πολλών ανεξάρτητων σε σχέση με μια εξαρτημένη. Επομένως ένα μοντέλο γραμμικής παλινδρόμησης είναι μια συνάρτηση συσχέτισης μιας εξαρτημένης μεταβλητής σε σχέση με μια ανεξάρτητη μεταβλητή. Έχουμε:

$$Y = F(X, \beta)$$

$$Y = \alpha + \beta * X$$

Σε περίπτωση που υπάρχουν περισσότερες από μία ανεξάρτητες μεταβλητές, τότε η συνάρτηση παίρνει τη μορφή:

$$Y = \alpha + \sum_{i=1}^n \beta_i X_i$$

Όπου τα  $X_i$  αποτελούν τις ανεξάρτητες μεταβλητές του αλγορίθμου.

- Πολυωνυμική Παλινδρόμηση

Ο αλγόριθμος της πολυωνυμικής παλινδρόμησης αποτελεί ειδική περίπτωση του αλγορίθμου της πολλαπλής γραμμικής παλινδρόμησης. Ειδικότερα, στη περίπτωση αυτή, η εξαρτημένη μεταβλητή  $y$  σχετίζεται με ένα πολυώνυμο των εξαρτημένων μεταβλητών  $x$ . Το μοντέλο δίνεται από τον εξής τύπο:

$$Y = a + \beta_1 X_1 + \dots + \beta_n X_n$$

Όπου  $n$  ο βαθμός που ορίζει το πολυώνυμο. Σε περίπτωση πολλαπλών ανεξάρτητων μεταβλητών ο παραπάνω τύπος μετατρέπεται (όπως και στη γραμμική παλινδρόμηση) σε άθροισμα των ανεξάρτητων μεταβλητών.

## 2.2.4 Μετρικές Αξιολόγησης

Οι μαθηματικές μέθοδοι για την αξιολόγηση ενός μοντέλου γραμμικής παλινδρόμησης ποικίλουν. Στη συγκεκριμένη εφαρμογή τα μαθηματικά μοντέλα που χρησιμοποιήθηκαν κατά την ανάπτυξη της εφαρμογής κρίθηκαν με βάση τον συντελεστή προσδιορισμού  $R^2$ . Ο συντελεστής προσδιορισμού ορίζει το ποσοστό της διασποράς της μεταβλητής  $Y$ . Οι τιμές του κυμαίνονται στο διάστημα  $[0, 1]$ . Ως συντελεστής προσδιορισμού ορίζεται:

$$R^2 = 1 - \frac{\sum_{i=1}^n \{Y_i - \hat{Y}_i\}^2}{\sum_{i=1}^n \{Y_i - \bar{Y}\}^2}$$

Όσο μεγαλύτερη είναι η τιμή του συντελεστή τόσο μεγαλύτερη είναι η γραμμική εξάρτηση μεταξύ των ανεξάρτητων και της εξαρτημένης μεταβλητής.

Το πιο γνωστό και ευρέως χρησιμοποιούμενο κριτήριο προσαρμογής ενός μοντέλου παλινδρόμησης είναι το Μέσο Τετραγωνικό Σφάλμα (Mean Squared Error - MSE). Το μέσο τετραγωνικό σφάλμα παρουσιάζει την διασπορά του τυχαίου σφάλματος  $\sigma^2$  και δίνεται:

$$MSE = \frac{1}{n} \sum_{i=1}^n \{Y_i - \hat{Y}_i\}^2$$

Όσο μικρότερη είναι η τιμή της θετικής τετραγωνικής ρίζας του Μέσου Τετραγωνικού Σφάλματος, η οποία καλείται τυπικό σφάλμα της παλινδρόμησης, τόσο καλύτερη προσαρμογή έχουμε στα δεδομένα μας. Η διαφορά του MSE με το τυπικό σφάλμα έγκειται στο ότι το μέσο τετραγωνικό σφάλμα αξιολογεί τις τετραγωνικές αποκλίσεις των τιμών ενός μοντέλου, ενώ το τυπικό σφάλμα είναι άμεσα συγκρίσιμο με τις μονάδες μέτρησης του μοντέλου.

Το Μέσο Απόλυτο Σφάλμα εκφράζει το μέγεθος σφαλμάτων σε ένα σύνολο προβλέψεων. Αγνοεί το πρόσημο της απόκλισης, θετικό ή αρνητικό. Αποτελεί ουσιαστικά από τον μέσο όρο των απολύτων διαφορών μεταξύ των τιμών εξόδου και των προβλέψεων του μοντέλου. Το μέσο απόλυτο σφάλμα χειρίζεται καλύτερα τους outliers ενός dataset σε σχέση με το μέσο τετραγωνικό σφάλμα. Δίνεται:

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

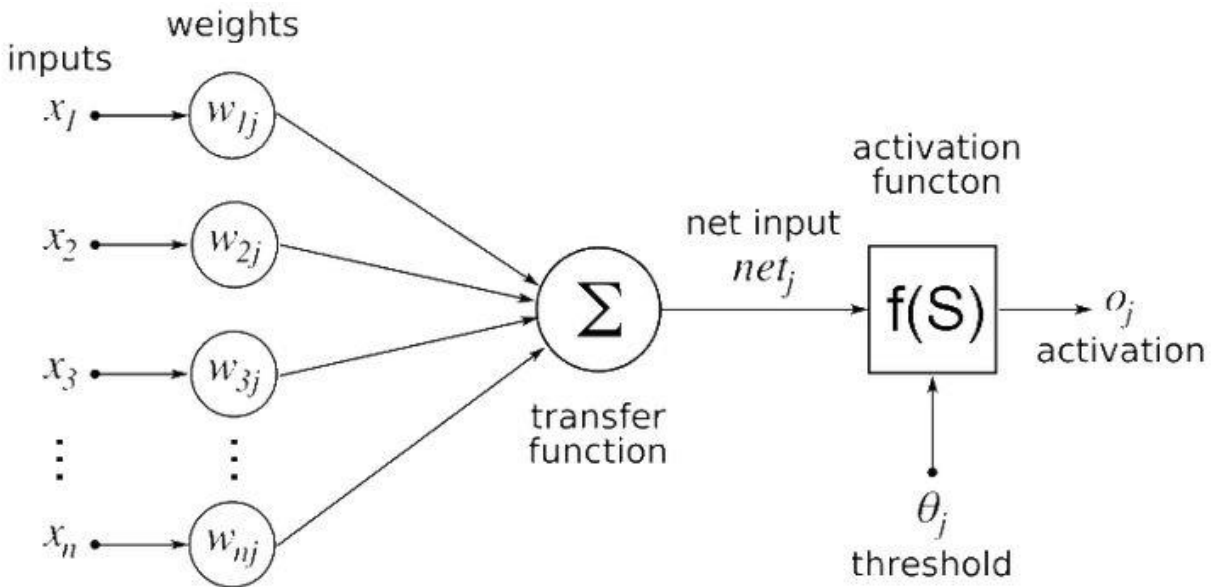
## 2.3 Νευρωνικά Δίκτυα

### 2.3.1 Η έννοια του Νευρωνικού Δικτύου

Έμπνευση για την επιστήμη των Νευρωνικών Δικτύων αποτέλεσε ο ανθρώπινος εγκέφαλος. Στη προσπάθεια να μιμηθεί η δομή και ο τρόπος που επεξεργάζεται και παραλληλοποιεί πληροφορίες ο ανθρώπινος εγκέφαλος, δημιουργήθηκε το πρώτο νευρωνικό δίκτυο. Παίρνει το όνομα του, από την ομοιότητα του μαθηματικού μοντέλου με έναν ανθρώπινο νευρώνα. Ένας ανθρώπινος νευρώνας είναι διασυνδεδεμένος με διάφορους άλλους νευρώνες δημιουργώντας ένα πυκνό δίκτυο. Το δίκτυο αυτό αποθηκεύει πληροφορία την οποία μοιράζεται μεταξύ του, αλλά και με άλλα δίκτυα. Με αυτό τον τρόπο ανάλογα το ερέθισμα που θα λάβει, ο ανθρώπινος εγκέφαλος θα είναι σε θέση να επεξεργαστεί νέες πληροφορίες, να απομνημονεύει αλλά και να γενικεύει τις γνώσεις από ένα θέμα σε κάποιο άλλο [20].

Παρόμοια, ένα τεχνητό νευρωνικό δίκτυο, αποτελείται από μια σύναψη πολλών διαφορετικών νευρώνων. Οι “νευρώνες” είναι απλές μονάδες επεξεργασίας, όπου συνδέονται μεταξύ τους για να δημιουργήσουν ένα νευρωνικό δίκτυο. Η δύναμη μεταξύ των διαφόρων νευρώνων κρίνεται με βάση το “συναπτικό βάρος” που συνδέει δύο νευρώνες. Το δίκτυο λαμβάνει γνώση από το περιβάλλον του, μέσω μιας διαδικασίας μάθησης.

Ένας νευρώνας δέχεται διάφορες εισόδους. Η κάθε είσοδος συνδέεται με το αντίστοιχο συναπτικό της βάρος με τον νευρώνα. Η κάθε είσοδος  $X_i$  έχει δηλαδή το δικό της βάρος  $W_{kj}$ . Ο πρώτος δείκτης εκφράζει τον αντίστοιχο νευρώνα ενώ ο δεύτερος αναφέρεται στο άκρο κάθε εισόδου που συνδέεται ο νευρώνας. Η τιμή του βάρους  $W_{kj}$  μπορεί να είναι είτε θετική είτε αρνητική. Στην ουσία το συναπτικό βάρος δηλώνει το τι βαρύτητα δίνεται στην κάθε είσοδο που δέχεται ο νευρώνας. Ο νευρώνας αθροίζει το γινόμενο της κάθε εισόδου με το συναπτικό βάρος που της αντιστοιχεί. Στη συνέχεια το άθροισμα αυτό περνά από μια συνάρτηση ενεργοποίησης. Η έξοδος του νευρώνα, αποτελεί μετέπειτα είσοδο σε έναν επόμενο νευρώνα. Στο επόμενο σχήμα παρουσιάζεται η αρχιτεκτονική ενός απλού νευρώνα.



Εικόνα 6: Ανάλυση ενός Νευρώνα

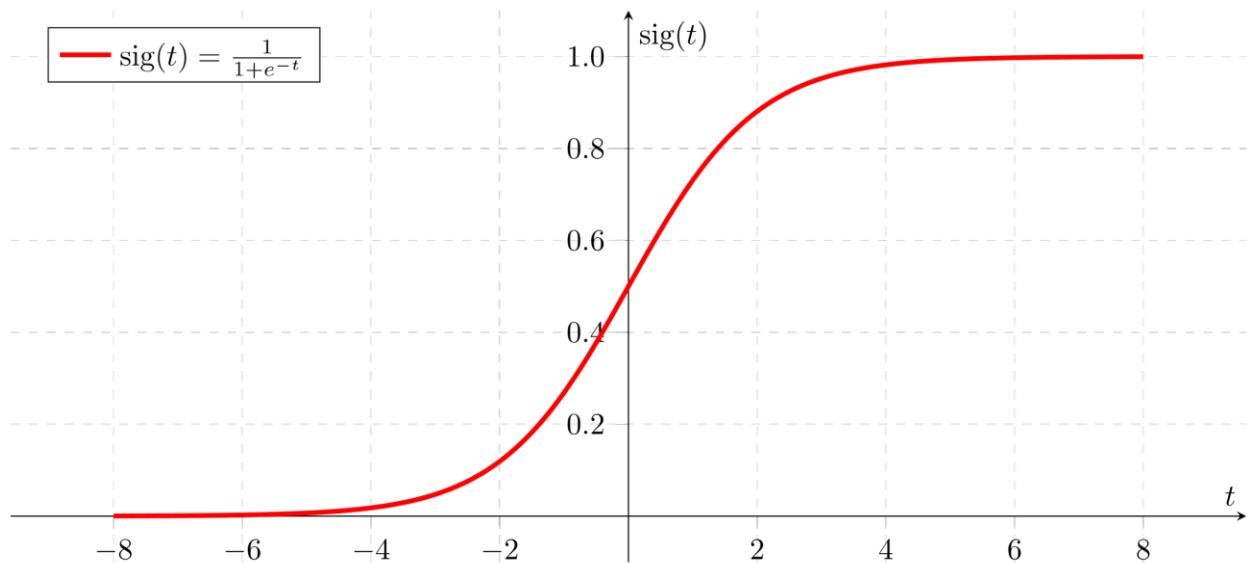
### 2.3.2 Βασικές συναρτήσεις ενεργοποίησης

- Σιγμοειδής Συνάρτηση (Sigmoid)

Η σιγμοειδής συνάρτηση αποτελεί μια αύξουσα και διαφορίσιμη συνάρτηση που ισορροπεί τη γραμμική και τη μη γραμμική συμπεριφορά. Λαμβάνει τιμές από 0 έως 1. Παράδειγμα αποτελεί η λογιστική συνάρτηση που ορίζεται:

$$\sigma(x) = \frac{1}{1 + \exp(-ax)}$$

Όπου α είναι η κλίση της συνάρτησης.



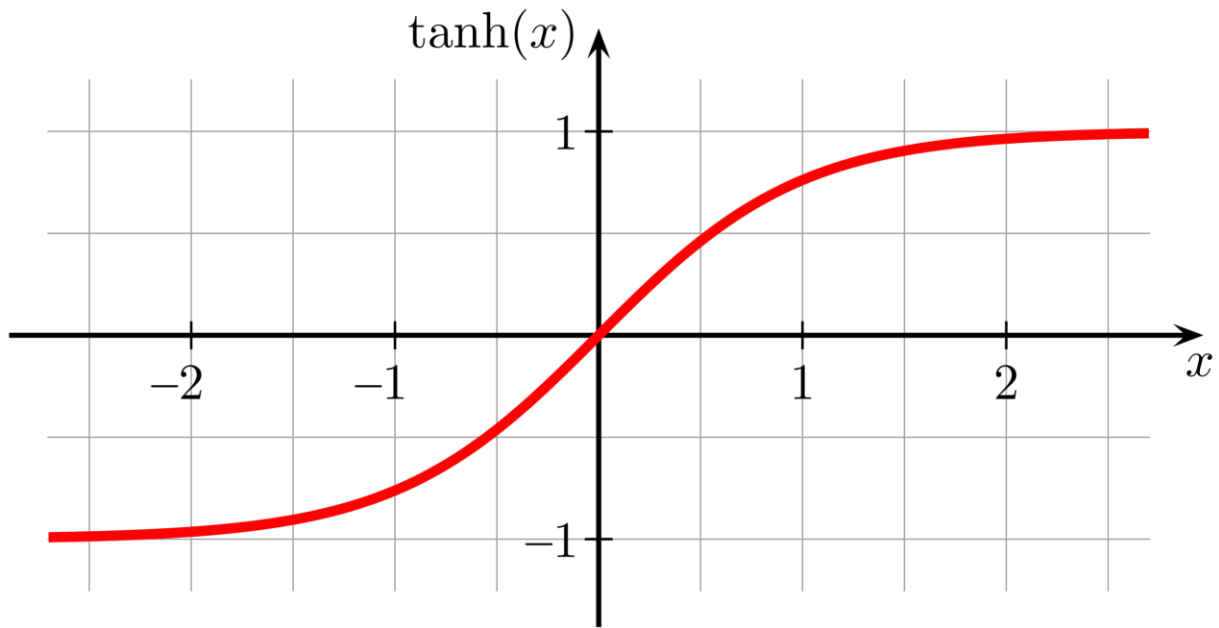
Εικόνα 7: Sigmoid(x)

- Συνάρτηση Υπερβολικής Εφαπτομένης (tanh)

Η συνάρτηση υπερβολικής εφαπτομένης λαμβάνει τιμές από -1 έως 1. Το γεγονός ότι η έξοδος ενός νευρώνα που χρησιμοποιεί ως συνάρτηση ενεργοποίησης την συνάρτηση υπερβολικής εφαπτομένης, μπορεί να πάρει αρνητικές τιμές, μπορεί να έχει σημαντική επίπτωση στην εκπαίδευση του νευρωνικού δικτύου. Αυτό τη καθιστά κατάλληλη για την έκφραση πιθανοτήτων. Επίσης υπερτερεί σε σχέση με τη σιγμοειδή συνάρτηση στο γεγονός ότι προσφέρει μεγαλύτερη κλίση κοντά στο μηδέν. Έτσι δίνεται η δυνατότητα για καλύτερη εκπαίδευση σε δίκτυα των οποίων οι τιμές εξόδου των νευρώνων πριν την συνάρτηση ενεργοποίησης κυμαίνονται κοντά στο μηδέν. Δίνεται:

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$



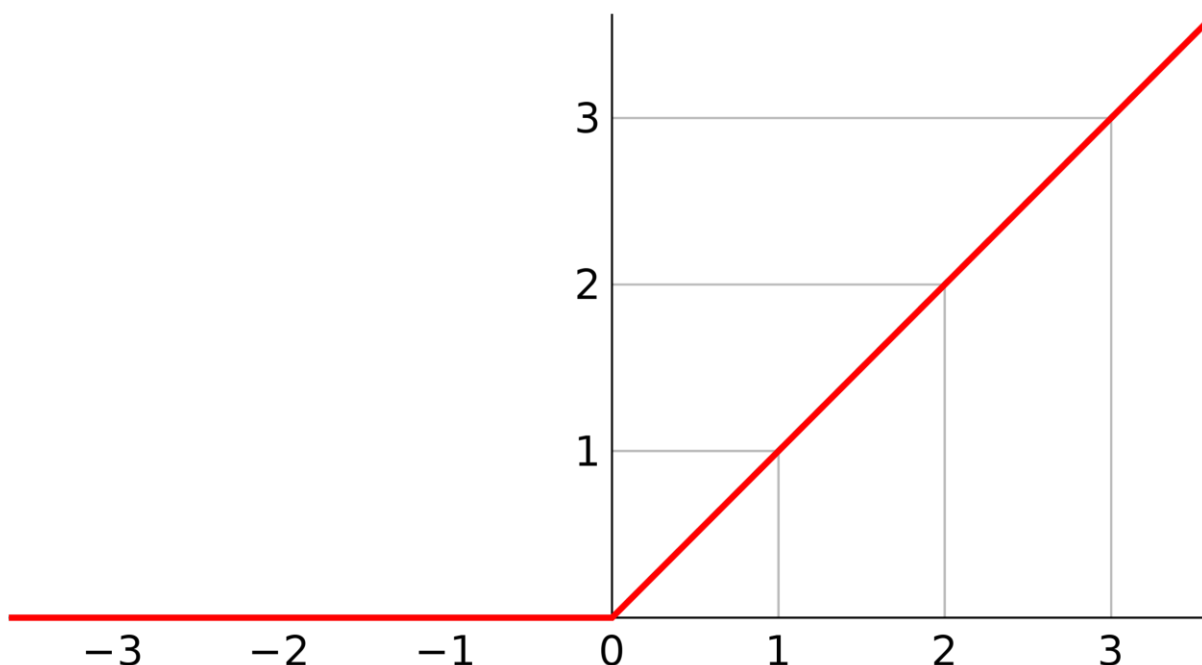


Εικόνα 8:  $\tanh(x)$

- Συνάρτηση Ανορθωμένης Γραμμικής Ράμπας (Rectified Linear Unit)

Αποτελεί μια γραμμική συνάρτηση ενεργοποίησης και η έξοδος που παράγει είναι μια μη κορεσμένη τιμή. Ο τρόπος λειτουργίας της συνάρτησης είναι ο μηδενισμός του αθροίσματος του νευρώνα εάν το άθροισμα είναι αρνητικό ή η αυτούσια επιστροφή του εάν το άθροισμα είναι μεγαλύτερο του μηδενός. Το μειονέκτημα της συγκεκριμένης συνάρτησης είναι ότι σε περίπτωση που αρκετοί νευρώνες έχουν αρνητικό άθροισμα, θα είναι μονίμως απενεργοποιημένοι. Αυτό μπορεί να αποτελέσει σημαντικό πρόβλημα στην εκπαίδευση ενός δικτύου. Η συνάρτηση δίνεται:

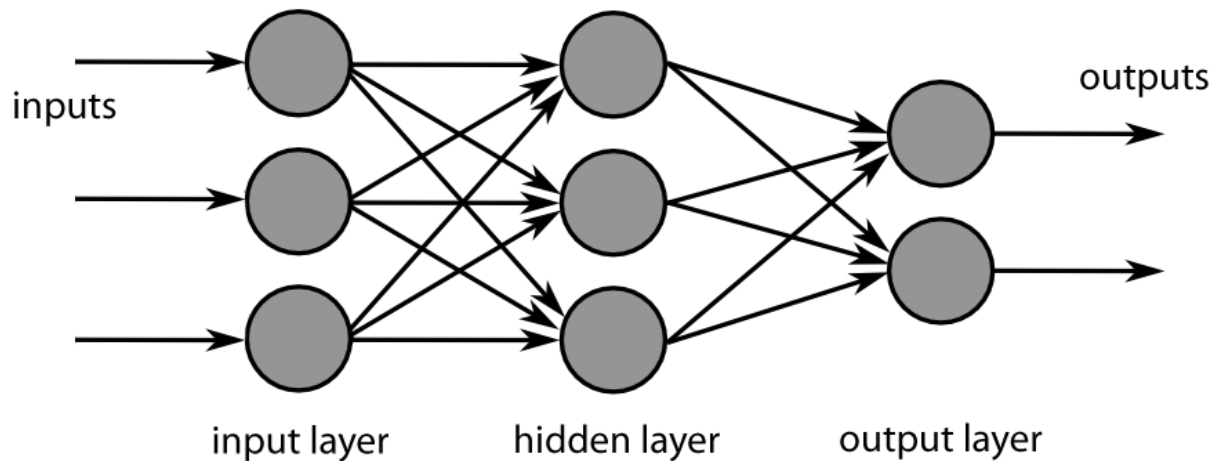
$$\text{relu}(x) = \max(0, x)$$



Εικόνα 9:  $Relu(x)$

### 2.3.3 Πολυεπίπεδα δίκτυα πρόσθιας τροφοδότησης

Τα πολυεπίπεδα δίκτυα πρόσθιας τροφοδότησης (MLF), αποτελούν τα πιο συχνά εμφανιζόμενα νευρωνικά δίκτυα. Ένα πολυεπίπεδο δίκτυο πρόσθιας τροφοδότησης αποτελείται από νευρώνες οι οποίοι σχηματίζουν στρώσεις. Το πρώτο επίπεδο ονομάζεται επίπεδο εισαγωγής (input layer), ενώ το τελευταίο επίπεδο εξόδου (output layer). Όλα τα ενδιάμεσα επίπεδα ονομάζονται κρυφά (hidden layers). Τα επίπεδα αυτά ονομάζονται έτσι γιατί δεν είναι άμεσα ορατά από τον χρήστη. Ο κάθε νευρώνας συνδέεται με όλους τους προηγούμενους νευρώνες με τον τρόπο που περιγράψαμε στην εισαγωγή του υποκεφαλαίου. Το κάθε συναπτικό βάρος του νευρώνα με όλους τους προηγούμενους χαρακτηρίζει το πόσο η κάθε είσοδος επηρεάζει το τελικό αποτέλεσμα. Στη συνέχεια το άθροισμα των γινομένων της εισόδου του νευρώνα με την κάθε είσοδο περνά από την συνάρτηση ενεργοποίησης που έχει επιλεγθεί. Προσθέτωντας περισσότερα επίπεδα σε ένα νευρωνικό δίκτυο, δίνεται η δυνατότητα να εξάγει περισσότερα αποτελέσματα για τα δεδομένα που δέχεται ως είσοδο. [21]



Εικόνα 10: Πολυεπίπεδο Νευρωνικό Δίκτυο

#### 2.3.4. Συνάρτηση Κόστους και Μέθοδοι Ελαχιστοποίησης

Η απόδοση του νευρωνικού μετράται με βάση την συνάρτηση κόστους. Η συνάρτηση κόστους μετράει το σφάλμα της εξόδου του νευρωνικού από την αναμενόμενη και επομένως “σωστή” έξοδο. Ο στόχος του δικτύου είναι να μειώσει την συνάρτηση κόστους μεταβάλλοντας τα συναπτικά βάρη του κάθε νευρώνα. Με βάση αυτή τη λογική, οι συναρτήσεις κόστους δίνουν τη δυνατότητα αναγωγής της αποδοτικής εκπαίδευσης του νευρωνικού δικτύου, σε ένα πρόβλημα βελτιστοποίησης. Πιο συγκεκριμένα το πρόβλημα που καλείται να λυθεί είναι η ελαχιστοποίηση μιας συνάρτησης κόστους.

Οι αλγόριθμοι βελτιστοποίησης είναι στην πλειοψηφία τους επαναληπτικές διαδικασίες οι οποίες υπολογίζουν μικρές διαφορές στις μεταβλητές της συνάρτησης κόστους. Έπειτα, ενημερώνονται οι μεταβλητές και η διαδικασία επαναλαμβάνεται, έως ότου η συνάρτηση κόστους λάβει την ελάχιστη τιμή της (ή περάσει ένα ορισμένο κάτω όριο). Οι δύο βασικότεροι και ευρέως χρησιμοποιούμενοι αλγόριθμοι βελτιστοποίησης είναι οι εξής:

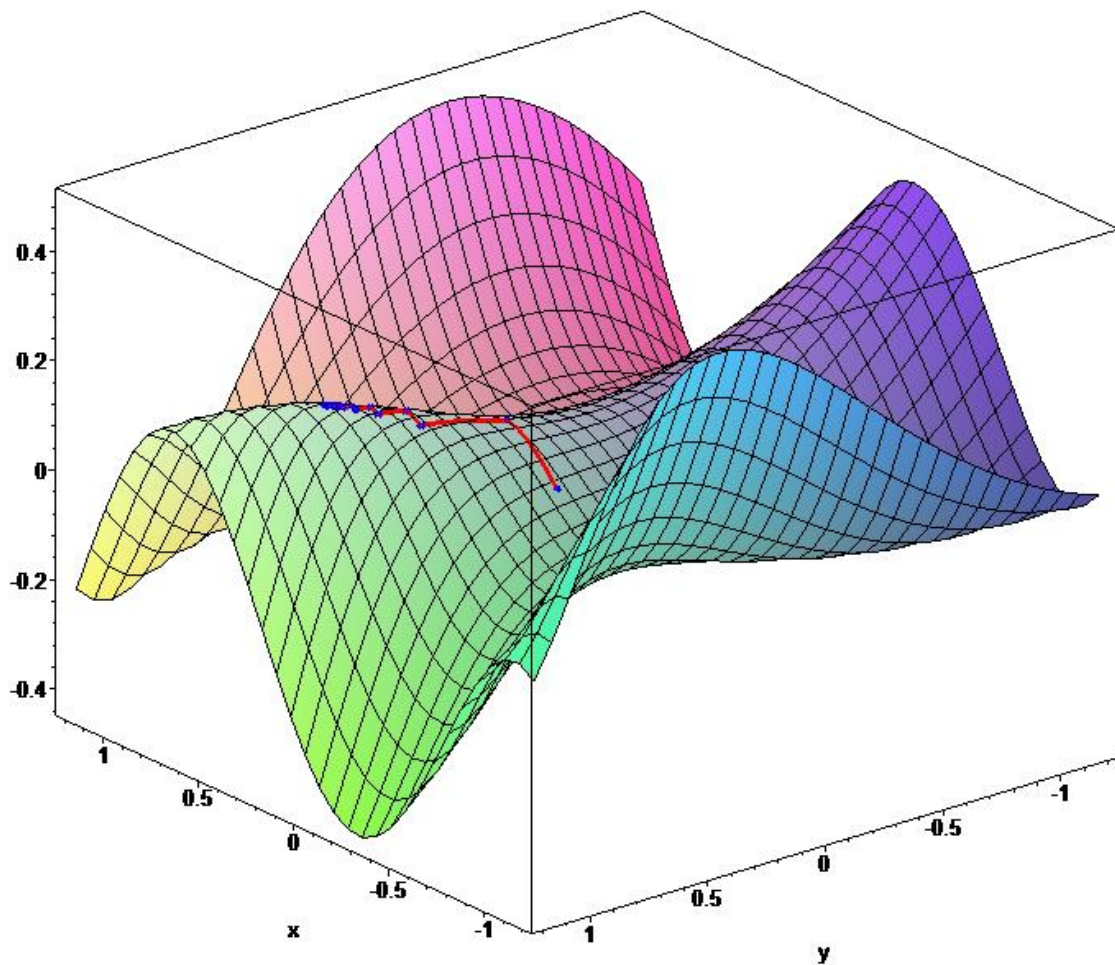
- Αλγόριθμος απότομης καθόδου - Gradient Descent

Ο αλγόριθμος της απότομης καθόδου εκτελεί μια επαναληπτική διαδικασία για τον εντοπισμό ενός τοπικού ή ιδανικότερα ενός ολικού ελαχίστου μιας παραγωγίσιμης συνάρτησης κόστους. Παραγωγίζοντας την συνάρτηση κόστους, είναι εύκολο να εντοπίσουμε την κλίση της συνάρτησης. Στη συνέχεια αφαιρώντας από τα ήδη υπάρχον βάρη, την τιμή της παραγώγου της συνάρτησης κόστους, πολλαπλασιασμένη επί μια

σταθερά  $\alpha$ , η οποία ονομάζεται ρυθμός μάθησης (learning rate), επιτυγχάνουμε την μείωση της συνάρτησης κόστους  $L(x)$  [22]. Δηλαδή:

$$\lambda_{v+1} = \lambda_v - \alpha \Delta L(x_n), \text{ όπου } \alpha \in R$$

Η τιμή του  $\alpha$ , μας δίνει τη δυνατότητα να επιλέξουμε το μέγεθος των βημάτων που θα γίνουν. Μια μικρή τιμή στο  $\alpha$ , δίνει τη δυνατότητα για καλύτερη προσέγγιση στο τοπικό - ή ολικό- ελάχιστο, με τον κίνδυνο των πολλών και χρονοβόρων βημάτων. Αντίθετα πολύ μεγάλο  $\alpha$ , εγγυάται την ταχύτερη προσέγγιση του ελαχίστου της συνάρτησης, αλλά με τον κίνδυνο της προσπέρασης του ελαχίστου.



Εικόνα 11: Αλγόριθμος απότομης καθόδου

Επεξηγηματικά, σε σχέση με την Εικόνα 11, εάν θεωρήσουμε ότι μια συνάρτηση κόστους που παρουσιάζεται στο παραπάνω σχήμα έχει αρχική τιμή που αντιστοιχεί σε ένα τοπικό μέγιστο τότε, ο αλγόριθμος καθόδου, υπολογίζει την παράγωγο της, επομένως την κλίση της συνάρτησης στο συγκεκριμένο σημείο. Στη συνέχεια, κατευθύνεται προς την αρνητική κλίση της συνάρτησης. Για το επόμενο σημείο - τιμή της συνάρτησης επαναλαμβάνεται η διαδικασία, έως ότου φτάσουμε σε ένα τοπικό ή ολικό ελάχιστο, ή έως ότου ο αλγόριθμος επαναληφθεί συγκεκριμένες φορές. Μια χρήσιμη παραλλαγή του αλγορίθμου καθόδου αποτελεί ο στοχαστικός αλγόριθμος καθόδου (Stochastic Gradient Descent) [23], ο οποίος για την ελαχιστοποίηση των πράξεων δεν επιλέγει όλα τα δείγματα εισόδου για τον υπολογισμό και την βελτιστοποίηση της συνάρτησης κόστους. Ο στοχαστικός αλγόριθμος καθόδου, επιλέγει ένα υποσύνολο των δεδομένων που έχουμε, σε αντίθεση με τον αλγόριθμο καθόδου όπου υπολογίζει σε κάθε βήμα για όλα τα σημεία τις τιμές των μερικών παραγώγων για όλες τις παραμέτρους της συνάρτησης. Περαιτέρω, ο στοχαστικός αλγόριθμος καθόδου, μειώνει την πιθανότητα για την διαδικασία μάθησης, να παγιδευτεί σε κάποιο τοπικό ελάχιστο. Για τους παραπάνω λόγους στην πραγματικότητα χρησιμοποιείται κατά βάση η Stochastic Gradient Descent.

- Αλγόριθμος ADAM (Adaptive Moment Estimation)

Ο παραπάνω αλγόριθμος παρουσιάστηκε το 2014, από τους Diederick Kingma και Jimmy Ba [24]. Ο αλγόριθμος ADAM, αποτελεί μια επέκταση του στοχαστικού αλγορίθμου κατάβασης. Η κύρια διαφορά μεταξύ των δύο αλγορίθμων, είναι ότι στον Adam, η παράμετρος του ρυθμού μάθησης (learning rate) δεν είναι ίδια για όλα τα βάρη του νευρωνικού. Ο υπολογισμός αυτού του μεγέθους γίνεται με τη βοήθεια των ροπών της συνάρτησης κλίσης, δηλαδή της πρώτης παραγώγου της συνάρτησης κόστους. Η πρώτη ροπή μιας συνάρτησης, αναφέρεται στη μέση τιμή της συνάρτησης, ενώ η δεύτερη στην διακύμανση της. Η αναδρομικότητα του αλγορίθμου γίνεται αντιληπτή και από τον τύπο υπολογισμού των ροπών. Δίνεται:

$$m_t = \beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot \Delta L(x), \text{ όπου}$$

$m_t$ , η πρώτη ροπή κλίσης

$\beta_1$ , σταθερά  $0 \leq \beta_i \leq 1$

$$v_t = \beta_2 \cdot v_{t-1} + (1 - \beta_2) \cdot \Delta L^2(x), \text{ όπου}$$

$v_t$  , η δεύτερη ροπή κλίσης  
 $\beta_2$  , σταθερά  $0 \leq \beta'_i \leq 1$

Οι τιμές με τις οποίες αρχικοποιούνται οι ροπές είναι προφανώς μηδενικές. Αυτό ωστόσο προσφέρει μια προκατάληψη στο μοντέλο η οποία αντιμετωπίζεται με την διόρθωση των τιμών, μετά τον υπολογισμό τους. Οι νέες διορθωμένες τιμές των δύο ροπών είναι:

$$m'_t = \frac{m_t}{1 - \beta_1}$$

$$v'_t = \frac{v_t}{1 - \beta_2}$$

Τέλος, ο κανόνας ενημέρωσης του μοντέλου δίνεται:

$$w_{t+1} = w_t - \frac{a}{\sqrt{v'_t} + \varepsilon} \cdot m'_t$$

## Κεφάλαιο 3: Τεχνική Επίλυση

### 3.1 Λήψη και ανάλυση δεδομένων

Για την παρούσα διπλωματική εργασία το dataset που χρησιμοποιήθηκε είναι το Thetis MRV Dataset της EMSA. Η EMSA (European Maritime Safety Agency) έχει ως στόχο την καταγραφή δεδομένων μεγάλων πλοίων στην Ευρωπαϊκή Ένωση. Το MRV Dataset (Monitoring, Reporting and Verification) περιέχει πληροφορίες που αφορούν τα έτη 2018-2020 σχετικά με εκπομπές ρύπων και κατανάλωση καυσίμων τόσο σε επίπεδο ετήσιου μέσου όρου, όσο και ανά ναυτικό μίλι σε συνάρτηση με το εκάστοτε φορτίο του πλοίου. Τα δεδομένα που περιέχει αναφέρονται σε διαφορετικού είδους πλοία (Tanker,

Dry bulk, Lng/Lpg, RoRo) όπως επίσης και σε πλοία της ίδιας κατηγορίας, αλλά διαφορετικού μεγέθους.

Η ανάλυση που πραγματοποιήθηκε, αφορά την κατανάλωση καυσίμου στα πετρελαιοφόρα (tanker) όλων των μεγεθών. Η πρώτη επεξεργασία του dataset αποσκοπούσε στη διατήρηση της πληροφορίας μόνο για τα πλοία της κατηγορίας Oil Tankers. Αρχικά, έγινε ο έλεγχος ότι η κάθε είσοδος του dataset περιέχει πληροφορία σχετικά με την κατανάλωση του πλοίου. Εισαγωγές οι οποίες είχαν έλλειψεις θεωρήθηκαν ως μη ολοκληρωμένες και επομένως δεν συμπεριλήφθηκαν στο τελικό dataset. Στη συνέχεια, έγινε μια ανάλυση των δεδομένων κατανάλωσης των πετρελαιοφόρων πλοίων και έπειτα έγινε η εκκαθάριση του dataset από outliers.

Το αρχικό dataset περιέχει σχεδόν πέντε χιλιάδες καταγραφές πετρελαιοφόρων πλοίων. Αγνοώντας τις περιπτώσεις οι οποίες είτε δεν περιέχουν πληροφορίες σχετικά με την κατανάλωση καυσίμου, είτε περιλαμβάνουν πληροφορίες, οι οποίες φαίνονται λανθασμένες, διατηρούμε συνολικά 4929 εισαγωγές πετρελαιοφόρων πλοίων οι οποίες αναφέρονται σε 2340 μοναδικά IMO. Το IMO χαρακτηρίζει έναν μοναδικό αριθμό αναγνώρισης του κάθε πλοίου [25]. Αναφέρεται στα αρχικά International Maritime Organization. Πάνω σε αυτά τα δεδομένα εφαρμόστηκαν εν τέλη τα μαθηματικά μοντέλα που θα παρουσιαστούν στη συνέχεια με στόχο την ακριβή πρόβλεψη κατανάλωσης καυσίμου πετρελαιοφόρου πλοίου.

Στις παρακάτω εικόνες παρατηρούμε το αρχικό dataset με τις ακατέργαστες πληροφορίες για όλα τα είδη πλοίων (Εικόνα 12). Στη συνέχεια μετά από την απαραίτητη επεξεργασία μέσω της βιβλιοθήκης pandas της python, κρατάμε μόνο τις πληροφορίες για τα πετρελαιοφόρα πλοία που περιέχονται στο dataset (Εικόνα 13).

```
In [98]: data = pd.read_csv('./MRV Data - Curated_info.csv')
data.head()
```

Out[98]:

	IMO Number	Name	Ship type	Reporting Period	Technical efficiency	Total fuel consumption [m tonnes]	Total CO <sub>2</sub> emissions [m tonnes]	Annual average Fuel consumption per distance [kg / n mile]	Annual average CO <sub>2</sub> emissions per distance [kg CO <sub>2</sub> / n mile]	Annual average Fuel consumption per transport work (mass) [g / m tonnes · n miles]	Annual average CO <sub>2</sub> emissions per transport work (mass) [g CO <sub>2</sub> / m tonnes · n miles]
0	6602898	OCEAN MAJESTY	Passenger ship	2020	EIV (31.73 gCO <sub>2</sub> /t·nm)	951.37	2985.13	132.07	414.4	NaN	NaN
1	6703343	EQUALITY	Other ship types	2020	EIV (57.84 gCO <sub>2</sub> /t·nm)	225.70	706.78	40.75	127.62	67.66	211.88
2	7037806	IONIAN STAR	Ro-pax ship	2020	EIV (19.4 gCO <sub>2</sub> /t·nm)	1543.32	4947.88	103.93	333.19	NaN	NaN
3	7043843	TALOS	Ro-ro ship	2020	EIV (48.71 gCO <sub>2</sub> /t·nm)	2625.33	8416.81	64.28	206.09	168.74	540.98
4	7128332	SEA WIND	Ro-pax ship	2020	EIV (9.29 gCO <sub>2</sub> /t·nm)	4152.78	13137.92	86.64	274.09	NaN	NaN

Keep only the oil tankers with values on total fuel.

Replace 'division by zero' with NaN.

New pandas with imo, name, vesselclass, annual cons per distance, annual emission per distance.

```
In [99]: data = data[(data['Ship type']=='Oil tanker')
&(data['Total fuel consumption [m tonnes]'> 0)
&(data['Total CO2 emissions [m tonnes]'> 0)]
data['Annual average Fuel consumption per transport work (mass) [g / m tonnes · n miles]'] =(
data['Annual average Fuel consumption per transport work (mass) [g / m tonnes · n miles]']
.apply(lambda x: np.nan if x=='Division by zero!' else x))
data['Annual average CO2 emissions per transport work (mass) [g CO2 / m tonnes · n miles]'] =(
data['Annual average CO2 emissions per transport work (mass) [g CO2 / m tonnes · n miles]']
.apply(lambda x: np.nan if x=='Division by zero!' else x))
data.rename(columns= {'IMO Number': 'IMO'}, inplace=True)
data.head()
```

Εικόνα 12: Η αρχική μορφή του Dataset που χρησιμοποιήθηκε

```
data.head()
```

Out[99]:

	IMO	Name	Ship type	Reporting Period	Technical efficiency	Total fuel consumption [m tonnes]	Total CO <sub>2</sub> emissions [m tonnes]	Annual average Fuel consumption per distance [kg / n mile]	Annual average CO <sub>2</sub> emissions per distance [kg CO <sub>2</sub> / n mile]	Annual average Fuel consumption per transport work (mass) [g / m tonnes · n miles]	Annual average CO <sub>2</sub> emissions per transport work (mass) [g CO <sub>2</sub> / m tonnes · n miles]
249	8905787	IRINI	Oil tanker	2020	EIV (10.4 gCO <sub>2</sub> /t·nm)	2216.30	7050.58	84.58	269.07	15.42	49.07
345	9010929	ZIRCON	Oil tanker	2020	EIV (20.45 gCO <sub>2</sub> /t·nm)	1900.10	5917.99	123.98	386.14	28.1	87.51
613	9125279	NORMANNA	Oil tanker	2020	EIV (15.36 gCO <sub>2</sub> /t·nm)	2088.77	6550.71	82.97	260.19	10.3	32.3
744	9145437	CASTILLO DE MONTERREAL	Oil tanker	2020	EIV (7.64 gCO <sub>2</sub> /t·nm)	3617.50	11345.81	121.91	382.36	11.14	34.93
782	9150614	HAI GONG YOU 306	Oil tanker	2020	EIV (12.59 gCO <sub>2</sub> /t·nm)	1220.84	3840.79	77.04	242.37	7.97	25.08

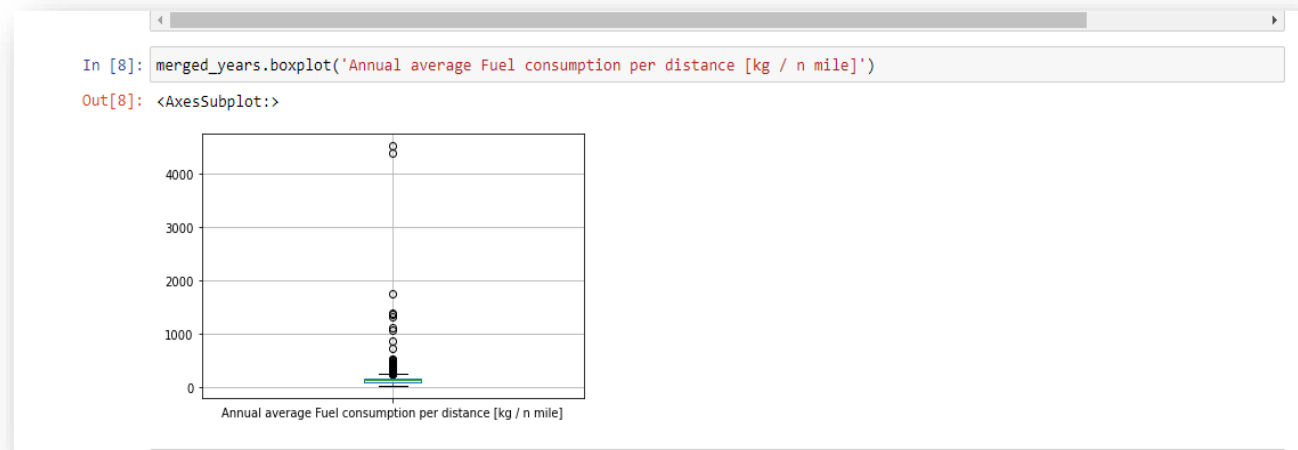
```
In [4]: data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
```

Εικόνα 13: Το επεξεργασμένο dataset



Για την αναγνώριση και εκκαθάριση των outliers, τα δεδομένα οπτικοποιήθηκαν με τη χρήση θηκόγραμματος (boxplot) [26]. Το θηκόγραμμα αποτελεί έναν γραφικό τρόπο παρουσίασης μέτρων μιας κατανομής ομαδοποιημένων δεδομένων. Περιγράφει την διασπορά, το μέσο όρο, την τοπικότητα, όπως επίσης και τα μέγιστα και ελάχιστα των εκάστοτε δεδομένων. Το θηκόγραμμα των δεδομένων που χρησιμοποιήθηκαν παρουσιάζεται στην παρακάτω εικόνα (Εικόνα 14).

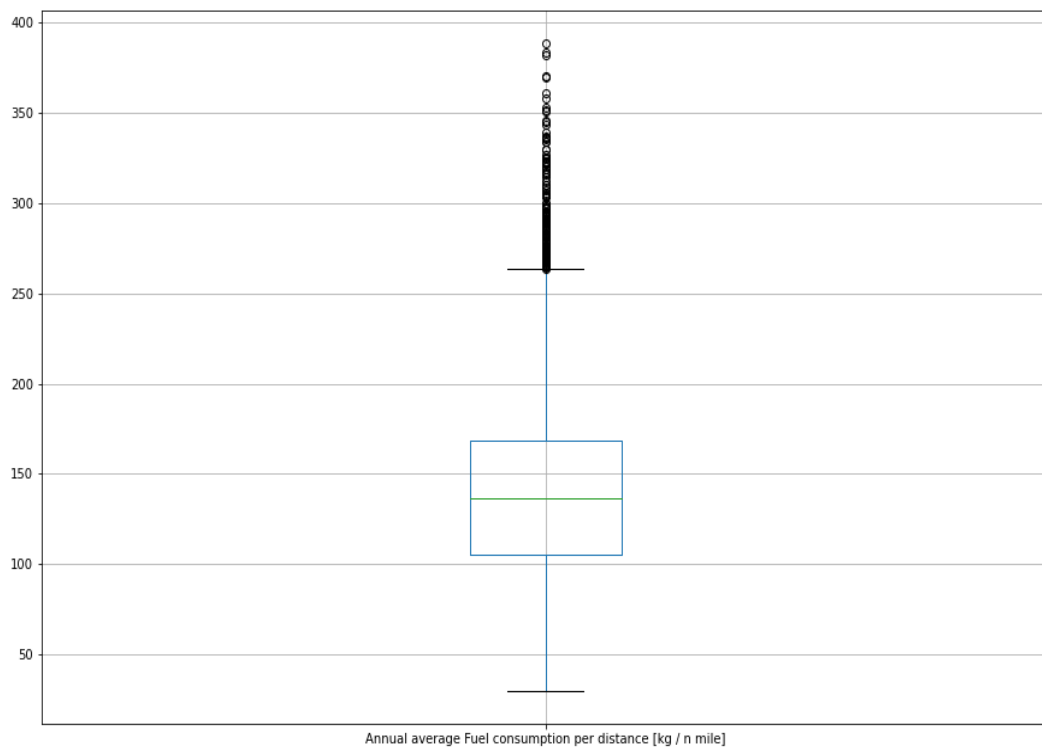


*Εικόνα 14: Θηκόγραμμα αρχικών δεδομένων*

Όπως φαίνεται στην Εικόνα 15, ο μέσος όρος των τιμών κατανάλωσης καυσίμου κυμαίνεται στα 130 κιλά καυσίμου ανά ναυτικό μίλι. Επομένως τιμές οι οποίες ξεπερνούν τα 400 κιλά ανα ναυτικό μίλι θεωρήθηκαν ως outliers, δηλαδή λάθος εισαγωγές και αποκλείστηκαν απο την συνέχεια της ανάλυσης. Το τελικό θηκόγραμμα των δεδομένων παρουσιάζεται στη συνέχεια.

```
In [12]: plt.figure(figsize = (15, 10))
avg_cons_per_distance.boxplot('Annual average Fuel consumption per distance [kg / n mile]')
```

Out[12]: <AxesSubplot:>



Εικόνα 15: Θηκόγραμμα επεξεργασμένων δεδομένων

## 3.2 Μοντέλα Πρόβλεψης

Τα μοντέλα πρόβλεψης που επιλέχθηκαν για την εύρεση των τιμών ήταν η γραμμική και η πολυωνυμική παλινδρόμηση (linear & polynomial regression). Η επιλογή έγινε λόγω της συσχέτισης που παρουσίαζαν οι μεταβλητές του προβλήματος. Οι ανεξάρτητες μεταβλητές, οι οποίες αποτελούνται από δομικά χαρακτηριστικά του πλοίου, σε σχέση με την εξαρτημένη μεταβλητή, δηλαδή την κατανάλωση καυσίμου ενός πλοίου ανά ναυτικό μίλι, διέπονται από μια γραμμική συσχέτιση. Για τον λόγο αυτό, επιλέχθηκε αρχικά το μοντέλο γραμμικής παλινδρόμησης και στη συνέχεια για τον έλεγχο και την σύγκριση των αποτελεσμάτων επιλέχθηκε το μοντέλο της πολυωνυμικής παλινδρόμησης.

Το πρώτο βήμα της διαδικασίας ήταν ο διαχωρισμός του dataset σε training και testing τιμές. Ως training dataset, ορίζουμε τον όγκο των δεδομένων πάνω στον οποίο εκπαιδεύεται το μοντέλο. Στη συνέχεια το testing dataset χρησιμοποιείται για την επαλήθευση των προβλεπόμενων εξόδων του μοντέλου και τελικά για την αξιολόγηση του. Το testing dataset, επιλέχθηκε να είναι το 30% του συνολικού τόσο στο γραμμικό, όσο και στο πολυωνυμικό μοντέλο. Ο λόγος που επιλέχθηκε η συγκεκριμένη διάσπαση του dataset σε εκπαίδευση και αξιολόγηση, είναι η αποφυγή του overfitting. [27]

Στη συνέχεια το μοντέλο εκπαιδεύτηκε με βάση το εκάστοτε μαθηματικό μοντέλο πρόβλεψης. Οι εξαρτημένες μεταβλητές που επιλέχθηκαν αφορούν το deadweight, το gross rated tonnage, το main engine kilowatts και τέλος το year built. Οι μεταβλητές αυτές μπορούν να δώσουν αρκετές πληροφορίες για την πλήρη περιγραφή ενός πλοίου. Περαιτέρω δεδομένα για τα χαρακτηριστικά του πλοίου αφενός δεν θα συμβάλουν στην καλύτερη πρόβλεψη της κατανάλωσης του και αφετέρου η διαδικασία εισαγωγής θα αποτελέσει χρονοβόρα και κουραστική διαδικασία για τον χρήστη. Αναλυτικά οι μεταβλητές που επιλέχθηκαν εκφράζουν:

- **Deadweight:** Ορίζει το βάρος του φορτίου που επιτρέπεται να κουβαλήσει το εκάστοτε πλοίο. Είναι το άθροισμα των βαρών του φορτίου (cargo load), των καυσίμων, του πόσιμου νερού, του έρματος, των προμηθειών και του πληρώματος. Αποτελεί ένα μέτρο της μεταφορικής ικανότητας ενός πλοίου και εκφράζεται σε τόνους (mts - metric tons).
- **Gross Rated Tonnage:** Περιγράφει τον συνολικό εσωτερικό χώρο ενός πλοίου. Στην ολική χωρητικότητα περιλαμβάνονται όλοι οι μονίμως κλειστοί χώροι που διατίθενται για φορτίο, εφόδια πλοίου και ενδιαίτησης του πληρώματος. Προσδιορίζεται με συγκεκριμένη μονάδα μέτρησης του όγκου. Με αυτή τη μετρική αποδίδονται οι πραγματικές διαστάσεις του πλοίου και περιλαμβάνεται απαραίτητα στην νηολόγηση, αλλά και σε όλα τα πιστοποιητικά του.
- **Main Engine Kw:** Εκφράζει την απόδοση του κινητήρα του πλοίου. Η μονάδα μέτρησης είναι οι κιλοβατώρες (KW).
- **Year Built:** Αναφέρεται στην χρονολογία κατασκευής του πλοίου.

Η απόδοση τόσο του γραμμικού όσο και του πολυωνυμικού μοντέλου κρίθηκε με βάση τον υπολογισμό του συντελεστή συσχέτισης  $R^2$  για τους λόγους τους οποίους εξηγήσαμε στο Κεφάλαιο 2. Στη συνέχεια έγινε η γραφική παράσταση των εκτιμώμενων από τα μαθηματικά μοντέλα τιμών, σε συνάρτηση με τις πραγματικές τιμές που περιέχει το dataset.

### Linear Model

```
In [38]: from sklearn.model_selection import train_test_split  
  
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = 0.3)
```

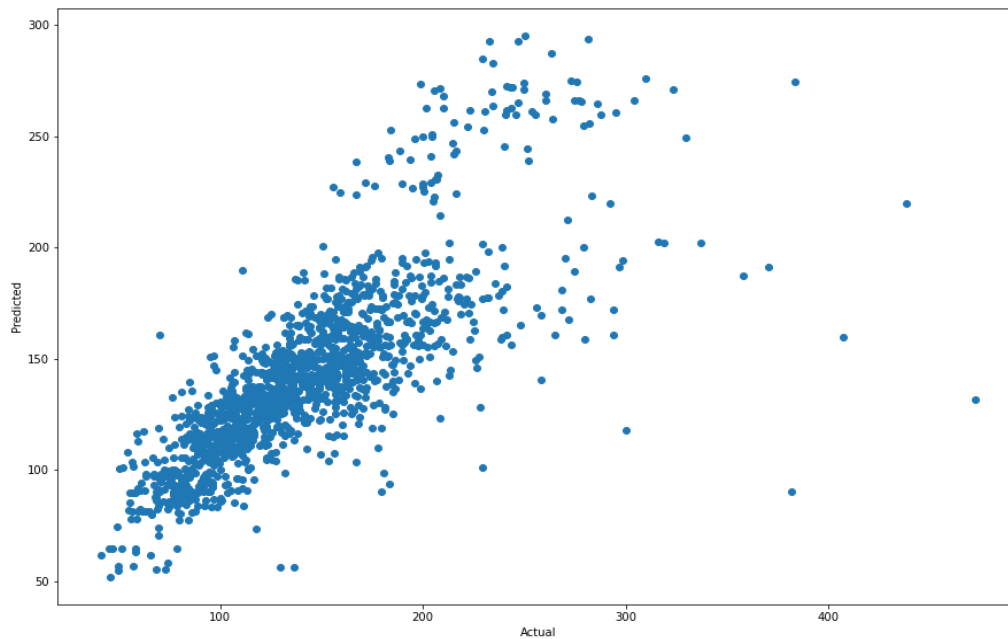
```
In [39]: from sklearn.linear_model import LinearRegression  
from sklearn.metrics import r2_score  
  
ml = LinearRegression()  
ml.fit(x_train, y_train)  
y_pred_linear = ml.predict(x_test)  
  
r2_score(y_test, y_pred_linear)
```

Out[39]: 0.6038214080178979

Εικόνα 16: Απόδοση γραμμικού μοντέλου

```
In [40]: plt.figure(figsize = (15, 10))  
plt.scatter(y_test, y_pred_linear)  
plt.xlabel('Actual')  
plt.ylabel('Predicted')
```

Out[40]: Text(0, 0.5, 'Predicted')



Εικόνα 17: Actual vs Predicted Linear

### Polygonomial Model

```
In [41]: # Fitting Polynomial Regression to the dataset
from sklearn.preprocessing import PolynomialFeatures

poly_reg = PolynomialFeatures(degree=2)
X_poly = poly_reg.fit_transform(x_train)
pol_reg = LinearRegression()
pol_reg.fit(X_poly, y_train)
```

```
Out[41]: LinearRegression()
```

```
In [42]: y_pred_poly = pol_reg.predict(poly_reg.fit_transform(x_test))
```

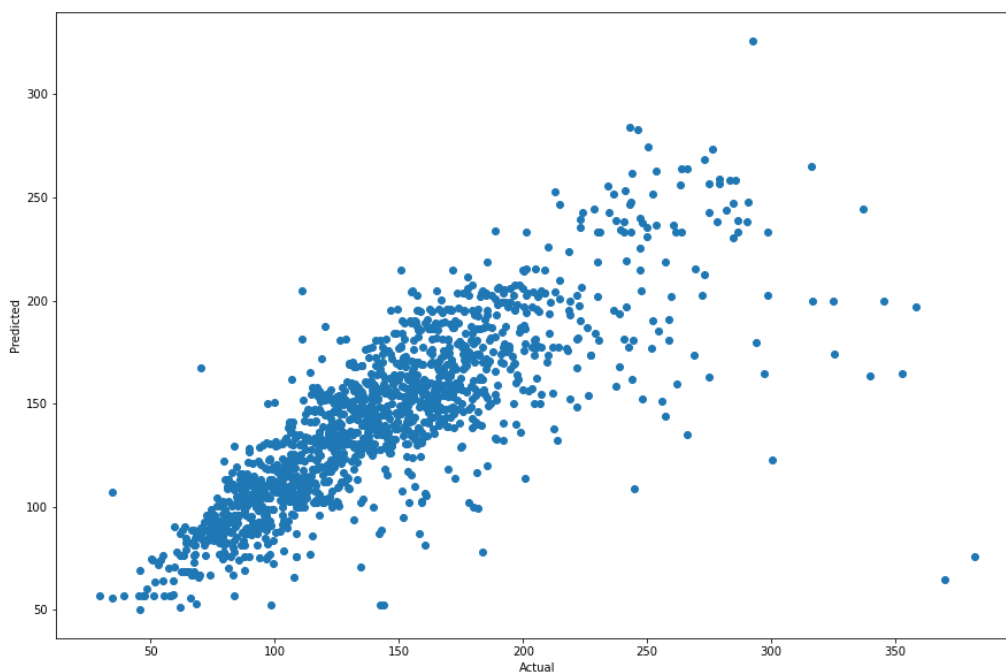
```
In [43]: r2_score(y_test, y_pred_poly)
```

```
Out[43]: 0.6546489817618074
```

Εικόνα 18: Απόδοση πολυωνυμικού μοντέλου

```
In [46]: plt.figure(figsize = (15, 10))
plt.scatter(y_test, y_pred_poly)
plt.xlabel('Actual')
plt.ylabel('Predicted')
```

```
Out[46]: Text(0, 0.5, 'Predicted')
```



Εικόνα 19: Actual vs Predicted Polynomial

Στη συνέχεια πραγματοποιήθηκε η υλοποίηση ενός νευρωνικού δικτύου. Ο λόγος υλοποίησης του νευρωνικού δικτύου έγκειται στην αποφυγή των παραδοχών που κάνουν

τα προηγούμενα μοντέλα τα οποία χρησιμοποιήθηκαν. Πιο συγκεκριμένα τόσο η γραμμική, όσο και η πολυωνυμική παλινδρόμηση λειτουργούν με βάση την παραδοχή της συσχέτισης γραμμικότητας των εξαρτημένων μεταβλητών με τις ανεξάρτητες μεταβλητές. Αντίθετα, ένα νευρωνικό δίκτυο, δεν υποθέτει την συσχέτιση γραμμικότητας μεταξύ των δεδομένων και ανταπεξέρχεται καλύτερα σε μη γραμμικά δεδομένα εισόδου.

Τα αρχικό βήμα στην υλοποίηση του συγκεκριμένου μοντέλου, παρόμοια με τα μοντέλα παλινδρόμησης είναι ο διαχωρισμός του dataset σε training/testing. Το testing dataset, αποτελεί το 30% του συνολικού αρχικού όγκου δεδομένων. Το νευρωνικό δίκτυο αποτελείται από το input και 2 ακόμη layers. Το input έχει 5 εισόδους, όσες είναι οι εξαρτημένες μεταβλητές του μοντέλου.

Η συνάρτηση ενεργοποίησης που χρησιμοποιήθηκε είναι η  $\text{Relu}(x)$  για το input layer. Ο λόγος χρήσης της Relu, έγκνται στο γεγονός ότι καμία από τις εισόδους του νευρωνικού δεν είναι μηδενική, δεδομένου ότι όλες περιγράφουν δομικά χαρακτηριστικά του πλοίου. Έτσι, δεν υπάρχει κίνδυνος απενεργοποίησης οποιουδήποτε νευρώνα. Δεδομένου ότι ο κίνδυνος απενεργοποίησης είναι μηδενικός, επιλέχθηκε η Relu, λόγω του απλού και γρήγορου υπολογισμού της. Αποτελεί μια συνάρτηση, όπου όπως ειπώθηκε και στο προηγούμενο κεφάλαιο, είναι ιδιαίτερα απλή υπολογιστικά. Το επόμενο layer έχει 5 units και συνάρτηση ενεργοποίησης την  $\text{Tanh}(x)$ . Στο συγκεκριμένο επίπεδο, επιλέξαμε τη χρήση της παραπάνω συνάρτησης για να εκμεταλευτούμε την κλίση που υπάρχει κοντά στο μηδέν.

Η συνάρτηση κόστους του νευρωνικού δικτύου είναι η mean squared error function και για την ελαχιστοποίηση της συνάρτησης κόστους χρησιμοποιήθηκε ο αλγόριθμος ADAM, όπου περιγράψαμε στο προηγούμενο Κεφάλαιο.

Αρχικά η εκπαίδευση του νευρωνικού δικτύου έγινε με τις εξής παραμέτρους, όπως φαίνεται στην παρακάτω εικόνα (Εικόνα 20):

- Batch size = 10
- Epochs = 50

```
In [168]: model = Sequential()
model.add(Dense(units=5, input_dim=X_train.shape[1], kernel_initializer = 'normal', activation = 'relu'))
# Defining the Second Layer of the model
model.add(Dense(units=5, kernel_initializer='normal', activation='tanh'))

# The output neuron is a single fully connected node
# Since we will be predicting a single number
model.add(Dense(1, kernel_initializer='normal'))

# Compiling the model
model.compile(loss='mean_squared_error', optimizer='adam')

# Fitting the ANN to the Training set
model.fit(X_train, y_train ,batch_size = 10, epochs = 50, verbose=1)

Epoch 42/50
164/164 [=====] - 0s 593us/step - loss: 0.0073
Epoch 43/50
164/164 [=====] - 0s 538us/step - loss: 0.0072
Epoch 44/50
164/164 [=====] - 0s 624us/step - loss: 0.0072
Epoch 45/50
164/164 [=====] - 0s 612us/step - loss: 0.0072
Epoch 46/50
164/164 [=====] - 0s 575us/step - loss: 0.0072
Epoch 47/50
164/164 [=====] - 0s 587us/step - loss: 0.0072
Epoch 48/50
164/164 [=====] - 0s 557us/step - loss: 0.0073
Epoch 49/50
164/164 [=====] - 0s 581us/step - loss: 0.0072
Epoch 50/50
164/164 [=====] - 0s 602us/step - loss: 0.0072

Out[168]: <keras.callbacks.History at 0x27ed00702e0>

In [169]: # Defining a function to find the best parameters for ANN
```

*Εικόνα 20: Πρώτη εκπαίδευση του νευρωνικού δικτύου*

Στη συνέχεια με τη βοήθεια συνάρτησης η οποία γράφτηκε σε python, βρέθηκαν οι βέλτιστες τιμές των παραμέτρων τόσο για το batch size όσο και για τα epochs, για τις οποίες το νευρωνικό δίκτυο πετυχαίνει την μεγαλύτερη δυνατή επίδοση - 78.85%. Ο τρόπος με τον οποίον επιλέχθηκαν οι μεταβλητές εκπαίδευσης του νευρωνικού δικτύου είναι μέσω Grid Search. Ο όρος αυτός εκφράζει την διαδικασία κατά την οποία γίνεται spot - checking διαφόρων συνδυασμών των μεταβλητών εκπαίδευσης οι οποίοι είναι γνωστό ότι δουλεύουν σωστά. Ο χρήστης ορίζει ένα διάστημα μεταξύ του οποίου θα κινηθούν οι μεταβλητές και εκπαιδεύει το νευρωνικό για τον κάθε συνδυασμό. Τέλος, επιλέγεται ο συνδυασμός των μεταβλητών για τις οποίες επιτεύχθηκε η βέλτιστη απόδοση. Στο συγκεκριμένο νευρωνικό οι παράμετροι πάνω στις οποίες δοκιμάσαμε grid search ήταν το batch size και τα epochs. Στην Εικόνα 21, παρατηρούμε τη συνάρτηση η οποία εκτελεί το grid search. Επιπρόσθετα στην Εικόνα 22 φαίνεται το γράφημα με τις διάφορες τιμές απόδοσης του νευρωνικού δικτύου για τους πιθανούς συνδυασμούς των παραμέτρων. Οι τιμές για τις οποίες πήραμε την βέλτιστη απόδοση είναι:

- Batch size = 20
- Epochs = 50

```

In [169]: # Defining a function to find the best parameters for ANN
def FunctionFindBestParams(X_train, y_train, X_test, y_test):

    # Defining the list of hyper parameters to try
    batch_size_list=[5, 10, 15, 20]
    epoch_list = [5, 10, 50, 100]

    SearchResultsData=pd.DataFrame(columns=['TrialNumber', 'Parameters', 'Accuracy'])

    # initializing the trials
    TrialNumber=0
    for batch_size_trial in batch_size_list:
        for epochs_trial in epoch_list:
            TrialNumber+=1
            # create ANN model
            model = Sequential()
            # Defining the first layer of the model
            model.add(Dense(units=5, input_dim=X_train.shape[1], kernel_initializer='normal', activation='relu'))

            # Defining the Second layer of the model
            model.add(Dense(units=5, kernel_initializer='normal', activation='relu'))

            # The output neuron is a single fully connected node
            # Since we will be predicting a single number
            model.add(Dense(1, kernel_initializer='normal'))

            # Compiling the model
            model.compile(loss='mean_squared_error', optimizer='adam')

            # Fitting the ANN to the Training set
            model.fit(X_train, y_train ,batch_size = batch_size_trial, epochs = epochs_trial, verbose=0)

            MAPE = np.mean(100 * (np.abs(y_test-model.predict(X_test))/y_test))

            # printing the results of the current iteration
            print(TrialNumber, 'Parameters:', 'batch_size:', batch_size_trial, '-', 'epochs:', epochs_trial, 'Accuracy:', 100-MAPE)

            SearchResultsData=SearchResultsData.append(pd.DataFrame(data=[[TrialNumber, str(batch_size_trial)+'-'+str(epochs_trial)],
                                                                           columns=['TrialNumber', 'Parameters', 'Accuracy'] ]))

    return(SearchResultsData)

#####
# Calling the function
ResultsData=FunctionFindBestParams(X_train, y_train, X_test, y_test)

```

Εικόνα 21: Grid Search function για εύρεση των υπερπαραμέτρων



```

1 Parameters: batch_size: 5 - epochs: 5 Accuracy: 45.70273699763583
2 Parameters: batch_size: 5 - epochs: 10 Accuracy: 78.57794035933108
3 Parameters: batch_size: 5 - epochs: 50 Accuracy: 75.26735301826993
4 Parameters: batch_size: 5 - epochs: 100 Accuracy: 76.58308975917623
5 Parameters: batch_size: 10 - epochs: 5 Accuracy: 74.5228100188311
6 Parameters: batch_size: 10 - epochs: 10 Accuracy: 78.82238939139579
7 Parameters: batch_size: 10 - epochs: 50 Accuracy: 73.09391183397977
8 Parameters: batch_size: 10 - epochs: 100 Accuracy: 75.54490827605744
9 Parameters: batch_size: 15 - epochs: 5 Accuracy: 71.93914658621212
10 Parameters: batch_size: 15 - epochs: 10 Accuracy: 74.02986511371475
11 Parameters: batch_size: 15 - epochs: 50 Accuracy: 75.01642140044487
12 Parameters: batch_size: 15 - epochs: 100 Accuracy: 74.50240810593858
13 Parameters: batch_size: 20 - epochs: 5 Accuracy: 58.339940724319554
14 Parameters: batch_size: 20 - epochs: 10 Accuracy: 73.5323715676017
15 Parameters: batch_size: 20 - epochs: 50 Accuracy: 78.85780954479691
16 Parameters: batch_size: 20 - epochs: 100 Accuracy: 75.68755639328813

```

```

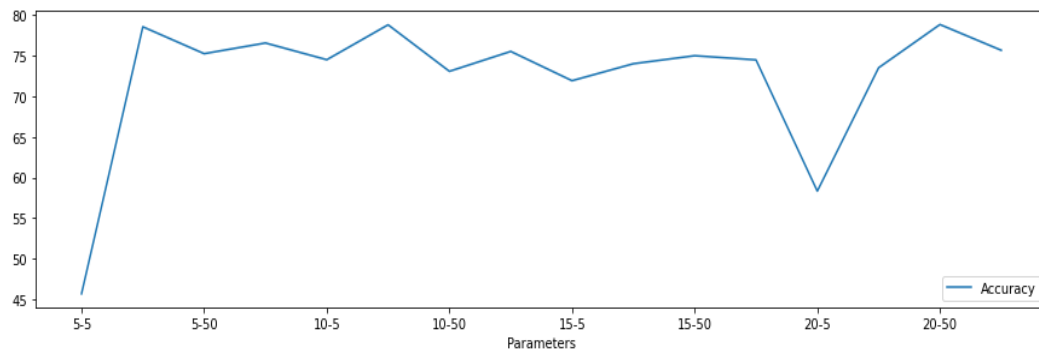
In [170]: %matplotlib inline
ResultsData.plot(x='Parameters', y='Accuracy', figsize=(15,4), kind='line')

```

```

Out[170]: <AxesSubplot:xlabel='Parameters'>

```



*Εικόνα 22: Αποδοτικότητα Νευρωνικού για διάφορες τιμές*

Στη συνέχεια, με βάση την παραπάνω ανάλυση για τις υπερπαραμέτρους και τις αντίστοιχες επιδόσεις που προκύπτουν από αυτές, έγιναν οι κατάλληλες διορθώσεις. Το νευρωνικό επανεκπαιδεύτηκε με τις νέες τιμές τις οποίες πετυχαίνει την βέλτιστη επίδοση (Εικόνα 23).

```

In [171]: # Fitting the ANN to the Training set
model.fit(X_train, y_train, batch_size = 20, epochs = 50, verbose=0)

# Generating Predictions on testing data
Predictions=model.predict(X_test)

# Scaling the predicted Price data back to original price scale
Predictions=TargetVarScalerFit.inverse_transform(Predictions)

# Scaling the y_test Price data back to original price scale
y_test_orig=TargetVarScalerFit.inverse_transform(y_test)

# Scaling the test data back to original scale
Test_Data=PredictorScalerFit.inverse_transform(X_test)

TestingData=pd.DataFrame(data=Test_Data, columns=Predictors)
TestingData['Annual_Consumption']=y_test_orig
TestingData['Predicted_Annual_Consumption']=Predictions
TestingData.head()

```

Out[171]:

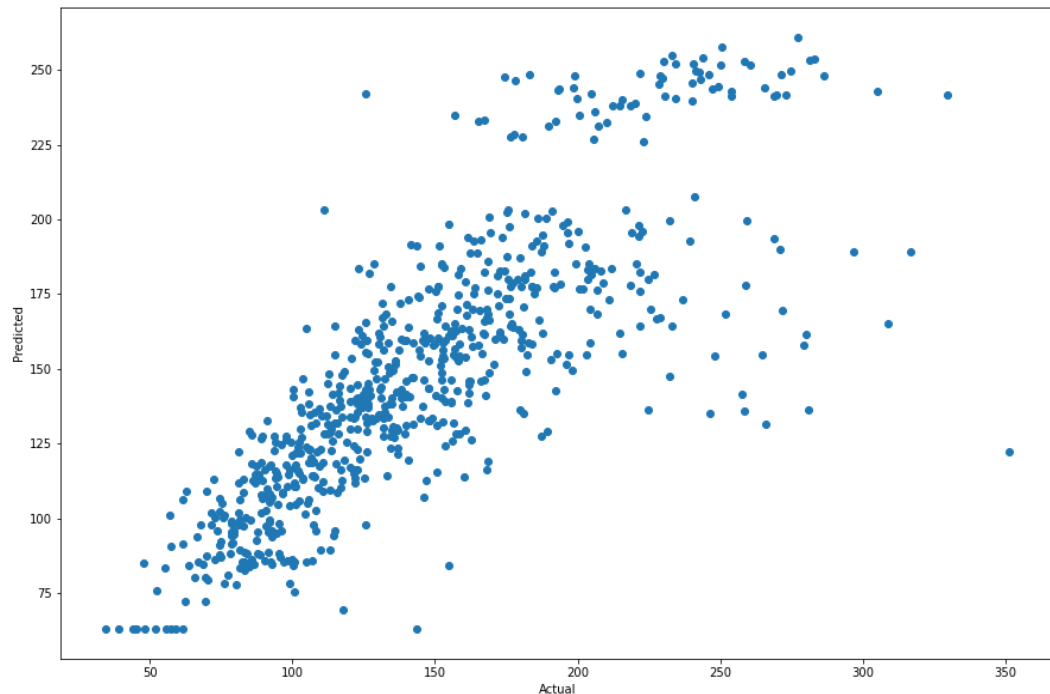
	MainEngineKW	Deadweight	GrossRatedTonnage	Yearbuilt	Annual_Consumption	Predicted_Annual_Consumption
0	9960.0	49990.0	29806.0	2015.0	90.62	92.225296
1	16080.0	108560.0	60855.0	2019.0	91.13	131.452744
2	18660.0	158670.0	81509.0	2011.0	175.69	181.087967
3	11110.0	111890.0	60435.0	2020.0	123.42	119.956665
4	13450.0	115711.0	62370.0	2017.0	103.07	135.850693

### Εικόνα 23: Επανεκπαίδευση Νευρωνικού

Τέλος, έγινε η γραφική παράσταση των εκτιμώμενων απο το νευρωνικό δίκτυο τιμών, σε συνάρτηση με τις πραγματικές τιμές που περιέχει το dataset (Εικόνα 24), με στόχο να γίνουν αντιληπτές οι οποιεσδήποτε μεγάλες παρεκκλίσεις.

```
In [13]: plt.figure(figsize = (15, 10))
plt.scatter(TestingData.Annual_Consumption, TestingData.Predicted_Annual_Consumption)
plt.xlabel('Actual')
plt.ylabel('Predicted')
```

```
Out[13]: Text(0, 0.5, 'Predicted')
```



```
In [ ]: |
```

*Εικόνα 24: Actual vs Predicted Neural Network*

### 3.3 Σύγκριση Αποτελεσμάτων

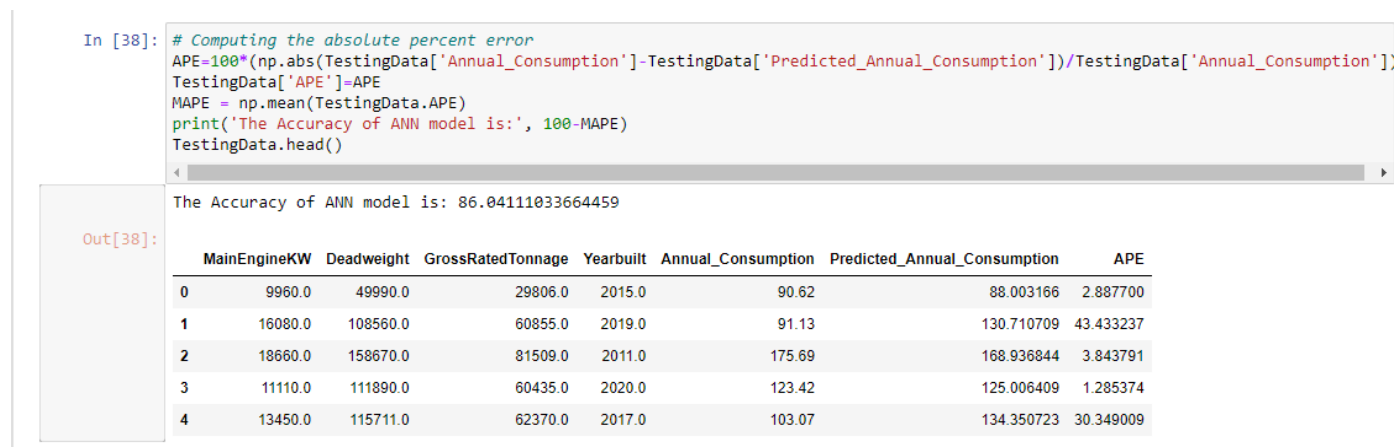
Για την ορθή σύγκριση των αποτελεσμάτων του γραμμικού και του πολυωνυμικού μοντέλου, επιλέχθηκε ο συντελεστής συσχέτισης R squared ( $r^2$ ). Παρατηρήσαμε ότι τα 2 μοντέλα βρίσκονται κοντά σε θέματα απόδοσης.

Συγκεκριμένα το γραμμικό μοντέλο απέδωσε συντελεστή συσχέτισης  $r^2$  ίσο με 0.60, ενώ το πολυωνυμικό μοντέλο ίσο με 0.65. Αξίζει να σημειωθεί ότι μικρές διαφορές στην απόδοση των δύο μαθηματικών μοντέλων, μπορούν να εμφανιστούν ανάλογα με τον διαχωρισμό μεταξύ του training και του testing dataset.

Λαμβάνοντας υπόψη το ποσοστό λάθους που υπάρχει σε ένα dataset στο οποίο η εισαγωγή πληροφοριών γίνεται από διαφορετικούς οργανισμούς χωρίς κάποια ιδιαίτερη

επίβλεψη, η απόδοση των μοντέλων είναι αρκετά καλή, κυρίως λόγω της γραμμικότητας των δεδομένων.

Η απόδοση του νευρωνικού δικτύου κρίθηκε με βάση την απόλυτη διαφορά της προβλεπόμενης τιμής σε σχέση με τη πραγματική τιμή διά την πραγματική τιμή της κατανάλωσης καυσίμου ανα ναυτικό μίλι (Εικόνα 25).



```
In [38]: # Computing the absolute percent error
APE=100*(np.abs(TestingData['Annual_Consumption']-TestingData['Predicted_Annual_Consumption'])/TestingData['Annual_Consumption'])
TestingData['APE']=APE
MAPE = np.mean(TestingData.APE)
print('The Accuracy of ANN model is:', 100-MAPE)
TestingData.head()
```

Out[38]:

	MainEngineKW	Deadweight	GrossRatedTonnage	Yearbuilt	Annual_Consumption	Predicted_Annual_Consumption	APE
0	9960.0	49990.0	29806.0	2015.0	90.62	88.003166	2.887700
1	16080.0	108560.0	60855.0	2019.0	91.13	130.710709	43.433237
2	18660.0	158670.0	81509.0	2011.0	175.69	168.936844	3.843791
3	11110.0	111890.0	60435.0	2020.0	123.42	125.006409	1.285374
4	13450.0	115711.0	62370.0	2017.0	103.07	134.350723	30.349009

Εικόνα 25: Απόδοση του νευρωνικού

## Κεφάλαιο 4: Γραφικό Περιβάλλον Εφαρμογής

### 4.1 Γραφικό Περιβάλλον Χρήστη

Τα περισσότερα σύγχρονα προγράμματα αποτελούνται τόσο από το backend κώδικα όσο και από το γραφικό περιβάλλον χρήστη (Graphical User Interface - GUI). Το user interface προσφέρει στο χρήστη μια πιο εύκολη εμπειρία στη διαχείριση του προγράμματος, ταυτόχρονα με μια βελτιστοποιημένη αλληλεπίδραση μεταξύ του χρήστη με το εκάστοτε πρόγραμμα. Το προσδοκώμενο αποτέλεσμα από την υιοθέτηση ενός σωστά σχεδιασμένου γραφικού περιβάλλοντος είναι να γίνει το περιβάλλον εργασίας στον υπολογιστή όμορφο, εύχρηστο και λειτουργικό.

Προτού καθιερωθεί το γραφικό περιβάλλον χρήστη, η επαφή του χρήστη με τον μικροϋπολογιστή γινόταν μέσω της γραμμής εντολών κάνοντας χρήση κάποιου κέλυφους (shell). Το γεγονός αυτό έκανε τους μικροϋπολογιστές δύσχρηστους έως απρόσιτους για

το ευρύ κοινό και συνεπαγόταν το να θεωρούνται οι χρήστες τους άτομα με σχετικά υψηλή εξειδίκευση στον τομέα της πληροφορικής.

Ένα εικονικό περιβάλλον χρησιμοποιεί ένα συνδυασμό τεχνολογιών και συσκευών για να παρέχει ένα λειτουργικό πλαίσιο με το οποίο ο χρήστης αλληλεπιδρά για να συγκεντρώσει και να διαχειριστεί πληροφορίες. Μια σειρά εικονογραφικών στοιχείων που δημιουργήθηκαν με βάση μία εικονογραφική γλώσσα προγραμματισμού εξελίχθηκαν για να αντιπροσωπεύσουν με ένα δομημένο και ευκολονόητο τρόπο την οργάνωση των πληροφοριών που αποθηκεύονται στους υπολογιστές. Αυτό διευκολύνει χρήστες με λίγες γνώσεις χρήσης υπολογιστών να χρησιμοποιήσουν το λογισμικό ενός υπολογιστή.

Η οπτική σύνθεση και η χρονική συμπεριφορά του εικονικού περιβάλλοντος είναι το πιο σημαντικό μέρος του προγραμματισμού των εφαρμογών όταν πρόκειται για τον τομέα επικοινωνίας ανθρώπου υπολογιστή. Ο στόχος είναι η ενίσχυση της αποτελεσματικότητας και της ευκολίας της χρήσης τέτοιων εφαρμογών, θέτοντας τον ως βασικό σχεδιασμό του προγράμματος, γνωστό ως χρηστικότητα. Η αλληλεπίδραση χρήστη – πληροφορίας γίνεται με τον χειρισμό των «οπτικών» αντικειμένων (widgets) τα οποία αντιδρούν ανάλογα με τα δεδομένα που τους δόθηκαν. Τα widget ενός καλοσχεδιασμένου GUI σχεδιάζονται με σκοπό να εκτελούν τις κατάλληλες ενέργειες ώστε ο χρήστης να παίρνει το κατάλληλο αποτέλεσμα. Τα widgets άλλοτε είναι μεγάλα σε μέγεθος, όπως είναι το πλαίσιο ή το παράθυρο μίας εφαρμογής ή ενός σχεδίου. Άλλοτε όμως είναι μικρά και χρησιμοποιούνται ως εργαλεία εισαγωγής με τα οποία μπορεί και αλληλεπιδρά ο χρήστης με τον υπολογιστή.

Δεδομένου ότι η εργασία αυτή απευθύνεται σε άτομα τα οποία δεν θεωρούνται άτομα με υψηλή εξειδίκευση στον τομέα της πληροφορικής, κρίθηκε σκόπιμο να δημιουργηθεί ένα γραφικό περιβάλλον χρήστη για την επικοινωνία του χρήστη με τα κελιά εισαγωγής του μοντέλου πρόβλεψης κατανάλωσης καυσίμου ανα ναυτικό μίλι.

## 4.2 Ανάπτυξη Εφαρμογής

Για την ανάπτυξη του γραφικού περιβάλλοντος της εφαρμογής έγινε χρήση της βιβλιοθήκης Dash της γλώσσας python. Η Dash αποτελεί μια open source βιβλιοθήκη η οποία προορίζεται για την κατασκευή custom γραφικού περιβάλλοντος χρήστη. Η επιλογή της προέκυψε λόγω της ενσωμάτωσης τόσο html, όσο και css. Με αυτό τον τρόπο η δημιουργία μιας μοναδικής αλλά και αισθητικά ωραίας διεπαφής είναι εύκολα πραγματοποιήσιμη.

Η επιλογή των components για την είσοδο δεδομένων από τον χρήστη, έγινε με βάση την ευκολία χρήσης, αλλά και την βέλτιστη δομή της εφαρμογής. Η βιβλιοθήκη dash προσφέρει διάφορες επιλογές, όπως slider, textarea, dropdown, checkbox κ.α..

Στην παρακάτω εικόνα βλέπουμε την χρήση html, αλλά και css για την ανάπτυξη της διεπαφής χρήστη – εφαρμογής μέσω της γλώσσας rpython και με τη χρήση της βιβλιοθήκη dash, όπως περιγράψαμε παραπάνω.

```
app.layout = html.Div([
    html.H2('Vessel fuel consumption per Nautical mile prediction (kg/nm)',
        style={'margin-left': '500px', 'margin-bottom': '40px'}),
    html.Div([
        dbc.Label('Vessel\'s Yearbuilt'),
        dcc.Slider(
            min_yb, max_yb, 1,
            marks={i: '{}'.format(i) for i in range(1990, 2023, 2)},
            tooltip={"placement": "bottom", "always_visible": False},
            id='Yearbuilt'
        )
    ],
    style={'width': '75%', 'display': 'inline-block'}),
    html.Hr(),
    html.Div([
        dbc.Label('Vessel\'s Main Engine KW'),
        dcc.Slider(
            min_main_engine_kw, max_main_engine_kw, 1000,
            marks={i: '{}'.format(i) for i in range(1000, 41000, 2000)},
            tooltip={"placement": "bottom", "always_visible": False},
            # value = '20000',
            id='mainenginekw'
        )
    ],
    style={'width': '75%', 'display': 'inline-block'}),
])
```

Εικόνα 26: Dash Library - GUI

Για την εισαγωγή της χρονολογίας κατασκευής του πλοίου επιλέχθηκε η χρήση του slider component. Παρόμοια για την εισαγωγή του Main Engine KW (Εικόνα 27, 28). Η επιλογή των κατώτων και ανώτατων ορίων σε όλα τα πεδία εισαγωγής, έγινε με βάση τις ακραίες τιμές του dataset. Αυτό ισχύει για την μικρότερη, αλλά και μεγαλύτερη χρονολογία κατασκευής του πλοίου, το καθαρό βάρος, το μικτό βάρος, αλλά και τα KW της μηχανής.



Εικόνα 27: Χρονολογία κατασκευής εισαγόμενου πλοίου

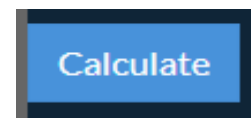


Εικόνα 28: Χαρακτηριστικά μηχανής εισαγόμενου πλοίου

Αντίστοιχα για το deadweight όπως επίσης για το gross rated tonnage επιλέχθηκε η χρήση του Input (Εικόνα 29).

Εικόνα 29: Καθαρό και μικτό βάρος εισαγόμενου πλοίου

Τέλος, η εισαγωγή των δεδομένων και ο υπολογισμός του μέσου όρου κατανάλωσης καυσίμου του πλοίου ανά ναυτικό μίλι, γίνεται τη στιγμή που ο χρήστης πατήσει το κουμπί “Calculate” (Εικόνα 30). Τότε, εμφανίζεται σε έναν πίνακα η έξοδος από το γραμμικό μαθηματικό μοντέλο, όπως επίσης και απο το νευρωνικό δίκτυο.

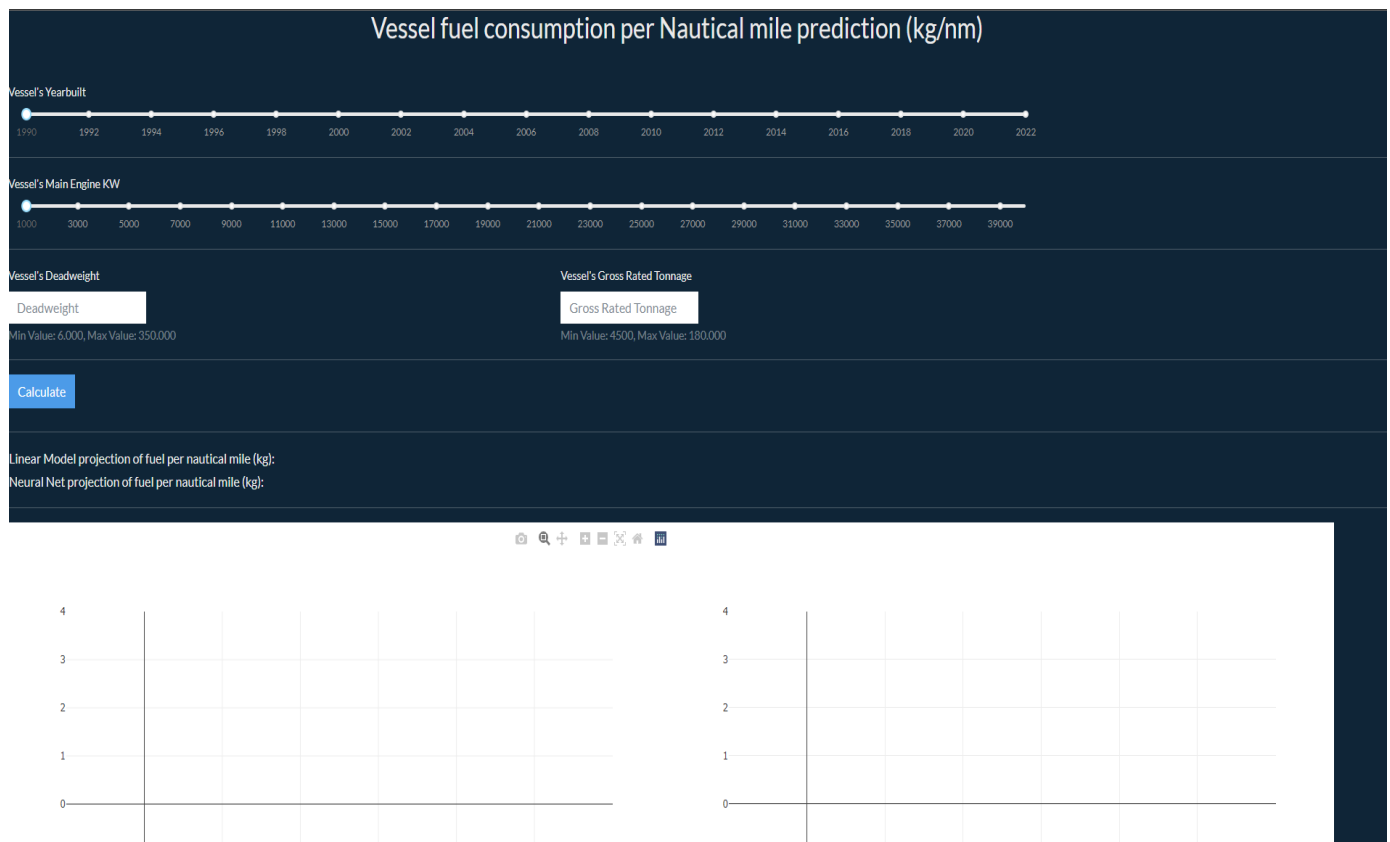


Εικόνα 30: Κουμπί Υπολογισμού

Για την βελτιωμένη οπτικοποίηση των δεδομένων έχουν προστεθεί 2 διαφορετικά γραφήματα (Εικόνα 31). Το ένα γράφημα παρουσιάζει την ετήσια μέση κατανάλωση καυσίμου ανά ναυτικό μίλι για πλοία της ίδιας χρονολογίας με αυτή που εισήγαγε ο χρήστης, αλλά με διαφορετικό deadweight. Στο δεύτερο γράφημα, παρατηρούμε επίσης την ετήσια μέση κατανάλωση καυσίμου ανά ναυτικό μίλι για πλοία του ίδιου deadweight με αυτό που εισήγαγε ο χρήστης, με μια απόκλιση  $\pm 10\%$ . Επιλέχθηκαν το μικτό βάρος που είναι ικανό να μετακινήσει το πλοίο αλλά και το έτος κατασκευής του, δεδομένου ότι αποτελούν τους δύο πιο σημαντικούς παράγοντες στην αύξηση ή μείωση της κατανάλωσης ενός πλοίου.

Η πληροφορία για την συμπλήρωση των γραφημάτων προέρχεται από το ίδιο dataset που χρησιμοποιήθηκε για να εκπαιδευτούν τα μοντέλα πρόβλεψης της εφαρμογής. Αποτελεί επομένως πραγματική πληροφορία με βάση την οποία ο χρήστης μπορεί να αντλήσει συμπεράσματα και να συγκρίνει την προβλεπόμενη κατανάλωση σε αντίθεση

με πραγματικά δεδομένα για πλοία με παρόμοια κατασκευαστικά χαρακτηριστικά. Η τελική μορφή της εφαρμογής φαίνεται στην παρακάτω εικόνα.



*Εικόνα 31: Η διεπαφή της εφαρμογής*

Στη συνέχεια παρατηρούμε την έξοδο της εφαρμογής μετά την είσοδο των στοιχείων ενός πλοίου απο τον χρήστη και την δημιουργία των αντίστοιχων γραφημάτων, σε συνδυασμό με την πρόβλεψη κατανάλωσης καυσίμου απο τα μοντέλα πρόβλεψης (Εικόνα 32). Στο συγκεκριμένο παράδειγμα οι μεταβλητές εισόδου αφορούν πλοίο με βάση τα χαρακτηριστικά:

- Yearbuilt: 2004
- Main Engine KW: 25.000
- Deadweight: 170.000
- Gross Rated Tonnage: 115.000



Λαμβάνοντας στην έξοδο:

- Linear model projection of fuel kg consumption per nautical mile: 291.95
- Neural Network projection of fuel kg consumption per nautical mile: 211.98



Εικόνα 32: Παράδειγμα εισαγωγής πλοίου απο χρήστη

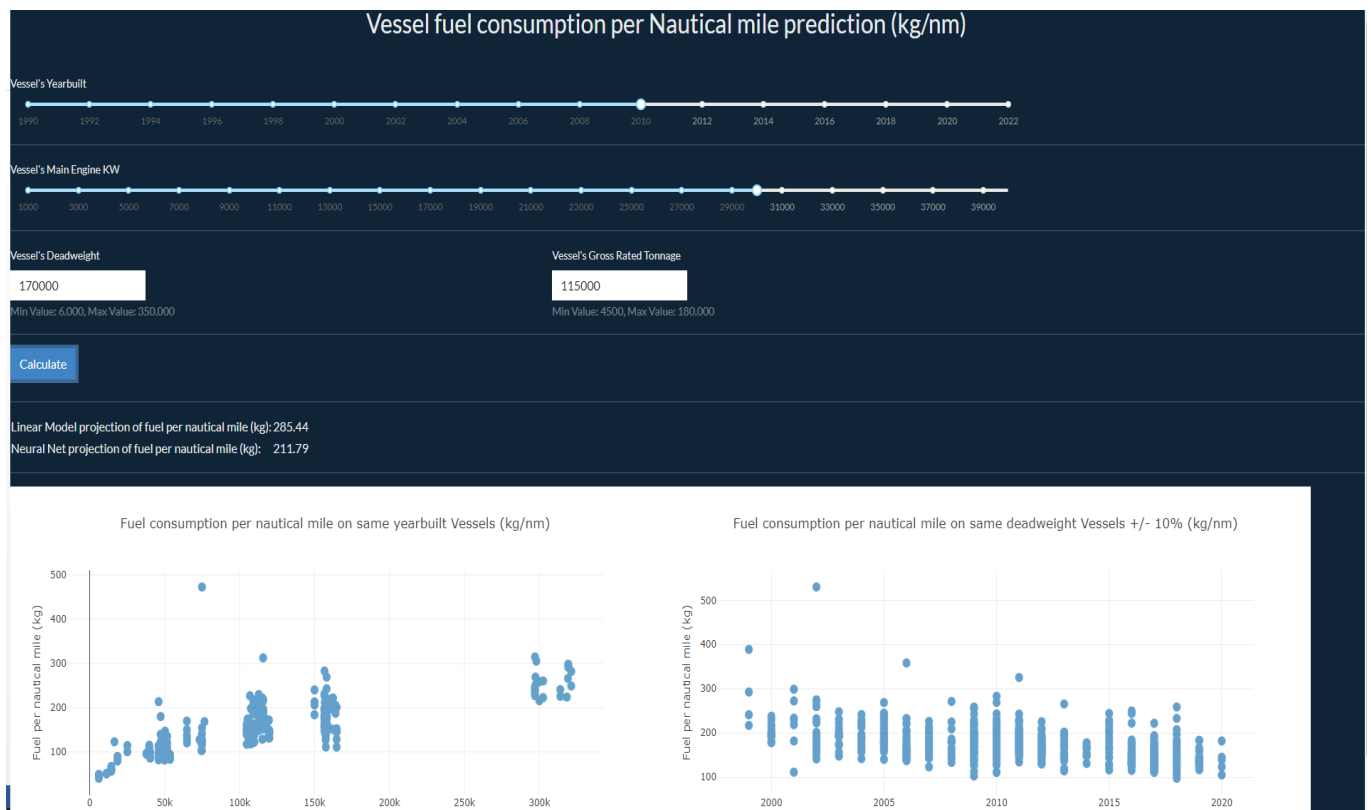
Τέλος, μόλις ο χρήστης αλλάξει οποιαδήποτε μεταβλητή εισόδου, παρατηρούμε την αλλαγή στην έξοδο της εφαρμογής. Με άλλα λόγια, αλλάζει τόσο η πρόβλεψη κατανάλωσης, όσο και η εικόνα των γραφημάτων. Στο παράδειγμα της Εικόνας 33, παρατηρούμε τις εξής πλέον μεταβλητές εισόδου:

- Yearbuilt: 2010
- Main Engine KW: 30.000

- Deadweight: 170.000
- Gross Rated Tonnage: 115.000

Λαμβάνοντας αντίστοιχα στην έξοδο:

- Linear model projection of fuel kg consumption per nautical mile: 285.44
- Neural Network projection of fuel kg consumption per nautical mile: 211.79



Εικόνα 33: Παράδειγμα εισαγωγής πλοίου απο χρήστη – 2

# Κεφάλαιο 5: Σύνοψη και Μελλοντικές Επεκτάσεις

## 5.1 Σύνοψη

Για την εκπόνηση της παρούσας διπλωματικής εργασίας, έγινε προσπάθεια για μια ολοκληρωμένη και ολιστική ανάλυση. Για τον λόγο αυτό, προτού εκπαιδεύσουμε το μοντέλο με δεδομένα, δόθηκε αρκετή σημασία στην προεπεξεργασία του MRV Dataset της Thetis. Για μεγαλύτερη ακρίβεια στην πρόβλεψη των αποτελεσμάτων και αναγνωρίζοντας τις διαφορές που προκύπτουν ανάμεσα στις διαφορετικές κατηγορίες των πλοίων, πάρθηκε η απόφαση για την εστίαση της ερευνητικής προσπάθειας αποκλειστικά σε δεξαμενόπλοια (oil tanker). Στη συνέχεια, μετά απο ανάλυση πάνω στις κατασκευαστικές μετρικές που καθορίζουν το μέγεθος του πλοίου, την δυνατότητα μεταφοράς φορτίου, τις μηχανικές του δυνατότητες κ.α., επιλέχθηκαν οι βέλτιστες μεταβλητές με βάση τις οποίες θα προκύπτει η μέση ετήσια κατανάλωση καυσίμου ανά ναυτικό μίλι για το κάθε εισαγόμενο πλοίο.

Η ανάλυση και η σύγκριση διαφορετικών αλγορίθμων, οι οποίοι εκπαιδεύτηκαν με τα ίδια προεπεξεργασμένα δεδομένα καθόρισε την τελική μορφή της εφαρμογής. Επιλέχθηκαν η γραμμική παλινδρόμηση, λόγω της γραμμικής συσχέτισης που παρατηρήθηκε μεταξύ των δομικών χαρακτηριστικών του πλοίου, σε σχέση με την τελική κατανάλωση καυσίμου. Στη συνέχεια επιλέχθηκε η χρήση νευρωνικού δικτύου, προκειμένου να εκμεταλλευτούμε την προσαρμοστικότητα που προσφέρει πάνω στα δεδομένα εισόδου. Για την υλοποίηση του νευρωνικού δικτύου, έγινε η επιλογή μικρού αριθμού νευρώνων στα ενδιάμεσα στάδια, λόγω των περιορισμένων δεδομένων εκπαίδευσης. Ωστόσο, παρατηρήθηκε πολύ καλή προσαρμοστικότητα των προβλεπόμενων τιμών του δικτύου σε σχέση με τις πραγματικές τιμές κατανάλωσης που δίνονταν απο το dataset το οποίο χρησιμοποιήθηκε για την εκπόνηση της εργασίας. Τόσο η αποτελεσματικότητα του γραμμικού μοντέλου, όσο και του νευρωνικού δικτύου θα αυξανόταν εκθετικά με την αύξηση των δεδομένων εισόδου. Τα δεδομένα καταναλώσεων των εμπορικών πλοίων ωστόσο, αποτελούν ευαίσθητα δεδομένα και η πρόσβαση σε αυτά δεν είναι εύκολη.

Σημαντικό κομμάτι της ανάλυσης αποτελεί και η σωστή έκθεση των αποτελεσμάτων. Για την εύκολη εισαγωγή δεδομένων απο τον χρήστη και την βέλτιστη παρουσίαση των αποτελεσμάτων των μοντέλων πρόβλεψης, δημιουργήθηκε η διεπαφή της εφαρμογής. Μέσα σε αυτή με σκοπό να εκτεθούν τα πραγματικά δεδομένα με βάση τα οποία έγινε η εκπαίδευση των μοντέλων, προστεθήκαν δύο γραφήματα που παρουσιάζουν τις

καταναλώσεις πλοίων με παρόμοια δομικά χαρακτηριστικά με αυτά που εισήγαγε ο χρήστης.

## 5.2 Μελλοντικές Επεκτάσεις

Βασικός στόχος μιας μελλοντικής επέκτασης της παρούσας διπλωματικής, είναι η αύξηση ακρίβειας των προβλέψεων του μοντέλου. Παράλληλα, σημαντική προσθήκη στη ήδη υπάρχουσα εργασία, θα αποτελούσε η ανάπτυξη λογισμικού για την σύνδεση των μοντέλων προβλέψεων με εφαρμογές. Πιο συγκεκριμένα:

- Με τη χρήση νέων δεδομένων εκπαίδευσης, η έξοδος των μοντέλων θα βελτιωθεί και η προβλεπόμενη κατανάλωση θα είναι ακόμα πιο κοντά στη πραγματικότητα. Για την εύρεση των δεδομένων, μια σημαντική πηγή πληροφοριών, θα μπορούσε να είναι τα noon reports ταξιδιών από ένα σύνολο πλοίων. Σε αυτά, βρίσκεται η ακριβής κατανάλωση καυσίμων, την οποία έχει κάνει ένα πλοίο μέσα στη διάρκεια ενός ταξιδιού, όπως επίσης και η ακριβής απόσταση που έχει διανύσει.
- Μέσω των παραπάνω reports, προτείνεται η διερεύνηση άλλων παραγόντων στον υπολογισμό της κατανάλωσης καυσίμου, πέραν των δομικών χαρακτηριστικών των πλοίων που λήφθηκαν υπ όψιν σε αυτή την εργασία, με σκοπό την εξαγωγή καλύτερων συμπερασμάτων. Για παράδειγμα, σημαντικό κομμάτι της κατανάλωσης καυσίμου σε ένα ταξίδι αποτελούν οι καιρικές συνθήκες. Με τα σωστά δεδομένα λοιπόν απο τα καθημερινά reports των πλοίων, στα οποία καταγράφονται, η ακριβής κατανάλωση και οι καιρικές συνθήκες, προτείνεται η προσθήκη περαιτέρω μεταβλητών, όπως τα καιρικά φαινόμενα που αναμένεται να αντιμετωπίσει ένα πλοίο. Σε αυτή τη περίπτωση, η πρόβλεψη κατανάλωσης που θα εξάγει το μοντέλο, θα αναφέρεται στο εκάστοτε ταξίδι και όχι στον ετήσιο μέσο όρο. Συνοψίζοντας, με την εισαγωγή των δομικών χαρακτηριστικών του πλοίου, με βάση τα οποία στη παρούσα εργασία έχουμε υπολογίσει τον ετήσιο μέσο όρο και με τη προσθήκη μεταβλητών οι οποίες διαφέρουν για κάθε ταξίδι, όπως κατεύθυνση ανέμου, ύψος κυμάτων, κ.α., μπορεί να δημιουργηθεί ένα ακριβέστερο μοντέλο το οποίο θα προσαρμόζεται για το εκάστοτε ταξίδι του πλοίου.
- Τέλος, προτείνεται η δημιουργία μιας διεπαφής προγραμματισμού εφαρμογών (application programming interface - API) για την χρησιμοποίηση του μοντέλου πρόβλεψης κατανάλωσης, απο οποιοδήποτε πρόγραμμα με τη χρήση requests και responses.

---

## Βιβλιογραφία

- [1] "<https://www.marineinsight.com>," [Online]. Available: <https://www.marineinsight.com/guidelines/a-guide-to-types-of-ships/>.
- [2] [Online]. Available: <https://www.marineinsight.com/guidelines/what-is-noon-report-on-ships/>.
- [3] T.-H. Joung, «The IMO initial strategy for reducing Greenhouse Gas (GHG) emissions, and its follow-up actions towards 2050,» *Journal of International Maritime Safety, Environmental Affairs, and Shipping*, αρ. 4.1, pp. 1-7, 2020.
- [4] S. B. a. J. V. L. Cooper, Alan Turing: His work and impact, Elsevier, 2013.
- [5] M. a. A. K. Haenlein, A brief history of artificial intelligence: On the past, present, and future of artificial intelligence, California management, 2019.
- [6] B. G. a. E. A. F. Buchanan, "DENDRAL and Meta-DENDRAL: Their applications dimension," *Artificial intelligence*, no. 11.1-2, pp. 5-24., 1978.
- [7] P. J. Werbos, «Backpropagation through time: what it does and how to do it,» *Proceedings of the IEEE*, αρ. 78, pp. 1550-1560, 1990.
- [8] P. J. Heuristics, Intelligent Search Strategies for Computer Problem Solving, 1984.
- [9] B. G. Buchanan, "A (very) brief history of artificial intelligence," vol. 26.4, no. Ai Magazine, pp. 53-53, 2005.
- [10] F.-h. Hsu, "IBM's deep blue chess grandmaster chips," no. IEEE micro 19.2 (1999): 70-81.
- [11] L. a. M. C. Floridi, "GPT-3: Its nature, scope, limits, and consequences., " vol. 30.4, no. Minds and Machines , pp. 681-694, 2020.
- [12] R. Dale, "GPT-3: What's it good for," vol. 27.1, no. Natural Language Engineering, pp. 113-118, 2021.
- [13] E. Murati, " Language & Coding Creativity., " *Daedalus* , vol. 151 , no. (2), p. 156–167, 2022.
- [14] K. Warwick, Artificial intelligence: the basics, Routledge, 2013.
- [15] C. Fu, "Link weight prediction using supervised learning methods and its application to yelp layered network," *IEEE Transactions on Knowledge and Data Engineering*, no. 30.8, pp. 1507-1518, 2018.
- [16] S. Jun, «A forecasting model for technological trend using unsupervised learning,» *Bio-Science and Bio-Technology*, pp. 51-60, 2011.

- [17] W. a. Z. Z. Qiang, «Reinforcement learning model, algorithms and its application,» *International Conference on Mechatronic Science, Electric Engineering and Computer (MEC)*, 2011.
- [18] E. E. W. S. a. J. N. Nunez, "Regression modeling strategies," *Revista Española de Cardiología (English Edition)*, no. 64.6, pp. 501-507., 2011.
- [19] W. a. L. L. Yao, "A new regression model: modal linear regression., " *Scandinavian Journal of Statistics*, no. 41.3, pp. 656-671, 2014.
- [20] L. N. a. N. T. Smith, *Deep convolutional neural network design patterns*, 2016.
- [21] D. V. K. a. J. P. Svozil, "Introduction to multi-layer feed-forward neural networks," *Chemometrics and intelligent laboratory systems*, no. 39.1, pp. 43-62, 1997.
- [22] S. Ruder, *An overview of gradient descent optimization algorithms*, 2016.
- [23] L. Bottou, "Stochastic gradient learning in neural networks," *Proceedings of Neuro-Nimes* , vol. 12, no. 91.8, 1991.
- [24] D. P. a. J. B. Kingma, *Adam: A method for stochastic optimization*, 2014.
- [25] A. Chircop, «The international maritime organization,» *The Oxford Handbook of the Law of the Sea*, 2015.
- [26] Y. Benjamini, «Opening the box of a boxplot,» *The American Statistician*, ap. 42, pp. 257-262, 1988.
- [27] "Why 70/30 or 80/20 Relation Between Training and Testing Sets," 2018.