ΑΛΓΟΡΙΘΜΟΙ ΤΕΧΝΗΤΗΣ ΝΟΗΜΟΣΥΝΗΣ ΣΤΗ ΝΑΥΤΙΛΙΑ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Του

ΕΥΡΙΠΙΔΗ ΓΕΡΑΣΙΜΟΥ

Σχολή Μηχανικών

Τμήμα Μηχανικών Βιομηχανικής Σχεδίασης και Παραγωγής

Επιβλέπουσα: Ελένη Αικατερίνη Λελίγκου

Συνεπιβλέπων: Ευθύμιος Παριώτης Αν. Καθηγητής Σχολής Ναυτικών Δοκίμων

Αθήνα, Ιούλιος 2021

ΜΕΛΗ ΕΞΕΤΑΣΤΙΚΗΣ ΕΠΙΤΡΟΠΗΣ

Επιβλέπουσα Καθηγήτρια: Ελένη Αικατερίνη Λελίγκου	Eleni Aikaterini Leligkou Digitally signed by Eleni Aikaterini Leligkou Date: 2021.08.03 12:19:42 +03'00'
Αναπληρωτής Καθηγητής ΣΝΔ-Συνεπιβλέπων: Ευθύμιος Παριώτης	
Επ. Καθηγήτρια: Παρασκευή Ζαχαρία	

(Η σελίδα αυτή είναι σκοπίμως κενή)

ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΔΙΠΛΩΜΑΤΙΚΗΣ ΕΡΓΑΣΙΑΣ

Ο κάτωθι υπογεγραμμένος Ευριπίδης Γεράσιμος του Γεωργίου, με αριθμό μητρώου 71446180 φοιτητής του Πανεπιστημίου Δυτικής Αττικής της Σχολής Μηχανικών του Τμήματος Βιομηχανικής Σχεδίασης και Παραγωγής, δηλώνω υπεύθυνα ότι:

«Είμαι συγγραφέας αυτής της διπλωματικής εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, οι όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών ή λέξεων, είτε ακριβώς είτε παραφρασμένες, αναφέρονται στο σύνολό τους, με πλήρη αναφορά στους συγγραφείς, τον εκδοτικό οίκο ή το περιοδικό, συμπεριλαμβανομένων και των πηγών που ενδεχομένως χρησιμοποιήθηκαν από το διαδίκτυο. Επίσης, βεβαιώνω ότι αυτή η εργασία έχει συγγραφεί από μένα αποκλειστικά και αποτελεί προϊόν πνευματικής ιδιοκτησίας τόσο δικής μου, όσο και του Ιδρύματος.

Παράβαση της ανωτέρω ακαδημαϊκής μου ευθύνης αποτελεί ουσιώδη λόγο για την ανάκληση του πτυχίου μου».

Ο Δηλών

Ευριπίδης Γεράσιμος

Ευχαριστίες

Ευχαριστώ θερμά την οικογένεια μου, του φίλους μου και συμφοιτητές μου Αγγελική και Σπύρο για την υποστήριξη τους σε όλο το διάστημα των σπουδών μου. Τέλος, θέλω να ευχαριστήσω την υπεύθυνη καθηγήτρια μου κ. Ελένη Αικατερίνη Λελίγκου για την πολύτιμη βοήθεια και καθοδήγησή της στην ανάπτυξη της διπλωματικής εργασίας.

Περίληψη

Σε αυτήν την διπλωματική εργασία μελετάται η εφαρμογή μηχανισμών τεχνητής νοημοσύνης στην μοντελοποίηση της λειτουργίας του πλοίου. Η έμφαση δίνεται στην κατάλληλη επιλογή παραμέτρων που οι αλγόριθμοι θα λάβουν υπ' όψη τους ώστε να επιτευχθεί η μεγαλύτερη δυνατή ακρίβεια πρόβλεψης λειτουργίας αυτής.

Λέξεις κλειδιά

Υπερπαράμετροι, Εξόρυξη Δεδομένων, Μηχανική Μάθηση, Δεδομένα, RapidMiner, Python.

Abstract

The topic of this dissertation is the application of artificial intelligence mechanisms in the modeling of ship operation. Emphasis is placed in the appropriate selection of the hyperparameters that the algorithms will take into account in order to achieve the highest possible accuracy of its operation prediction.

Key Words

Hyperparameter, Data Mining, Machine Learning, Data, RapidMiner, Python.

Πίνακας Περιεχομένων

(Η σελίδα αυτή είναι σκοπίμως κενή)

Α) Μέρος Πρώτο:

Κεφάλαιο 1: Εισαγωγή

1.1 Το αντικείμενο της Διπλωματικής Εργασίας

Το αντικείμενο της διπλωματικής εργασίας είναι, ο ευρύτερος τομέας της μηχανικής μάθησης (Machine Learning), με την οποία δημιουργούνται, εκπαιδεύονται και επεξεργάζονται διάφορα μοντέλα. Τα δεδομένα που χρησιμοποιούνται από αυτά τα μοντέλα συλλέγονται και προεπεξεργάζονται (Data Preprocessing) με την βοήθεια της εξόρυξης δεδομένων (Data mining).

1.2 Η ενασχόληση της Διπλωματικής Εργασίας

Η παρούσα διπλωματική εργασίας έχει ως ενασχόληση αρχικά, την παρουσίαση μιας μεθόδου για την μείωση των δεδομένων κατά την προεπεξεργασία. Στην συνέχεια, έχει ως στόχο την δημιουργία διάφορων μοντέλων και την επεξεργασία των υπερπαραμέτρων τους με σκοπό να δημιουργηθεί το πιο ιδανικό μοντέλο για την βάση δεδομένων που χρησιμοποιήθηκε.

1.3 Δομή της Διπλωματικής Εργασίας

Το πρώτο μέρος της εργασίας (Κεφάλαια 1 έως και 5) το οποίο εμπεριέχει μια θεωρητική μελέτη μαζί με την επεξήγηση των επιμέρους εννοιών και μεθόδων υλοποίησης των λειτουργιών που χρησιμοποιούνται στην επίλυση προβλημάτων τεχνητής νοημοσύνης.

Το δεύτερο μέρος της εργασίας (Κεφάλαια 6 έως και 8) διεξάγετε μια πειραματική έρευνα που με την βοήθεια της εξόρυξη των δεδομένων, της μηχανική μάθησης και ορισμένων εργαλείων προγραμματισμού γίνεται η προεπεξεργασία και η μοντελοποίηση της επιλεγμένης βάσης δεδομένων.

Το τρίτο μέρος της εργασίας έχει ως στόχο τα προσωπικά συμπεράσματα, τις συγκρίσεις των μοντέλων με σκοπό την εύρεση του καλύτερου και οι δυσκολίες που αντιμετωπίστηκαν. Επίσης, στο τρίτο μέρος ανήκει η βιβλιογραφία και οι ιστοσελίδες των προγραμμάτων.

Κεφάλαιο 2: Data Mining – Εξόρυξη Δεδομένων

2.1 Ορισμός του Data Mining

Η ανάγκη της κατανόησης για μεγάλες, πολύπλοκες και πλούσιες σε πληροφορίες βάσεις δεδομένων είναι κοινός στόχος σχεδόν όλων των τμημάτων των επιχειρήσεων, της επιστήμης όπως και της μηχανικής. Στον κόσμο των επιχειρήσεων, τα δεδομένα τα εταιρικά και των πελατών είναι κοινώς αναγνωρισμένα ως στρατηγικό πλεονέκτημα για την κάθε εταιρία. Η ικανότητα για την εξαγωγή σημαντικής πληροφορίας κρυμμένης σε αυτά τα δεδομένα και η χρησιμοποίηση της γίνεται εξαιρετικά απαραίτητη στην σημερινή εποχή για τον ανταγωνισμό της κάθε επιχείρησης. Όλη η διαδικασία με την βοήθεια που παρέχετε από διάφορες υπολογιστικές μεθοδολογίες και τεχνικές για την ανακάλυψη σημαντικής πληροφορίας μέσα σε αυτά τα ακατέργαστα δεδομένα, ονομάζεται Εξόρυξη Δεδομένων (Data Mining).

2.2 Είδη δεδομένων στα οποία μπορεί να γίνει εξόρυξη

Η εξόρυξη δεδομένων μπορεί να εφαρμοστεί σε οποιοδήποτε είδος δεδομένων εφόσον αυτά τα δεδομένα έχουν κάποια αξία για μια εφαρμογή. Οι πιο βασικές μορφές δεδομένων για το Data Mining είναι δεδομένα από βάσεις δεδομένων (Database 2.2.1), δεδομένα από αποθήκες δεδομένων (Warehouse Data 2.2.2) και δεδομένα συναλλαγών (Transactional Data 2.2.3).

2.2.1 Βάσεις δεδομένων – Database

Μια βάση δεδομένων είναι μια οργανωμένη συλλογή δομημένων δεδομένων ή πληροφοριών. Η οποία ελέγχεται συνήθως από ένα σύστημα διαχείρισης βάσεων δεδομένων (DBMS «Database Management System»). Τα δεδομένα, το DBMS μαζί και οι εφαρμογές που σχετίζονται σε αυτά αναφέρονται ως ένα σύστημα βάσης δεδομένων.[3]

Στους πιο αναγνωρισμένους τύπους Database τα δεδομένα είναι συνήθως μοντελοποιημένα σε σειρές και στήλες πάνω σε μια σειρά πινάκων, ώστε να μπορούν να κάνουν την αναζήτηση και την επεξεργασία των δεδομένων πιο εύχρηστη και αποτελεσματική.

2.2.2 Αποθήκες δεδομένων – Warehouse Data

Η αποθήκη δεδομένων είναι ένα αποθετήριο πληροφοριών οι οποίες συλλέγονται από πολλές πηγές, έπειτα, αποθηκεύονται σε ένα ενοποιημένο σχήμα και συνήθως διαμένουν σε έναν μόνο ιστότοπο. Οι αποθήκες δεδομένων είναι κατασκευασμένες μέσα από μια διαδικασία καθορισμού, ολοκλήρωσης, μετασχηματισμού, φόρτωσης και περιοδικής ανανέωσης δεδομένων. Η μοντελοποίηση του Warehouse Data πραγματοποιείται συνήθως από μια πολυδιάστατη δομή δεδομένων, η οποία ονομάζεται κύβος δεδομένων (data cube).

2.2.3 Δεδομένα Συναλλαγών - Transactional Data

Η κάθε εγγραφή σε μια βάση δεδομένων συναλλαγών καταγράφει μια συναλλαγή. Αυτή η συναλλαγή τις περισσότερες φορές περιλαμβάνει έναν μοναδικό αριθμό ταυτότητας συναλλαγής (trans ID) και την λίστα με τα στοιχεία που αποτελούν αυτήν την συναλλαγή. Μια βάση δεδομένων συναλλαγών μπορεί να έχει ακόμα και επιπρόσθετους πίνακες, οι οποίοι περιέχουν άλλες πληροφορίες που σχετίζονται με τις συναλλαγές, όπως η περιγραφή διάφορων στοιχείων, δηλαδή πληροφορίες σχετικά με τον πωλητή ή το υποκατάστημα και ούτω καθεξής.

2.3 Η διαδικασία της εξόρυξης δεδομένων

Η πειραματική διαδικασία στην εξόρυξη δεδομένων είναι η ακόλουθη:

1. **Καθορισμός του προβλήματος και δημιουργία υποθετικών λύσεων** (State the problem and formulate the hypothesis): Οι περισσότερες μελέτες μοντελοποίησης βάσεων δεδομένων, εκτελούνται σε συγκεκριμένους τομείς εφαρμογών. Άρα για να υπάρξει το επιθυμητό αποτέλεσμα, θα να καθοριστεί το πρόβλημα με μεγάλη ακρίβεια, ώστε να μπορούν να δημιουργηθούν αρκετές υποθετικές λύσεις για την συνέχεια της διαδικασίας.
2. **Συλλογή των δεδομένων** (Collect the data): Υπάρχουν δυο διαφορετικοί τρόποι συλλογής δεδομένων. Ο πρώτος τρόπος είναι όταν η διαδικασία επιλογής των δεδομένων βρίσκεται υπό τον έλεγχο ενός ειδικού (μοντελοποιητή), άρα, τα δεδομένα που θα χρησιμοποιηθούν είναι ελεγχόμενα. Αυτή η προσέγγιση ονομάζεται και designed experiment. Ο δεύτερος τρόπος είναι όταν ο ειδικός δεν μπορεί να επηρεάσει τη διαδικασία οπότε είναι απλά ένας παρατηρητής. Αυτή η προσέγγιση ονομάζεται observational approach. Αυτή η διαδικασία είναι πολύ σημαντική για την εξόρυξη των δεδομένων, καθώς, ο δημιουργός του μοντέλου θα μπορεί να έχει πιο εύχρηστη διαχείριση στην δοκιμή και στην δημιουργία του μοντέλου.

3. **Προετοιμασία των δεδομένων** (Preprocessing Data): Αφού γίνει η συλλογή των δεδομένων θα πρέπει να γίνει μια προετοιμασία, καθώς, τα δεδομένα συνήθως δεν είναι όλα ελεγχόμενα. Οπότε, για να μην υπάρχει λάθος στην λειτουργεία και μετέπειτα στο αποτέλεσμα που θα δώσει το μοντέλο, θα πρέπει να ελεγχθεί για σφάλματα στις τιμές και να γίνει σωστή διαμόρφωση και επιλογή στις μεταβλητές που θα χρησιμοποιηθούν.
4. **Επιλογή του μοντέλου** (Estimate the model): Η επιλογή του μοντέλου είναι το σημαντικότερο κομμάτι της διαδικασίας, επειδή, σε αυτήν την επιλογή βασίζεται το σημαντικότερο μέρος του τελικού αποτελέσματος. Η διαδικασία της επιλογής δεν είναι εύκολη, καθώς, η εφαρμογή μπορεί να βασιστεί σε διάφορα μοντέλα και πρέπει να γίνει η επιλογή του καλύτερου από αυτά.
5. **Δημιουργία του μοντέλου και συμπεράσματα** (Interpret the model and draw conclusions): Η δημιουργία του μοντέλου εξαρτάται στον μεγαλύτερο βαθμό από την ακρίβεια των αποτελεσμάτων που έχουν τεθεί σαν στόχο. Κατά αυτόν τον τρόπο, ακολουθεί η διαδοχή πως αν ο στόχος χρειάζεται μεγάλη ακρίβεια τότε το μοντέλο θα είναι αρκετά περίπλοκο. Συμπερασματικά, αν η ακρίβεια των αποτελεσμάτων του μοντέλου είναι κοντά σε αυτήν του στόχου τότε έχει γίνει σωστή επιλογή του μοντέλου.

2.4 Σημαντικά ζητήματα που αντιμετωπίζει το Data Mining

Η εξόρυξη δεδομένων είναι ένας άκρως αναπτυσσόμενος κλάδος με μεγάλες δυνατότητες. Επομένως, πρέπει να αντιμετωπίζει ορισμένα ζητήματα. Μερικά από τα πιο κύρια ζητήματα στην εξόρυξη δεδομένων είναι τα εξής:

1. **Η μεθοδολογία της εξόρυξης** (mining methodology): Οι μεθοδολογίες αυτές είναι η διερεύνηση νέων ειδών γνώσεων, η εξόρυξη σε πολυδιάστατο χώρο και η ενσωμάτωση των μεθόδων από διάφορους κλάδους. Ορισμένες από αυτές τις μεθόδους διερευνούν τον τρόπο με τον οποίο μπορούν να χρησιμοποιηθούν συγκεκριμένα μέτρα και παράμετροι ώστε να μπορεί να εκτιμηθεί η σημαντικότητα των ανακαλυφθέντων μοτίβων.
2. **Η αλληλεπίδραση των χρηστών** (user interaction): Ο χρήστης έχει από τους πιο σημαντικούς ρόλους στην διαδικασία της εξόρυξης δεδομένων. Καθώς, αρκετά πεδία μιας έρευνας περιλαμβάνουν τον τρόπο αλληλεπίδρασης με ένα σύστημα εξόρυξης δεδομένων, τον τρόπο ενσωμάτωσης των βασικών γνώσεων ενός χρήστη στην εξόρυξη και τον τρόπο οπτικοποίησης και κατανόησης των αποτελεσμάτων του data mining.
3. **Αποτελεσματικότητα και Επεκτασιμότητα** (efficiency and scalability): Αυτή η μέθοδος χρησιμοποιείται κατά τη σύγκριση των αλγορίθμων της εξόρυξης δεδομένων. Οι αλγόριθμοι πρέπει να είναι αποτελεσματικοί και επεκτάσιμοι προκειμένου να εξάγουν αποτελεσματικές πληροφορίες από τεράστιες ποσότητες δεδομένων σε πολλές αποθήκες δεδομένων ή σε δυναμικές ροές δεδομένων.

4. **Ποικιλία από τύπους δεδομένων** (diversity of data types): Υπάρχει πάρα πολύ μεγάλη ποικιλία από διαφορετικούς τύπους, γεγονός που δημιουργεί αρκετές προκλήσεις στην εξόρυξη δεδομένων.
5. **Εξόρυξη δεδομένων και κοινωνία** (data mining and society): Με το data mining να διεισδύει όλο και περισσότερο στην καθημερινή ζωή του ανθρώπου, είναι σημαντικό να γίνει μελέτη στο αντίκτυπο που έχει στην κοινωνία. Όπως είναι, η ακατάλληλη αποκάλυψη ή χρήση δεδομένων και η πιθανή παραβίαση των ατομικών δικαιωμάτων προστασίας προσωπικών δεδομένων.

Τέλος, αρκετά από αυτά τα ζητήματα έχουν αντιμετωπιστεί σε αρκετό βαθμό, παρόλα αυτά συνεχίζουν να ενθαρρύνουν για περαιτέρω έρευνα και βελτίωση στην εξόρυξη των δεδομένων.

Κεφάλαιο 3: Μηχανική Μάθηση – Machine Learning

3.1 Τι ορίζουμε ως μάθηση

Ως μάθηση (learning) ορίζουμε τη διαδικασία βελτίωσης της επίδοσης ενός συστήματος σε μια συγκεκριμένη εργασία μετά από την παρατήρηση πολλών παραδειγμάτων. Για να υπάρξει μια μάθηση απαιτούνται τρία βασικά συστατικά: [4]

1. Ένα περιβάλλον το οποίο να προσφέρει δεδομένα υπό μορφή παραδειγμάτων στο σύστημα.
2. Ένα κριτήριο αξιολόγησης της επίδοσης του συστήματος.
3. Μια συγκεκριμένη εργασία την οποία το σύστημα καλείται να εκτελέσει.

3.1.1 Η μάθηση στη ζωή του ανθρώπου

Ο άνθρωπος εφαρμόζει διαδικασίες μάθησης εκούσια ή ακούσια καθ' όλη τη διάρκεια της ζωής του. Όπως για παράδειγμα, ένα παιδί μαθαίνει τη μητρική του γλώσσα ακούγοντας τις ομιλίες των γονιών του και των άλλων ανθρώπων στο περιβάλλον του.

Για τους ανθρώπους, η αναγνώριση αντικειμένων ή συμβόλων είναι μια εργασία ρουτίνας, την οποία επαναλαμβάνουμε συχνά χωρίς να δίνουμε ιδιαίτερη σημασία σε αυτή. Για παράδειγμα, ο άνθρωπος μπορεί να διαβάσει ένα χειρόγραφο κείμενο, ακόμη και αν ο γραφικός χαρακτήρας του συγγραφέα του είναι άγνωστος. [4]

3.2 Τι είναι η Μηχανική μάθηση

Η μηχανική μάθηση είναι ένας τομέας της Τεχνητής Νοημοσύνης που ασχολείται με την ανάπτυξη αλγορίθμων μάθησης, δηλαδή αλγορίθμων που βελτιώνουν την επίδοση ενός συστήματος σε διάφορα προβλήματα. Για να επιτευχθεί η βελτίωση, δίνεται στον αλγόριθμο ένα ικανό πλήθος παραδειγμάτων ώστε να μπορέσει να εκπαιδευτεί. Η βελτίωση συνήθως γίνεται σταδιακά επειδή ο αλγόριθμος στις περισσότερες περιπτώσεις είναι επαναληπτικός, καθώς, εξετάζει τα παραδείγματα πολλές φορές σε διάφορες «εποχές μάθησης». [4]

3.3 Στόχος και αξία της Μηχανικής Μάθησης

3.3.1 Στόχος

Η διαδικασία της μάθησης δε μπορεί να υλοποιηθεί με απλή απομνημόνευση όλων των πιθανών εκδοχών ενός αντικειμένου. Για παράδειγμα, δεν είναι πρακτικά εφικτό για τον άνθρωπο να δημιουργήσει μια βάση δεδομένων με φωτογραφίες από ένα αντικείμενο και να περιμένει να αναγνωρίσει το αντικείμενο αυτό εκτελώντας απλώς μια αναζήτηση σε αυτή τη βάση δεδομένων. Είναι σχεδόν βέβαιο ότι η συγκεκριμένη εικόνα δε θα βρεθεί, καθώς υπάρχει αυτό το αντικείμενο σε απεριόριστα χρώματα, σχήματα, σχέδια και διαστάσεις. Από το παραπάνω παράδειγμα γίνεται προφανές πως ο στόχος της μάθησης δεν είναι η συμβατική απομνημόνευση των δεδομένων, που μπορεί να επιτευχθεί με τη χρήση ενός απλού πίνακα αποθήκευσης. Αντιθέτως, ο στόχος της μάθησης είναι **η δυνατότητα παραγωγής σωστών εκτιμήσεων** σχετικά με δεδομένα τα οποία αντιμετωπίζονται για πρώτη φορά από το σύστημα. [4]

3.3.2 Αξία

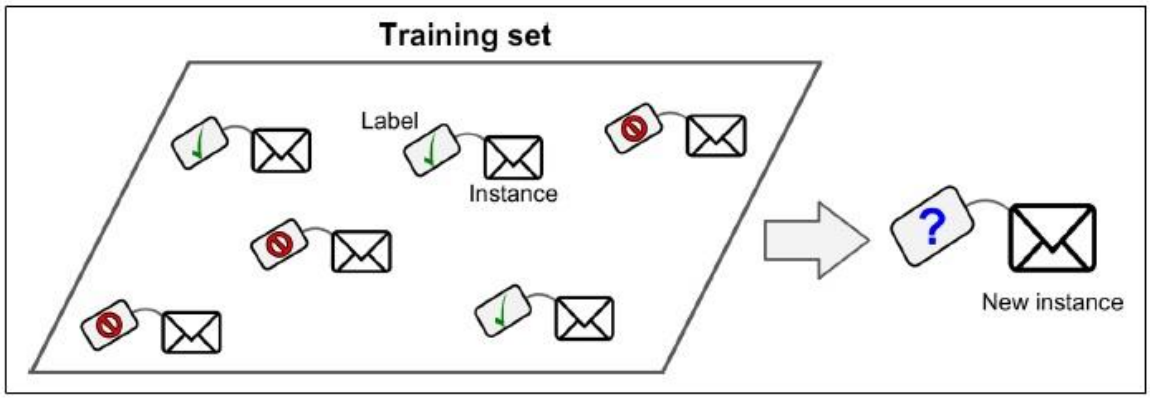
Η αξία της μηχανικής μάθησης εντοπίζεται στην παραπάνω ιδιότητα. Σε πάρα πολλές εφαρμογές είναι πολύ σημαντικό ο χρήστης να μπορεί να εκτιμήσει την κατηγορία στην οποία ανήκουν άγνωστα μέχρι στιγμής δεδομένα ή να προβλέψει την αξία στην οποία αντιστοιχούν τα δεδομένα χωρίς να τα έχει ξαναδεί. Για να μπορέσει το σύστημα να ανταπεξέλθει στην παραπάνω απαίτηση, είναι απαραίτητη η παρουσίαση παρόμοιων αντικειμένων ή δεδομένων με τέτοιον τρόπο ώστε να αποκαλύπτεται η κρυμμένη σχέση μεταξύ των μεταβλητών. [4]

3.4 Τύποι Συστημάτων Μηχανικής Μάθησης

Οι πιο βασικοί τύποι συστημάτων μηχανικής μάθησης είναι τρεις: η μάθηση με επίβλεψη (Supervised Learning 3.4.1), μάθηση χωρίς επίβλεψη (Unsupervised Learning 3.4.2) και μάθηση με ενίσχυση (Reinforcement learning 3.4.3).

3.4.1 Μάθηση με επίβλεψη (Supervised Learning)

Στην μάθηση με επίβλεψη, τα δεδομένα εκπαίδευσης που τροφοδοτούνται στον αλγόριθμο περιλαμβάνουν τις επιθυμητές λύσεις, οι οποίες ονομάζονται labels.



Εικόνα 3.1: Επιλογή επιθυμητών λύσεων. [5]

Μια τυπική εργασία της μάθησης με επίβλεψη είναι **η κατηγοριοποίηση** (classification). Ένα παράδειγμα είναι, το φίλτρο των ανεπιθύμητων στο email. Είναι εκπαιδευμένο με πάρα πολλά παραδείγματα από emails μαζί με την κατηγορία τους (spam ή ham). Με αυτόν τον τρόπο μπορεί να μάθει να ξεχωρίζει τα καινούργια email αν ανήκουν σε αυτές τις κατηγορίες.

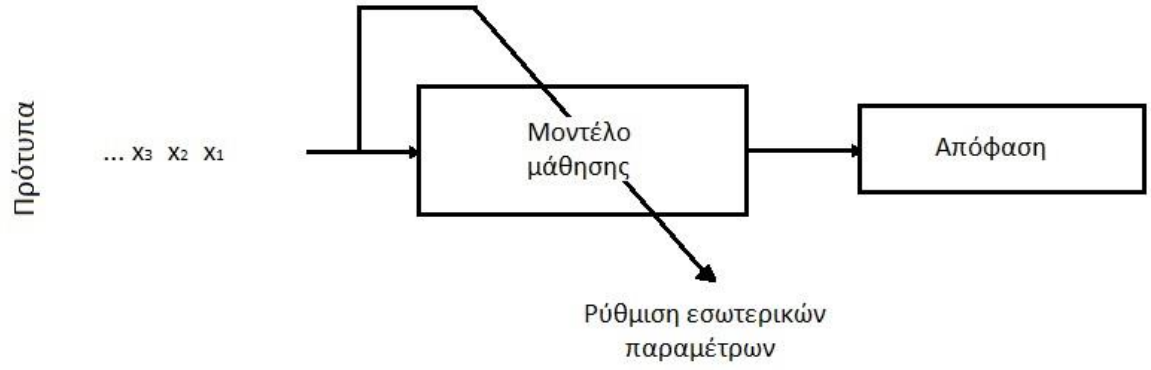
Μια ακόμα τυπική εργασία του συστήματος αυτού είναι **η πρόβλεψη μιας αριθμητικής τιμής σαν στόχος**, όπως είναι η τιμή της αξίας ενός αυτοκινήτου. Για να εκπαιδευτεί αυτό το σύστημα χρειάζεται να λάβει πολλά δεδομένα σαν παραδείγματα από διάφορα αυτοκίνητα μαζί με τα χαρακτηριστικά τους (όπως είναι τα κυβικά, η ηλικία, η μάρκα). Αυτή η διαδικασία ονομάζεται παλινδρόμηση (regression).

Μερικοί από τους πιο σημαντικούς αλγορίθμους μάθησης με επίβλεψη είναι οι εξής:

- k – Nearest Neighbors
- Linear Regression
- Logistic Regression
- Support Vector Machines (SVM)
- Decision Trees
- Random Forests
- Neural networks

3.4.2 Μάθηση χωρίς επίβλεψη (Unsupervised Learning)

Στην διαδικασία μάθησης αυτού του τύπου το μοντέλο χρησιμοποιεί τα πρότυπα εισόδου $x_1, x_2, x_3, \ldots$, χωρίς όμως να διαθέτει πληροφορίες σχετικά με τους στόχους. Δηλαδή το μοντέλο προσπαθεί να μάθει από μόνο του.



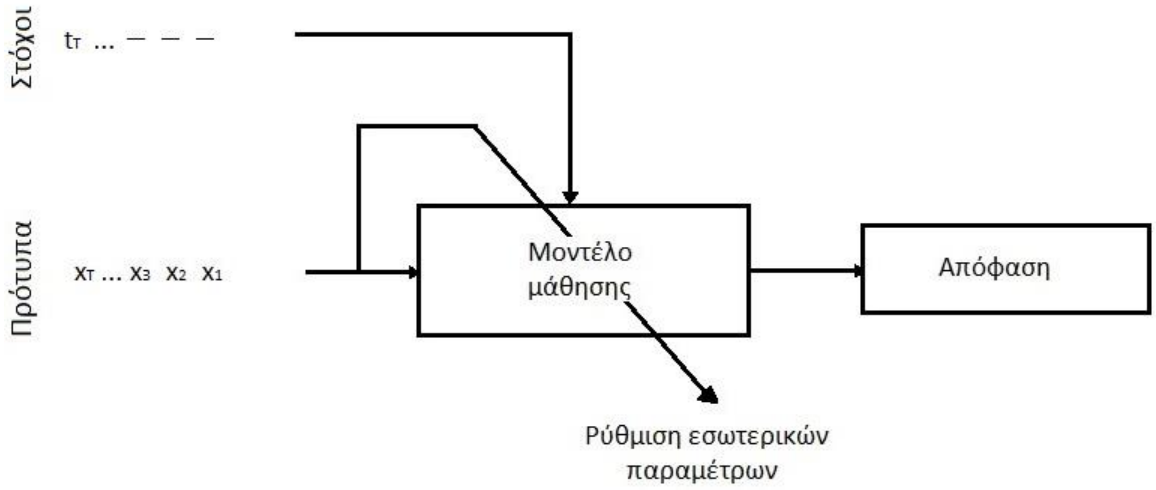
Εικόνα 3.2: Το βασικό μοντέλο της μάθησης χωρίς επίβλεψη. [4]

Μερικοί από τους πιο σημαντικούς αλγορίθμους μάθησης χωρίς επίβλεψη είναι οι εξής:

- Ομαδοποίηση (Clustering):
  - K-Means
  - BDSCAN
  - Hierachical cluster Analysis (HCA)
- Ανίχνευση ανωμαλιών και ανίχνευση καινοτομίας (Anomaly detection and novelty detection)
  - One-class Support Vector Machine
  - Isolation Forest
- Οπτικοποίηση και μείωση διαστάσεων
  - Principal Component Analysis (PCA)
  - Kernel PCA
  - Locally Linear Embedding (LLE)
  - t-Distributed Stochastic Neighbor Embedding (t-SNE)

3.4.3 Μάθηση με ενίσχυση (Reinforcement Learning)

Στην περίπτωση αυτή, το μοντέλο χρησιμοποιεί μια ακολουθία προτύπων εισόδου $x_1, x_2, x_3, ...$ και οι στόχοι είναι συνήθως τιμές ανταμοιβής ή τιμωρίας. Τις περισσότερες φορές και ιδίως όταν το σύστημα μαθαίνει να παίζει παιχνίδια, υπάρχει μία μόνο ανταμοιβή και αυτή γίνεται γνωστή στο τέλος της ακολουθίας. [4]



Εικόνα 3.3: Το βασικό μοντέλο της μάθησης με ενίσχυση. [4]

Για παράδειγμα, πολλά ρομπότ εφαρμόζουν την μάθηση με ενίσχυση για να μάθουν πώς να περπατάνε. Ένα καλό παράδειγμα αυτού του μοντέλου είναι το πρόγραμμα που ονομάζεται DeepMind AlphaGo, το οποίο, τον Μάιο του 2017 κέρδισε τον παγκόσμιο πρωταθλητή Ke Jie του παιχνιδιού Go. Το πρόγραμμα εκπαιδεύτηκε με όλες τις πιθανότητες νίκης αναλύοντας εκατομμύρια παιχνίδια και παίζοντας πάρα πολλά ενάντια τον εαυτό του. Τέλος, η διαδικασία της εκπαίδευσης ήταν απενεργοποιημένη κατά την διάρκεια των παιχνιδιών με τον πρωταθλητή, καθώς το AlpaGo, έπαιζε με τις κινήσεις που είχε εκπαιδευτεί. [5]

3.5 Overfitting και Underfitting

3.5.1 Overfitting

Η Υπεργενίκευση είναι κάτι που οι άνθρωποι κάνουν πολύ συχνά. Δυστυχώς όμως και οι μηχανές μπορούν να πέσουν στην ίδια παγίδα, εάν ο διαχειριστής του μοντέλου δεν είναι προσεκτικός. Στην μηχανική μάθηση αυτό ονομάζεται overfitting, δηλαδή το μοντέλο έχει καλή απόδοση στα δεδομένα εκπαίδευσης, αλλά δεν αποδίδει σωστά αποτελέσματα ή δεν αποδίδει καλά στα νέα δεδομένα.

Για να γίνει επίλυση της υπερμοντελοποίησης (overfitting) θα πρέπει να γίνει καλύτερος ο διαχωρισμός του συνόλου των δεδομένων σε σύνολο εκπαίδευσης (train set) και σύνολο δοκιμής (test set). Εφαρμόζοντας αυτήν την τεχνική, μπορούν να γίνουν αρχικά διάφορες δοκιμές στο σύνολο της εκπαίδευσης, ώστε να βρεθεί το πιο ταιριαστό σύνολο για αυτό το μοντέλο που χρησιμοποιείται, ώστε να υπάρχουν τα καλύτερα αποτελέσματα.

3.5.2 Underfitting

Το underfitting είναι άλλο ένα πρόβλημα που αντιμετωπίζουν τα μοντέλα, καθώς συμβαίνει όταν το μοντέλο είναι πολύ απλό για να «μάθει» την υποκείμενη δομή των δεδομένων. Το πρόβλημα του underfitting παρατηρείται όταν το μοντέλο δεν έχει καλή απόδοση στα δεδομένα εκπαίδευσης, επηρεάζοντας έτσι και την συνολική του απόδοση.

Αυτές είναι μερικές από τις κύριες λύσεις για αυτό το πρόβλημα:

- Επιλογή πιο δυνατού μοντέλου με λιγότερες παραμέτρους.
- Επιλογή καλύτερων χαρακτηριστικών στον αλγόριθμο εκμάθησης.
- Μείωση των περιορισμών στο μοντέλο (όπως είναι και οι regularization hyperparameter).

Κεφάλαιο 4: Προ επεξεργασία των Δεδομένων – Data Preprocessing

4.1 Τι είναι η προ επεξεργασία των δεδομένων

Στην σημερινή εποχή, όταν γίνεται αναφορά σε βάσεις δεδομένων, συνήθως εννοείται πως είναι μεγάλες βάσεις δεδομένων σε όγκο, οι οποίες προέρχονται από διάφορες πηγές. Αυτό έχει σαν αποτέλεσμα, τα δεδομένα τις περισσότερες φορές να μην είναι στην κατάλληλη δομή ή οι πληροφορίες που έχουν λάβει να μην είναι ορθές (δηλαδή να έχουν ακραίες τιμές – outliers), ώστε το μοντέλο που θα χρησιμοποιήσει την συγκεκριμένη βάση να μην έχει το επιθυμητό αποτέλεσμα στην ακρίβεια. Άρα, το Data Preprocessing είναι ένα σημαντικό κομμάτι τις συνολικής διαδικασίας της εξόρυξης δεδομένων, καθώς ο ρόλος του είναι να κάνει τον ποιοτικό έλεγχο των δεδομένων και έπειτα να διορθώνει με διάφορους αλγόριθμους την βάση δεδομένων όπου αυτή έχει σφάλματα.

4.2 Έλεγχος της ποιότητας των δεδομένων

Τα δεδομένα σε ένα data set μπορούν να θεωρηθούν ποιοτικά όταν ικανοποιούν τις απαιτήσεις της προβλεπόμενης χρήσης για την λειτουργία του επιλεγμένου μοντέλου. Υπάρχουν αρκετοί παράγοντες που αντιπροσωπεύουν την ποιότητα των δεδομένων, όπως είναι η ακρίβεια, η πληρότητα, η συνέπεια, η επικαιρότητα, η αξιοπιστία και η ερμηνεία.

Ατελή δεδομένα μπορεί να προκύψουν για διάφορους λόγους, όπως να μην θεωρήθηκαν σημαντικά κατά την δημιουργία της βάσης δεδομένων ή να μην καταγράφηκαν λόγω μιας δυσλειτουργίας στον εξοπλισμό και ούτω καθεξής. Για αυτόν τον λόγο ώστε να γίνει και η σωστή λειτουργία του μοντέλου μετέπειτα γίνεται το preprocessing των δεδομένων.

4.3 Οι διαδικασίες της προ επεξεργασίας των δεδομένων

Όλες οι διαδικασίας του data preprocessing είναι οι εξής: Ενσωμάτωση των δεδομένων (Data Integration 4.3.1), η Κατηγοριοποίηση των δεδομένων (Data Normalization 4.3.2), η μεταμόρφωση των Δεδομένων (Data Transformation 4.3.3), ο Καθαρισμός των Δεδομένων (Data Cleaning 4.3.4) και η μείωση των δεδομένων (Data Reduction 4.3.5).

4.3.1 Ενσωμάτωση των δεδομένων – Data Integration

Ένα σημαντικό μέρος της διαδικασίας ενσωμάτωσης είναι η δημιουργία ενός «χάρτη» δεδομένων που θα καθορίσει τον τρόπο με τον οποίο η κάθε περίπτωση θα πρέπει να διευθετηθεί σε μια κοινή δομή, έτσι ώστε να μπορεί να παρουσιαστεί ένα πραγματικό παράδειγμα στα δεδομένα το οποίο θα αντιστοιχεί πιο πολύ στον πραγματικό κόσμο.

4.3.1.1 Εύρεση περιττών χαρακτηριστικών – Finding Redundant Attributes

Ο πλεονασμός στα χαρακτηριστικά είναι ένα σοβαρό πρόβλημα που πρέπει να αποφευχθεί όσο το δυνατόν περισσότερο. Καθώς προκαλεί μεγάλη αύξηση στο μέγεθος του συνόλου των δεδομένων, άρα συνεπάγεται ότι ο χρόνος μοντελοποίησης του αλγορίθμου data mining θα αυξηθεί, όπως και να προκαλέσει overfitting στο λαμβανόμενο μοντέλο. Ένα χαρακτηριστικό μπορεί να θεωρηθεί περιττό όταν προέρχεται από ένα άλλο χαρακτηριστικό ή από ένα σύνολο αυτών. Επίσης, ασυμφωνίες στα ονόματα των χαρακτηριστικών μπορούν να προκαλέσουν πλεονασμό. Ο πλεονασμός αυτός στα attributes μπορεί να εντοπιστεί την ανάλυση της συσχέτισης των μεταβλητών (Correlation Analysis). Μέσω αυτής της ανάλυσης γίνεται μέτρηση πόσο ισχυρή είναι η επίπτωση του ενός χαρακτηριστικού στο άλλο.

Οι μέθοδοι που χρησιμοποιούνται στην συσχέτιση αυτή είναι, (i) η μέθοδος του x^2 για τα ονομαστικά δεδομένα και (ii) ο συσχετισμός απόδοσης και η συνδιακύμανση για τα αριθμητικά δεδομένα.

(i) Στα ονομαστικά δεδομένα γίνεται χρήση της μεθόδου x^2 (ονομάζεται και στατιστική μέθοδος x^2 του Pearson). Υποθέτοντας ότι, υπάρχουν δύο μεταβλητές Α και Β με τις τιμές τους αντίστοιχα $(\alpha_1, \alpha_2, \dots, \alpha_n)$ για το Α και $(b_1, b_2, \dots, b_n)$ για το Β, έτσι ώστε να γίνει η συσχέτιση των μεταβλητών. Για να γίνει αυτό όμως, δημιουργείτε ένας πίνακας με αυτές τις μεταβλητές, στον οποίο τα στοιχεία της μεταβλητής Α είναι στήλες (A_j) και τα στοιχεία της μεταβλητής Β είναι γραμμές (B_j) άρα, μπορεί εμφανιστεί ένας πιθανός συσχετισμός σε κάθε κελί του πίνακα μεταξύ των (A_j, B_j).

Η τιμή του x^2 υπολογίζεται από την σχέση: $x^2 = \sum_{i=1}^{a} \sum_{j=1}^{b} \frac{(o_{ij} - e_{ij})^2}{e_{ij}}$, (Σχέση 4,1)

Στον οποίο τύπο το o_{ij} είναι ο δείκτης συχνότητας στην σχέση (A_j, B_j) και το e_{ij} είναι ο δείκτης της αναμενόμενης συχνότητας ο οποίος δημιουργείτε από την σχέση:

$$e_{ij} = \frac{count(A = a_i) \times count(B = b_i)}{n} \quad \text{(Σχέση 4,2)}$$

Όπου το n είναι το πλήθος όλων των στοιχείων στο data set, το count(A=a_i) είναι ο συνολικός αριθμός των γραμμών του πίνακα που έχει δημιουργηθεί οι οποίοι περιέχουν την συγκεκριμένη τιμή και το count(B=b_i) είναι το σύνολο των γραμμών για το την τιμή b_i. Έπειτα, γίνεται ο υπολογισμός του πολλαπλασιασμού με βάση όλο το μέγεθος του πίνακα $(a_n \times b_n)$. Στην συνέχεια, η μέθοδος x^2 ελέγχει την ανεξαρτησία μεταξύ των δύο μεταβλητών A , B και βασίζεται με $(a-1)\times(b-1)$ βαθμούς ελευθερίας. Εάν το αποτέλεσμα από τον πίνακα είναι χαμηλότερο από τον καθορισμένο (ή η στατιστική τιμή που υπολογίζεται είναι πάνω από την απαιτούμενη στον πίνακα), μπορεί να θεωρηθεί ότι η υπόθεση αυτή απορρίπτεται και ως εκ τούτο, οι μεταβλητές A και B συσχετίζονται στατιστικά.

(ii) Ανάμεσα σε δύο αριθμητικές μεταβλητές A και B ο πιο γνωστός συντελεστής συσχέτισης είναι η συσχέτιση απόδοσης (Correlation Coefficient) ο οποίος δίνεται από την σχέση: $r_{A,B} = \frac{\sum_{i=1}^{n}(a_i-\bar{A})(b_i-\bar{B})}{n\sigma_A\sigma_B} = \frac{\sum_{i=1}^{n}(a_i b_i)-n\bar{A}\bar{B}}{n\sigma_A\sigma_B}$, (Σχέση 4,3)

Στον οποίο τύπο, το n είναι το σύνολο των περιπτώσεων, το a_i και το b_i είναι οι τιμές των μεταβλητών A και B σε όλες τις περιπτώσεις, το $\bar{A}$ και το $\bar{B}$ είναι μέση τιμή για κάθε μεταβλητή, το σ_A και το σ_B είναι η τυπική απόκλιση για το A και το B και το Σ συμβολίζει το άθροισμα του γινομένου των δύο μεταβλητών. Να σημειωθεί ότι:

$$-1 \le r_{A,B} \le +1.$$

Όταν το $r_{A,B}$ (Σχέση 4,3) είναι μεγαλύτερο του 0, τότε οι δύο μεταβλητές A και B είναι θετικά συσχετιζόμενες μεταξύ τους, δηλαδή όταν αυξηθεί στην μια μεταβλητή η τιμή θα αυξηθεί και στην άλλη. Όσο υψηλότερος είναι ο συντελεστής απόδοσης, τόσο μεγαλύτερη είναι και η συσχέτιση μεταξύ τους. Επίσης, όσο μεγαλύτερη είναι η τιμή στο $r_{A,B}$ τόσο πιο μεγαλύτερη είναι η συσχέτιση στις δυο μεταβλητές. Όταν, το $r_{A,B}$ είναι ίσο με το 0, τότε φανερώνει πως οι μεταβλητές A και B είναι ανεξάρτητες και δεν υπάρχει καμία συσχέτιση μεταξύ τους. Εάν, όμως το $r_{A,B}$ είναι μικρότερο του 0 δηλαδή αρνητικό, τότε οι μεταβλητές A και B έχουν αρνητική συσχέτιση, οπότε όταν η τιμή της μίας μεταβλητής αυξάνετε τότε η τιμή της άλλης μειώνετε. Σε έναν οπτικό έλεγχο των αποτελεσμάτων που αποκτήθηκαν τα διασκορπισμένα διαγράμματα (Scatter Plots) μπορούν να είναι αρκετά χρήσιμο για να εξετάσουν πόσο συσχετίζονται τα δεδομένα.

Ομοίως με την συσχέτιση απόδοσης μια άλλη γνωστή μέθοδος της συντελεστής συσχέτισης είναι και η συνδιακύμανση. Αυτή, είναι ένα χρήσιμο και ευρέως χρησιμοποιούμενο μέτρο στις στατιστικές προκειμένου να ελέγξει κατά πόσο οι δύο μεταβλητές A και B αλλάζουν μαζί. Έχοντας υπόψη πως οι μέσες τιμές των μεταβλητών A και B είναι οι αναμενόμενες τιμές, δηλαδή $E(A) = \bar{A}$ και $E(B) = \bar{B}$ η συνδιακύμανση μεταξύ αυτών των δυο ορίζεται ως:

$$Cov(A,B) = E\big((A-\bar{A})(B-\bar{B})\big) = \frac{\sum_{i=1}^{n}(a_i-\bar{A})(b_i-\bar{B})}{n}.$$ (Σχέση 4,4)

Σε δύο μεταβλητές που οι τιμές τους θα είναι παρόμοιες μεταξύ τους, όταν το $A > \bar{A}$ τότε πιθανώς και το $B > \bar{B}$ έτσι αποδεικνύεται πως η συνδιακύμανση είναι θετική. Από την άλλη μεριά, όταν η μια μεταβλητή τείνει να είναι μεγαλύτερη από την αναμενόμενη τιμή της και η άλλη μεταβλητή είναι χαμηλότερη από την αναμενόμενη τιμή, τότε η συνδιακύμανση είναι αρνητική.

4.3.2 Κανονικοποίηση των δεδομένων – Data Normalization

Σε αυτήν την κατηγορία γίνεται αναφορά στους μετασχηματισμούς που μετατρέπουν την κατανομή των αρχικών τιμών στις μεταβλητές σε ένα νέο σύνολο τιμών με τις επιθυμητές ιδιότητες που ορίζει ο διαχειριστής του μοντέλου.

4.3.2.1 Κανονικοποίηση μικρότερου – μεγαλύτερου / Min – Max Normalization

Το min – max Normalization στοχεύει στην κλιμάκωση όλων των αριθμητικών τιμών ν μιας αριθμητικής μεταβλητής Α σε ένα καθορισμένο εύρος το οποίο είναι: $[new - min_A, new - max_A]$. Έτσι, θα ληφθεί μια μετασχηματισμένη τιμή από χρησιμοποιώντας τον ακόλουθο τύπο:

$$v' = \frac{v - min_A}{max_A - min_A}(new - max_A - new - min_A) + new - min_A, \qquad \text{(Σχέση 4,5)}$$

Όπου το max_A και το min_A είναι οι τιμές από το μεγαλύτερο και το μικρότερο αντίστοιχα.

Αυτός ο τύπος κανονικοποίησης είναι πολύ συνηθισμένος σε data sets που προετοιμάζονται σε χρήση για μεθόδους μάθησης που βασίζονται σε αποστάσεις. Χρησιμοποιώντας μια κανονικοποίηση για την αναπροσαρμογή όλων των δεδομένων της βάσης στο ίδιο εύρος τιμών, θα αποφευχθούν οι μεταβλητές στον υπολογισμό της «απόστασης», οι οποίες έχουν μεγάλη διαφορά $max_A - min_A$ έναντι άλλων που δεν έχουν. Αυτή η κανονικοποίηση είναι επίσης γνωστή για την επιτάχυνση της μαθησιακής διαδικασίας στο ANN (Artificial Neural Network), βοηθώντας με αυτόν τον τρόπο τα βάρη να συγκλίνουν γρηγορότερα.

4.3.2.2 Κανονικοποίηση Z-Score – Z-Score Normalization

Σε ορισμένες περιπτώσεις το min-max Normalization δεν είναι χρήσιμο ή δεν μπορεί να εφαρμοστεί. Αυτό συμβαίνει όταν, οι ελάχιστες ή οι μέγιστες τιμές της μεταβλητής δεν είναι γνωστές ή ακόμα και όταν είναι διαθέσιμες αυτές οι τιμές η παρουσία ακραίων και μεγάλων τιμών σε διαφορές μπορεί να προκαλέσει το min-max Normalization σε ανακρίβειες στα τελικά αποτελέσματα.

Εάν το $\bar{A}$ είναι η μέση τιμή της μεταβλητής Α και το σ_A είναι ο συντελεστής απόκλισης τότε η αρχική τιμή του v του Α κανονικοποιείται με τον τύπο:

$$v' = \frac{v-\bar{A}}{\sigma_A}. \qquad \text{(Σχέση 4,6)}$$

Με την υλοποίηση του μετασχηματισμού παρουσιάζει την μέση τιμή ίση με το 0 και την τυπική απόκλισης ίσο με το 1. Εάν η μέση τιμή και η τυπική απόκλιση δεν είναι διαθέσιμα τότε μπορούν να βρεθούν από τους ακόλουθους τύπους:

$$\bar{A} = \frac{1}{n}\sum_{i=1}^{n} v_i \qquad \text{(Σχέση 4.7)}$$

Και η τυπική απόκλιση σε συνδυασμό με την (Σχέση 4.7)

$$\sigma_A = +\sqrt{\frac{1}{n}\sum_{i=1}^{n}(v_i - \bar{A})^2} \qquad \text{(Σχέση 4.8)}$$

Μια άλλη παραλλαγή του Z-score Normalization χρησιμοποιεί τη μέση απόλυτη απόκλιση s_σ της μεταβλητής Α αντί για την τυπική απόκλιση. Ο τύπος της διαμορφώνεται ως:

$$s_A = \frac{1}{n}\sum_{i=1}^{n}|v_i - \bar{A}| \qquad \text{(Σχέση 4.9)}$$

Με αποτέλεσμα το Z-Score Normalization γίνεται:

$$v' = \frac{v-\bar{A}}{s_A} \qquad \text{(Σχέση 4.10)}$$

Το πλεονέκτημα στον υπολογισμό με τη μέση απόλυτη απόκλιση s_A έναντι της τυπικής απόκλισης είναι ισχυρότερη στις ακραίες τιμές, καθώς οι αποκλίσεις από τον μέσο όρο υπολογίζονται με απόλυτο $|v_i - \bar{A}|$ και όχι υψωμένο στο τετράγωνο.

4.3.3 Μετασχηματισμός των Δεδομένων – Data Transformation

Στην προηγούμενη υποενότητα (4.3.2) έγινε αναφορά σε μερικές βασικές τεχνικές μετασχηματισμού έτσι ώστε, να γίνει προσαρμογή του εύρους των τιμών από τις μεταβλητές στις ανάγκες του αλγορίθμου της εξόρυξης δεδομένων. Ο στόχος αυτής της υποενότητας είναι να γίνει ανάλυση ορισμένων διαδικασιών μεταμόρφωσης των μεταβλητών. Η μεταμόρφωση ή αλλιώς μετασχηματισμός των δεδομένων συνδυάζει συνήθως τα αρχικά ακατέργαστα δεδομένα χρησιμοποιώντας διάφορους μαθηματικούς τύπους.

4.3.3.1 Γραμμικοί Μετασχηματισμοί – Linear Transformation

Όταν η κανονικοποίηση δεν είναι αρκετή για την προσαρμογή των δεδομένων ώστε να βελτιώσει το επιλεγμένο μοντέλο, τότε η συγκέντρωση αυτή των πληροφοριών μέσα στις μεταβλητές μπορεί να είναι αρκετά επωφελής για τους γραμμικούς μετασχηματισμούς. Το Linear Transformation βασίζεται σε απλούς αλγεβρικούς μετασχηματισμούς όπως το άθροισμα, ο μέσος όρος και ούτω καθεξής. Αν $A = A_1, A_2 \dots, A_n$ είναι ένα σύνολο μεταβλητών και $B = B_1, B_2 \dots, B_n$ είναι το υποσύνολο της μεταβλητής Α. Τότε δημιουργείτε η ακόλουθη έκφραση:

$$Z = r_1 B_1 + r_2 B_2 + \cdots + r_n B_n \qquad \text{(Σχέση 4.11)}$$

Έτσι, κατασκευάζετε μια συμπληρωματική μεταβλητή Z λαμβάνοντας έναν γραμμικό συνδυασμό τιμών από το Β.

4.3.3.2 Πολυωνυμική Προσέγγιση Μετασχηματισμού – Polynomial Approximations of Transformations

Όταν δεν υπάρχει η δυνατότητα εξαγωγής «γνώσης» από την βάση δεδομένων, τότε ένας μετασχηματισμός f μπορεί να προσεγγιστεί μέσω ενός πολυωνυμικού μετασχηματισμού χρησιμοποιώντας μια «ωμή» αναζήτηση με έναν βαθμό την φορά.

Σε ένα σύνολο μεταβλητών $X_1 X_2, \dots, X_n$ πρέπει να γίνει ο υπολογισμός μιας παράγωγης μεταβλητής Υ από τις ήδη υπάρχων μεταβλητές. Οπότε, ορίζεται η μεταβλητή Υ ως συνάρτηση:

$$Y = f(X_1, X_2, \dots, X_n) \qquad \text{(Σχέση 4.12)}$$

Στο οποίο το f μπορεί να είναι ένα οποιοδήποτε είδος συνάρτησης. Κάθε γραμμή $X_i = (X_1, X_2, \dots, X_n)$ μπορεί να θεωρηθεί ως ένα σημείο στον ευκλείδειο χώρο. Χρησιμοποιώντας την προσέγγιση Weistrass, υπάρχει μια πολυωνυμική συνάρτηση f η οποία μπορεί να δώσει την τιμή του Y_i για κάθε τιμή του X_i.

Υπάρχουν πολλά πολυώνυμα τα οποία επαληθεύουν το $Y = f(X)$ όπως είναι στην [Σχέση 4.12], είναι διαφορετικά στην έκφραση τους όμως με την ίδια έξοδο για το σημείο που δίνεται στο σύνολο αυτών των δεδομένων.

4.3.3.3 Μετασχηματισμοί Κατάταξης – Rank Transformation

Μια αλλαγή στην κατανομή των μεταβλητών μπορεί να οδηγήσει σε αλλαγή της απόδοσης του μοντέλου, καθώς ενδέχεται να αποκαλυφθούν σχέσεις που είχαν επισκιαστεί από προηγούμενες κατανομές. Ο απλούστερος μετασχηματισμός για να επιτευχθεί αυτό σε

αριθμητικές μεταβλητές είναι η αντικατάσταση της τιμής των μεταβλητών με την κατάταξη τους. Η μεταβλητή που χρειάζεται τον μετασχηματισμό θα μετατραπεί σε μια μεταβλητή που περιέχει ακέραιες τιμές οι οποίες κυμαίνονται από 1 έως n, όπου n είναι το σύνολο των αριθμών σε αυτό το data set.

Στην συνέχεια γίνεται η μετατροπή των κατατάξεων σε κανονικές τιμές, που αντιπροσωπεύουν τις πιθανότητες τους στην κανονική κατανομή διαδίδοντας τις τιμές αυτές στην καμπύλη Gaussian χρησιμοποιώντας τον εξής μετασχηματισμό:

$$y = \Phi^{-1}\left(\frac{r_i - \frac{3}{8}}{n + \frac{1}{4}}\right) \qquad \text{(Σχέση 4.13)}$$

Όπου το r_i είναι η κατάταξη της παρατήρησης i και Φ η αθροιστική κανονική συνάρτηση.

Αυτός ο μετασχηματισμός είναι χρήσιμος για την απόκτηση μιας νέας μεταβλητής που είναι πολύ πιθανό να συμπεριφέρεται σαν μια κανονικά κατανεμημένη μεταβλητή. Ωστόσο αυτός ο μετασχηματισμός δεν μπορεί να εφαρμοστεί χωριστά στο training και test set. Επομένως, αυτός ο μετασχηματισμός συνιστάται μόνο όταν τα δεδομένα δοκιμών και εκπαίδευσης είναι τα ίδια.

4.3.4 Καθαρισμός των Δεδομένων – Data cleaning

Καθώς οι βάσεις δεδομένων τείνουν να περιέχουν ελλιπή δεδομένα που ενδέχεται να περιέχουν θόρυβο και να μην είναι συνεπή, εκεί χρησιμοποιείται ο καθαρισμός των δεδομένων (Data cleaning). Ο στόχος είναι να συμπληρωθούν οι τιμές που λείπουν, να εξομαλυνθεί ο θόρυβος ο οποίος αλλοιώνει τα δεδομένα και να διορθωθούν οι ασυνέπειες στα δεδομένα αυτά.

4.3.4.1 Μέθοδοι επίλυσης των τιμών που λείπουν (missing values) σε μια βάση δεδομένων

- **<u>Αγνόηση της γραμμής:</u>** Αυτή η μέθοδος είναι αποτελεσματική μόνο όταν η γραμμή περιέχει πολλές μεταβλητές με τιμές που λείπουν. Επίσης, είναι ιδιαίτερα μη αποτελεσματική μέθοδος όταν η κάθε μεταβλητή έχει διαφορετικό σύνολο από τις τιμές που τις λείπουν. Αγνοώντας μια σειρά σε μια μεταβλητή θα πρέπει να αφαιρεθεί και από τις υπόλοιπες μεταβλητές, με αποτέλεσμα να μην μπορούν να χρησιμοποιηθούν κάποια δεδομένα που θα μπορούσαν να ήταν χρήσιμα.
- **<u>Συμπλήρωση των κενών τιμών χειροκίνητα:</u>** Αυτή η προσέγγιση είναι χρονοβόρα και γίνεται και ανέφικτη, όταν υπάρχει μεγάλος όγκος δεδομένων και μεγάλος όγκος τιμών που λείπουν.
- **<u>Η χρήση μιας γενικής σταθεράς για την κάλυψη των κενών τιμών:</u>** Σε αυτήν την μέθοδο γίνεται αντικατάσταση όλων των τιμών που λείπουν με μια γενική

σταθερά. Από την στιγμή που όλα τα στοιχεία αυτά αντικατασταθούν με αυτήν την σταθερά, τότε το πρόγραμμα εξόρυξης μπορεί λανθασμένα να θεωρήσει πως αυτά τα δεδομένα είναι ιδιαίτερα και να μην έχει το επιθυμητό αποτέλεσμα. Ως εκ τούτου, αν και η μέθοδος αυτή είναι απλή, δεν είναι αξιόπιστη.

- **Χρήση της μέσης ή μεσαίας τιμής (mean of median) για την συμπλήρωση των τιμών που λείπουν:** Αυτή η μέθοδος βρίσκει την μέση τιμή μέσα από την βάση και συμπληρώνει όλα τα κενά με αυτήν, όταν πρόκειται για αριθμητικές τιμές και την μεσαία τιμή σε όλες τις άλλες περιπτώσεις.
- **Χρήση της μέσης ή μεσαίας τιμής για όλα τα δείγματα που ανήκουν στην ίδια κατηγορία με την επιλεγμένη σειρά:** Εάν, πρέπει να γίνει μια κατηγοριοποίηση στην βάση δεδομένων σύμφωνα με ένα επιλεγμένο κριτήριο από τον δημιουργό του μοντέλου, τότε γίνεται αντικατάσταση στα κενά της ίδιας κατηγορίας με την μέση τιμή της σειράς που είναι επιλεγμένη. Εάν η κατανομή των δεδομένων για την συγκεκριμένη κατηγορία είναι λοξή, τότε η μεσαία τιμή είναι καλύτερη επιλογή.
- **Χρήση της πιο πιθανής τιμής για τα missing values:** Αυτήν την μέθοδο την χρησιμοποιεί ένας αλγόριθμος πρόβλεψης (όπως το Decision tree), η οποία μέθοδος θα δείχνει αυτόματα την καλύτερη τιμή για τα κενά με βάση τα χαρακτηριστικά της βάσης.

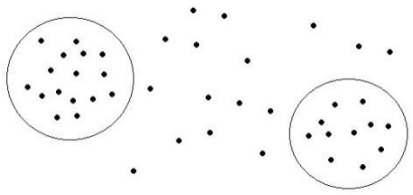
4.3.4.2 Μέθοδοι μείωσης του θορύβου στα δεδομένα

Πέρα από τα missing values σε μια βάση δεδομένων, μπορεί να υπάρξουν και δεδομένα με τον λεγόμενο ***θόρυβο*** (***Noise Data***). Ο θόρυβος στα δεδομένα είναι κάποιο τυχαίο σφάλμα ή μια διακύμανση σε μια μετρούμενη μεταβλητή.

Οι ακόλουθες τεχνικές χρησιμοποιούνται για την εξομάλυνση των δεδομένα από την μείωση του θορύβου:

- **Binning - Δεδομένα σε κιβώτια:** Οι μέθοδοι Binning εξομαλύνουν μια ταξινομημένη τιμή από ένα data set με βάση τις «γειτονικές» της τιμές, δηλαδή τις τιμές γύρω από την επιλεγμένη. Έπειτα, καθώς οι μέθοδοι Binning συμβουλεύονται τις «γειτονικές» τιμές, πρέπει να εκτελέσουν μια τοπική εξομάλυνση. Στην συνέχεια, όλες οι ταξινομημένες τιμές κατανέμονται σε έναν αριθμό από «κιβώτια» ή «καλάθια» (buckets or bins). Σε αυτές τις εξομαλύνσεις οι τιμές για την καλύτερη διόρθωση τους, μπορούν να αντικατασταθούν με την μέση ή την μεσαία τιμή (4.3.4.1). Όταν γίνεται εξομάλυνση από τα όρια των κιβωτίων, οι ελάχιστες και οι μέγιστες τιμές στα δεδομένα μέσα στο κιβώτιο θεωρούνται ως όρια του κιβωτίου. Έπειτα, κάθε τιμή του κιβωτίου αντικαθίσταται από την πλησιέστερη οριακή τιμή. Τέλος, όσο μεγαλύτερο είναι το πλάτος στα όρια, τόσο καλύτερο είναι το αποτέλεσμα στην εξομάλυνση.

- **<u>Regression – Παλινδρόμηση:</u>** Η εξομάλυνση των δεδομένων μπορεί να γίνει και με παλινδρόμηση, η οποία είναι μια τεχνική που προσαρμόζει τις τιμές των δεδομένων σε μια συνάρτηση. Η γραμμική παλινδρόμηση (Linear Regression) χρησιμοποιείται για την εύρεση της πιο συμβατής γραμμής ανάμεσα σε δυο μεταβλητές, με αποτέλεσμα να μπορεί να χρησιμοποιηθεί η μια μεταβλητή για να κάνει πρόβλεψη στην άλλη. Η πολλαπλή γραμμική παλινδρόμηση (Multiple Linear Regression) είναι μια επέκταση του Linear Regression, καθώς εμπλέκονται παραπάνω από δυο μεταβλητές, με αποτέλεσμα τα δεδομένα να προσαρμόζονται σε μια πολυδιάστατη επιφάνεια.
- **<u>Outlier Analysis - Ανάλυση Ακραίων τιμών:</u>** Η ανίχνευση των ακραίων τιμών γίνεται με ομαδοποίηση (Clustering), δηλαδή όλες οι τιμές που είναι παρόμοιες οργανώνονται σε ομάδες. Έτσι, οι τιμές που βρίσκονται εκτός του συνόλου των ομάδων θεωρούνται ακραίες τιμές (Εικόνα 4.1).



Εικόνα 4.1: Η ομαδοποίηση των τιμών (όλες οι ακραίες τιμές βρίσκονται εκτός κύκλου). [2]

4.3.5 Μείωση των δεδομένων – Data Reduction

Οι τεχνικές μείωσης των δεδομένων (Data Reduction) εφαρμόζονται όταν χρειάζεται να ληφθεί μια μειωμένη αναπαράσταση του συνόλου των δεδομένων που αρκετά μικρότερη σε όγκο συγκριτικά με την αρχική, όμως διατηρεί στενά την ακεραιότητα των αρχικών δεδομένων. Δηλαδή, η εξόρυξη των δεδομένων στο νέο μειωμένο data set μπορεί να είναι πιο αποτελεσματική καθώς παράγει τα σχεδόν ίδια αναλυτικά αποτελέσματα. Ακολουθεί μια ανάλυση μερικών από αυτές τις μεθόδους:

4.3.5.1 Ανάλυση Βασικών Στοιχείων - Principal Components Analysis (PCA)

Σε μια υποθετική βάση στην οποία τα προς μείωση δεδομένα αποτελούνται από γραμμές ή από διανύσματα δεδομένων (Data vectors), οι οποίες εμπεριέχουν n κατηγορίες ή μεταβλητές. Η ανάλυση των βασικών στοιχείων (PCA, ονομάζεται επίσης και μέθοδος Karhunen – Loeve ή K-L) αναζητά ορθογώνια διανύσματα k-n όπου μπορούν να χρησιμοποιηθούν καλύτερα για την αναπαράσταση των δεδομένων (όπου $k \leq n$). Αυτό έχει ώς αποτέλεσμα τα αρχικά δεδομένα να προβάλλονται σε πολύ μικρότερο χώρο, ώστε να γίνει εφικτή η μείωση των διαστάσεων. Άρα, το PCA δημιουργεί ένα «εναλλακτικό»

μικρότερο σύνολο μεταβλητών, στο οποίο προβάλλεται το αρχικό data set. Τέλος, το PCA μπορεί να εφαρμοστεί σε ταξινομημένες και μη ταξινομημένες μεταβλητές με αραιά και λανθασμένα δεδομένα. Όπως επίσης, μπορεί να εφαρμοστεί και σε πολυδιάστατα δεδομένα (multidimensional data) τα οποία μπορούν να αντιμετωπιστούν μειώνοντας το πρόβλημα σε δυο διαστάσεις.

4.3.5.2 Επιλογή Υποομάδας Κατηγοριών – Attribute Subset Selection

Η επιλογή υποσυνόλου κατηγοριών μειώνει το αρχικό μέγεθος του συνόλου των δεδομένων αφαιρώντας άσχετες ή περιττές κατηγορίες. Ο στόχος αυτής της μεθόδου είναι να βρει το ελάχιστο σύνολο από τις κατηγορίες, έτσι ώστε το πιθανό αποτέλεσμα κατανομής των κατηγοριών των δεδομένων να είναι όσο το δυνατόν πιο κοντά στην αρχική κατανομή η οποία εμπεριέχει όλες τις κατηγορίες. Οπότε η εξόρυξη δεδομένων σε ένα μειωμένο σύνολο από κατηγορίες αποκτά ένα επιπλέον πλεονέκτημα, το οποίο είναι πως όταν γίνεται η μείωση στον αριθμό των κατηγοριών που εμφανίζονται στα αποτελέσματα, τότε γίνεται πιο κατανοητή η νέα βάση που δημιουργείτε.

4.3.5.3 Ομαδοποίηση – Clustering

Οι τεχνικές ομαδοποίηση (Clustering techniques) θεωρούν τις γραμμές των δεδομένων σαν αντικείμενα. Στην συνέχεια, γίνεται ο διαχωρισμός των αντικειμένων σε ομάδες ή συστάδες (groups or clusters), με αυτόν τον τρόπο τα αντικείμενα μέσα στη συστάδα που θα δημιουργηθεί να είναι «παρόμοια» μεταξύ τους και «ανόμοια» με τα αντικείμενα των άλλων συστάδων. Η ομοιότητα ορίζεται στις περισσότερες φορές ως προς το πόσο κοντά είναι σε «απόσταση» τιμών τα αντικείμενα μεταξύ τους, με βάση μια προκαθορισμένη συνάρτηση απόστασης. Ενώ η ποιότητα μιας συστάδας μπορεί να αντιπροσωπεύεται από τη διάμετρο της, δηλαδή τη μέγιστη απόσταση μεταξύ οποιωνδήποτε δυο αντικειμένων σε αυτήν. Η απόσταση Centroid είναι ένα εναλλακτικό μέτρο για την ποιότητα της συστάδας, καθώς ορίζεται από τη μέση απόσταση κάθε αντικειμένου της συστάδας από το κεντροειδές της (το κεντροειδές είναι το μέσο αντικείμενο στη συστάδα).

Στο data Reduction η μέθοδος του clustering χρησιμοποιείται για την αντικατάσταση των πραγματικών δεδομένων με δεδομένα μέσα σε ομάδες. Η αποτελεσματικότητα αυτής της τεχνικής εξαρτάται από τη «φύση» των δεδομένων, δηλαδή είναι πιο αποτελεσματική για δεδομένα που μπορούν να οργανωθούν σε ομάδες, παρά για δεδομένα που είναι με missing values ή θόρυβο.

Κεφάλαιο 5: Μοντελοποίηση – Modeling

5.1 Γραμμική παλινδρόμηση – Linear Regression

Η αρχή λειτουργίας ενός γραμμικού μοντέλου υλοποιείται από μια γραμμική συνάρτηση. Γενικότερα, ένα γραμμικό μοντέλο κάνει μια πρόβλεψη υπολογίζοντας το άθροισμα των βαρών από τις μεταβλητές στην είσοδο του με συνδυασμό με μια σταθερά που ονομάζεται **όρος μεροληψίας** (bias term ή intercept term) και ανήκει στην κατηγορία του θορύβου. Η σχέση για αυτό το μοντέλο πρόβλεψης είναι η εξής:

$$\hat{y} = A_0 + A_1 x_1 + A_2 x_2 + \cdots + A_n x_n \qquad \text{(Σχέση 5.1)}$$

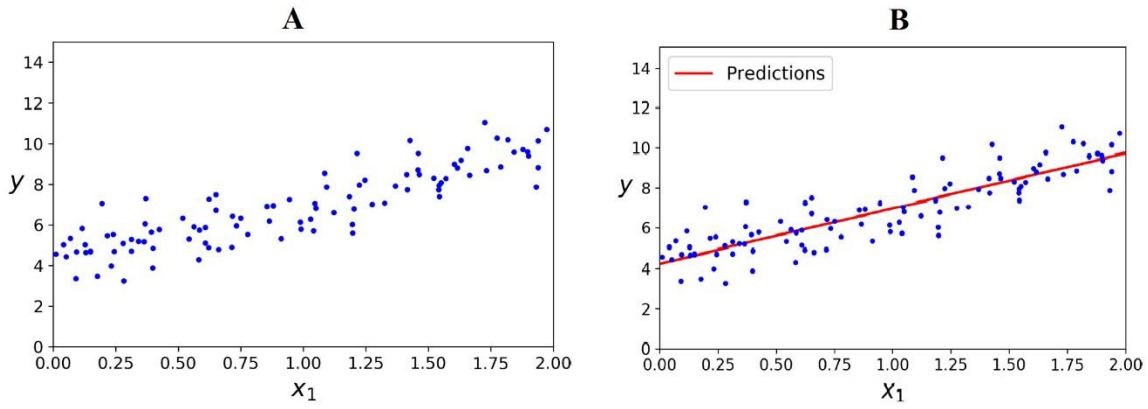
Στην οποία: το $\hat{y}$ είναι η τιμή της πρόβλεψης, το A_0 είναι ο όρος μεροληψίας, το x_n είναι η φυσική μεταβλητή και το A_n είναι φυσική μεταβλητή για το βάρος. Η (Σχέση 5.1) μπορεί να γίνει και πιο στοχευμένη χρησιμοποιώντας την σε μια διανυσματική μορφή: $\hat{y} = A * \chi$ (Σχέση 5.2), όπου το A είναι το διάνυσμα για τις σταθερές των μεταβλητών του βάρους και του όρου μεροληψίας A_0 και το x αντιπροσωπεύει το διάνυσμα των μεταβλητών.

Στην μηχανική μάθηση τα διανύσματα αυτά συχνά αντιπροσωπεύονται και ως διανύσματα σε πίνακες, δηλαδή είναι σε μια μορφή πινάκων δυο διαστάσεων με μια στήλη. Με αυτήν την λογική, εάν το A και το x είναι διανύσματα στήλης, τότε δημιουργείται η νέα σχέση για την πρόβλεψη $\hat{y} = A^T * x$ (Σχέση 5.3), στην οποία το A^T αντιπροσωπεύει τον ανάστροφο πίνακα του A. Φυσικά είναι η ίδια τιμή πρόβλεψης με την εξαίρεση ότι τώρα εκπροσωπείται ως μονοκύτταρος πίνακας (δηλαδή σε μια μορφή ενός πίνακα με ένα μόνο κελί).

Έπειτα πρέπει να γίνει η εκπαίδευση τους μοντέλου. Για να γίνει η εκπαίδευση του, θα πρέπει να καθοριστούν οι παράμετροι (μεταβλητές), έτσι ώστε το μοντέλο να ταιριάζει καλύτερα στο training set. Για τον σκοπό αυτόν χρειάζεται ένα μέτρο για το πόσο «καλά» ή «κακά» το μοντέλο ταιριάζει στα δεδομένα εκπαίδευσης. Το πιο κοινό μέτρο απόδοσης ενός μοντέλου παλινδρόμησης είναι η εύρεση της Ρίζας του μέσου τετραγωνικού σφάλματος (**RMSE** – Rout Mean Square Error). Επομένως, για την εκπαίδευση του μοντέλου γραμμικής παλινδρόμησης, πρέπει να βρεθεί η τιμή του A που ελαχιστοποιεί το RMSE. Για να βρεθεί η τιμή A που ελαχιστοποιεί τη συνάρτηση αυτή, υπάρχει η ακόλουθη μαθηματική εξίσωση (Σχέση 5.4) η οποία ονομάζεται Κανονική Εξίσωση (Normal Equation) και δίνει το αποτέλεσμα άμεσα:

$$\hat{A} = x^T * y \qquad \text{(Σχέση 5.4)}$$

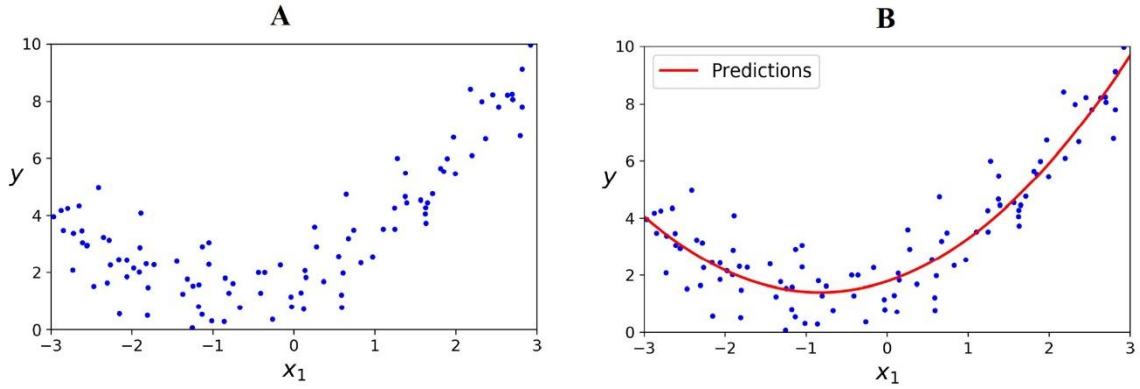
Αφού γίνει η εκπαίδευση του μοντέλου γραμμικής παλινδρόμησης (χρησιμοποιώντας την κανονική εξίσωση ή οποιονδήποτε άλλο αλγόριθμο) οι προβλέψεις αυτού του απλού μοντέλου γίνονται πολύ γρήγορες. Καθώς, ακόμα και να προστεθούν νέες παράμετροι ο χρόνος εκτέλεσης τους θα είναι απλώς προσθετικός.



Εικόνα 5.1: (A) Γραμμικό σύνολο δεδομένων με τυχαία δημιουργία, (B) Πρόβλεψη της γραμμικής παλινδρόμησης. [5]

5.1.1 Πολυωνυμική παλινδρόμηση – Polynomial Regression

Στην περίπτωση όμως που τα δεδομένα είναι πιο περίπλοκα από μια ευθεία γραμμή την λύση την δίνει η τεχνική που ονομάζεται Πολυωνυμική Παλινδρόμηση (Polynomial Regression). Αυτή η τεχνική χρησιμοποιεί ένα γραμμικό μοντέλο για την προσαρμογή μη γραμμικών δεδομένων. Αυτό επιτυγχάνετε προσθέτοντας στην [Σχέση 5.3] νέες μεταβλητές οι οποίες θα είναι δυνάμεις των ήδη υπάρχων μεταβλητών στο μοντέλο.



Εικόνα 5.2: (A) Μη γραμμικά και με θόρυβο δεδομένα (B) Πρόβλεψη του μοντέλου της πολυωνυμικής παλινδρόμησης. [5]

Επίσης, όταν υπάρχουν πολλές μεταβλητές στα δεδομένα το μοντέλο πολυωνυμικής παλινδρόμησης είναι ικανό να εντοπίσει σχέσεις μεταξύ τους (κάτι το οποία δεν είναι εφικτό να γίνει σε απλό μοντέλο Linear Regression). Αυτό γίνεται γιατί, το μοντέλο Polynomial Regression προσθέτει επίσης όλους τους συνδυασμούς των μεταβλητών μέχρι τον δεδομένο βαθμό του πολυωνύμου που εξετάζεται.

5.2 Λογιστική Παλινδρόμηση – Logistic Regression

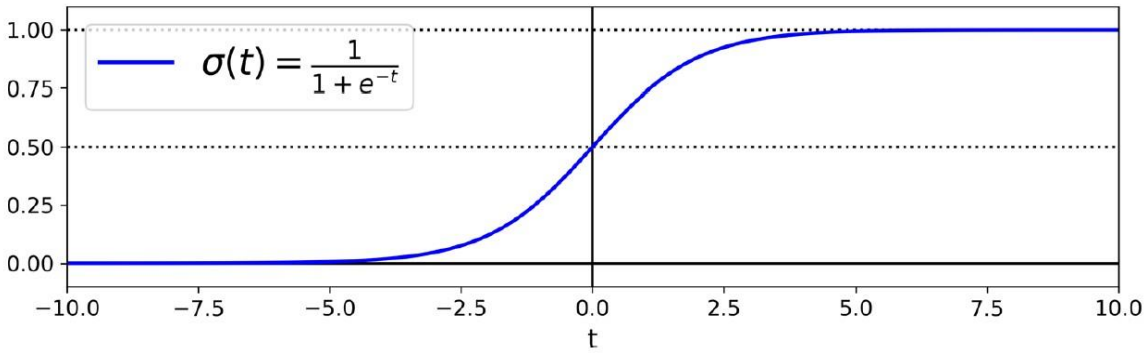
Ορισμένοι αλγόριθμοι παλινδρόμησης χρησιμεύουν περισσότερος για λύσεις προβλημάτων classification (κατηγοριοποίησης). Η Λογιστική Παλινδρόμηση (Logistic Regression ή Logit Regression) χρησιμοποιείται συνήθως για να γίνει η εκτίμηση της πιθανότητας πως η καινούργια εγγραφή η οποία θα καταχωρηθεί, θα ανήκει στην συγκεκριμένη κατηγορία που γίνεται ο έλεγχος (για παράδειγμα η πιθανότητα ενός email να είναι ανεπιθύμητο). Εάν η πιθανότητα που κάνει πρόβλεψη το μοντέλο για την συγκεκριμένη εγγραφή είναι μεγαλύτερη από το 50%, τότε το μοντέλο προβλέπει πως η εγγραφή ανήκει σε αυτή την κατηγορία (δηλαδή στα ανεπιθύμητα). Οπότε, αυτό το μοντέλο χωρίζεται σε δυο πιθανά αποτελέσματα, την **θετική κλάση** (positive class) η οποία έχει ως ένδειξη το δυαδικό «1» και την **αρνητική κλάση** (negative class) η οποία έχει ως ένδειξη το δυαδικό «0». Έτσι, καθιστάτε ως ένα μοντέλο δυαδικής κατηγοριοποίησης.

Η λειτουργία ενός μοντέλου Λογιστικής Παλινδρόμησης είναι περίπου η ίδια με αυτήν της γραμμικής παλινδρόμησης. Η μόνη διαφορά τους είναι πως το μοντέλο Logistic Regression υπολογίζει ένα σταθμισμένο άθροισμα των μεταβλητών εισαγωγής (μαζί και το bias term), όμως αντί να εξάγει το αποτέλεσμα απευθείας όπως το linear regression, δίνει στην έξοδο την λογαριθμική μορφή αυτού του αποτελέσματος. Η εκτιμώμενη πιθανότητα του μοντέλου Λογιστικής Παλινδρόμησης δίνεται από την ακόλουθη διανυσματική μορφή:

$$\hat{p} = \sigma(x^T * A) \qquad \text{(Σχέση 5.5)}$$

Στην οποία το σ είναι η σιγμοειδής συνάρτηση η οποία εξάγει έναν αριθμό μεταξύ του 0 και του 1. Η συνάρτηση αυτή ορίζεται ως:

$$\sigma(t) = \frac{1}{1+exp(-t)} \qquad \text{(Σχέση 5.6)}$$



Εικόνα 5.3: Λογιστική συνάρτηση του σ. [5]

Έτσι, γίνεται η παρατήρηση από την Εικόνα 5.3 ότι το σ(t) είναι μικρότερο από 0.5 όταν το t είναι μικρότερο από 0 και αντίστοιχα το σ(t) είναι μεγαλύτερο ή ίσο από 0.5 όταν το t είναι μεγαλύτερο ή ίσο με το 0.

Μόλις το μοντέλο της Λογιστικής Παλινδρόμησης βγάλε την εκτιμώμενη πιθανότητα από το $\hat{p}$ [Σχέση 5.5] για το που ανήκει το x τότε μπορεί να γίνει τη πρόβλεψη $\hat{y}$ αρκετά εύκολη. Η ακόλουθη εξίσωση αντιπροσωπεύει την πρόβλεψη του μοντέλου:

$$\hat{y} = \begin{cases} 0 \; \varepsilon\acute{\alpha}\nu \; \hat{p} < 0.5 \\ 1 \; \varepsilon\acute{\alpha}\nu \; \hat{p} \geq 0.5 \end{cases} \qquad \text{(Σχέση 5.7)}$$

Στο οποίο μοντέλο πρόβλεψης είναι 1 εάν το $x^T * A$ είναι θετικό και 0 εάν είναι αρνητικό.

5.2.1 Η εκπαίδευση του μοντέλου Logistic Regression

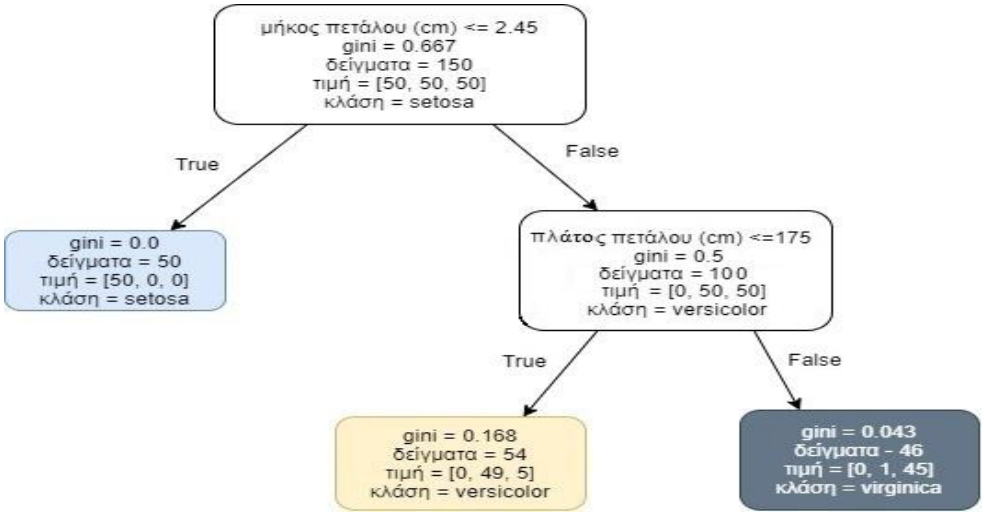
Ο στόχος της εκπαίδευσης είναι να γίνει ο κατάλληλος ορισμός της διανυσματικής μεταβλητής Α έτσι ώστε το μοντέλο να μπορεί να κάνει σωστή εκτίμηση. Δηλαδή, όταν υπάρχουν υψηλές πιθανότητες να είναι η κλάση θετική (y=1) και όταν υπάρχουν χαμηλές πιθανότητες να είναι η κλάση αρνητική (y=0). Αυτό μπορεί να διατυπωθεί με την συνάρτηση κόστους (Σχέση 5.8).

$$c(\theta) = \begin{cases} -\log(\hat{p}) \; \varepsilon\acute{\alpha}\nu \; y = 1 \\ -log(1 - \hat{p}) \; \varepsilon\acute{\alpha}\nu \; y = 0 \end{cases} \qquad \text{(Σχέση 5.8)}$$

Η συγκεκριμένη συνάρτηση κόστους αποκτάει περισσότερη σημασία ύπαρξης για τους εξής δυο λόγους. Καθώς το $-\log(t)$ μεγαλώνει όταν το t πλησιάζει το 0, οπότε το c(θ) θα είναι μεγάλο αν το μοντέλο εκτιμήσει μια πιθανότητα κοντά στο 0 για μια θετική κλάση και επίσης θα είναι πολύ μεγάλο το c(θ) αν το μοντέλο εκτιμήσει μια πιθανότητα κοντά στο 1 για μια αρνητική κλάση. Από την άλλη μεριά, το $-\log(t)$ είναι κοντά στο 0 όταν το t είναι κοντά στο 1, οπότε το c(θ) θα είναι κοντά στο 0 εάν το μοντέλο εκτιμήσει μια πιθανότητα κοντά στο 0 για μια αρνητική παρουσία ή κοντά στο 1 για μια θετική παρουσία.

5.3 Δέντρα Απόφασης – Decision Trees

Ο αλγόριθμος των δέντρων απόφασης είναι ένας ευρέως χρησιμοποιούμενος αλγόριθμος συνήθως για λύσεις προβλημάτων κατηγοριοποίησης (classification) και παλινδρόμησης (regression), ο οποίος χρησιμοποιεί τιμές μεταβλητών για να χωρέσει τον χώρο αποφάσεων σε μικρότερους υποχώρους με έναν επαναληπτικό τρόπο. Ένα Decision Tree έχει μια δομή δέντρου που παίρνει την μορφή ενός διαγράμματος ροής (Εικόνα 5.4), στο οποίο ο κάθε εσωτερικός κόμβος (κόμβος χωρίς «φύλλα» - nonleaf node) δηλώνει ένα test σε μια μεταβλητή, έπειτα κάθε κλάδος του δέντρου αντιπροσωπεύει το καθένα από τα αποτελέσματα του test και τέλος, κάθε κόμβος φύλλων (leaf node ή terminal node) εμπεριέχει μια ετικέτα κλάσης. Ο κορυφαίος κόμβος σε ένα τέτοιο δέντρο ονομάζεται **ριζικός κόμβος**. Έτσι δημιουργείται μια κατηγοριοποίηση.



Εικόνα 5.4: Δέντρο απόφασης κατηγοριοποίησης – classification (Διάγραμμα ροής - flow chart) μιας βάσης δεδομένων iris. [5]

5.3.1 Ο τρόπος πρόβλεψης του μοντέλου Decision Tree

Για τον τρόπο πρόβλεψης του δέντρου απόφασης θα γίνει ανάλυση της Εικόνας 5.4. Υποθετικά θα γίνει μια ταξινόμηση ενός λουλουδιού iris από την βάση δεδομένων. Ξεκινώντας από τον ριζικό κόμβο (βάθος 0, ο κόμβος στην κορυφή) στον οποίο γίνεται ο έλεγχος στο λουλούδι αν είναι μικρότερο ή ίσο από 2,45 cm, εάν είναι τότε μετακινείτε προς τα κάτω στον αριστερό θυγατρικό κόμβο της ρίζας (βάθος 1, αριστερά), άρα είναι και κλάση setosa. Στην περίπτωση όμως που ο ριζικός κόμβος είναι leaf node (δηλαδή δεν έχει άλλους θυγατρικούς κόμβους) τότε κάνει αμέσως την πρόβλεψη ότι το λουλούδι αυτό είναι κλάση setosa σύμφωνα με αυτόν τον κόμβο. Αν το λουλούδι όμως είναι μεγαλύτερο από 2,45 cm, τότε θα πρέπει να μετακινηθεί προς τα κάτω στον δεξί θυγατρικό κόμβο της ρίζας (βάθος 1, δεξιά), εάν αυτός ο κόμβος δεν είναι leaf node, τότε κάνει έναν ακόμα έλεγχο εάν το πλάτος του πετάλου είναι μικρότερο ή ίσο με 1,75 cm. Εάν είναι τότε είναι πιθανότατα ένα λουλούδι κλάσης versicolor (βάθος 2, αριστερά), εάν δεν είναι τότε είναι πιθανότατα ένα λουλούδι κλάσης virginica (βάθος 2, δεξιά).

Η κατηγορία **«δείγματα»** ενός κόμβου φανερώνει πόσες περιπτώσεις (σε αυτήν την περίπτωση πόσα λουλούδια) ελέγχτηκαν σε αυτόν τον κόμβο. Δηλαδή στον ριζικό κόμβο έγινε έλεγχος 150 δειγμάτων, στα οποία τα 50 είναι μικρότερα ή ίσα από 2,45 cm οπότε κατατέθηκαν στον θυγατρικό κόμβο με βάθος 1 αριστερά ενώ τα υπόλοιπα κατατέθηκαν στον θυγατρικό κόμβο με βάθος 1 δεξιά.

Η κατηγορία **«τιμή»** ενός κόμβου φανερώνει πόσες περιπτώσεις λουλουδιών που εκπαιδεύτηκαν σε αυτόν τον κόμβο ανήκουν στην κάθε κλάση. Για παράδειγμα ο κόμβος στο βάθος 2 αριστερά (δηλαδή ο κάτω αριστερά κόμβος) έχει εκπαιδεύσει 0 λουλούδια setosa, 49 λουλούδια versicolor και 5 λουλούδια virginica.

Τέλος, η κατηγορία **«gini»** του κόμβου μετρά κατά πόσο «καθαρός» είναι ένας κόμβος. Ένας κόμβος θεωρείτε καθαρός όταν όλες οι περιπτώσεις λουλουδιών που εκπαιδεύονται στον συγκεκριμένο κόμβο ανήκουν στην ίδια κλάση (τότε το gini ισούται με 0). Παραδείγματος χάρη, στον κόμβο με βάθος 1 αριστερά γίνεται η παρατήρηση πως έχει τιμή [50,0,0] άρα έχει εκπαιδεύσει μόνο λουλούδια setosa οπότε και η βαθμολογία στο gini είναι 0 και θεωρείται καθαρός. Ο υπολογισμός του gini γίνεται από την ακόλουθη εξίσωση:

$$G_i = 1 - \sum_{k=1}^{n} P_{i,k^2} \qquad \text{(Σχέση 5.9)}$$

Στον οποίο το $P_{i,k}$ υπολογίζεται από το k το οποίο είναι ο λόγος των περιπτώσεων της κάθε κλάσης μεταξύ του συνόλου της εκπαίδευσης του κόμβου. Για τον υπολογισμό του παραδείγματος τις Εικόνας 5.4 στο gini του κόμβου με βάθος 2 δεξιά γίνεται με τον εξής τρόπο: $1 - \left(\frac{0}{46}\right)^2 - \left(\frac{1}{46}\right)^2 - \left(\frac{45}{46}\right)^2 \approx 0{,}042633 \approx 0{,}043.$

5.3.1.2 Μέθοδος καθορισμού Gini ή Entropy?

Στο μοντέλο Decision Tree από προεπιλογή χρησιμοποιείται το μέτρο καθορισμού Gini, όμως μπορεί αυτό να αλλαχτεί και να επιλεχτεί το μέτρο καθορισμού Entropy, αλλάζοντας το κριτήριο του συγκεκριμένου υπερπαραμέτρου σε «Entropy». Ο προορισμός αυτού του ορισμού έχει προέλθει από τη θερμοδυναμική ως μέτρο μοριακής διαταραχής, δηλαδή η τιμή αυτή πλησιάζει στο μηδέν όταν τα μόρια είναι ακίνητα και καλά ταξινομημένα. Στην μηχανική μάθηση η εντροπία (entropy) ως μέτρο καθορισμού, δηλαδή η εντροπία ενός συνόλου είναι μηδέν όταν περιέχει περιπτώσεις μόνο μιας κλάσης. Ο υπολογισμός αυτής της κατηγορίας γίνεται με την ακόλουθη εξίσωση (Σχέση 5.10).

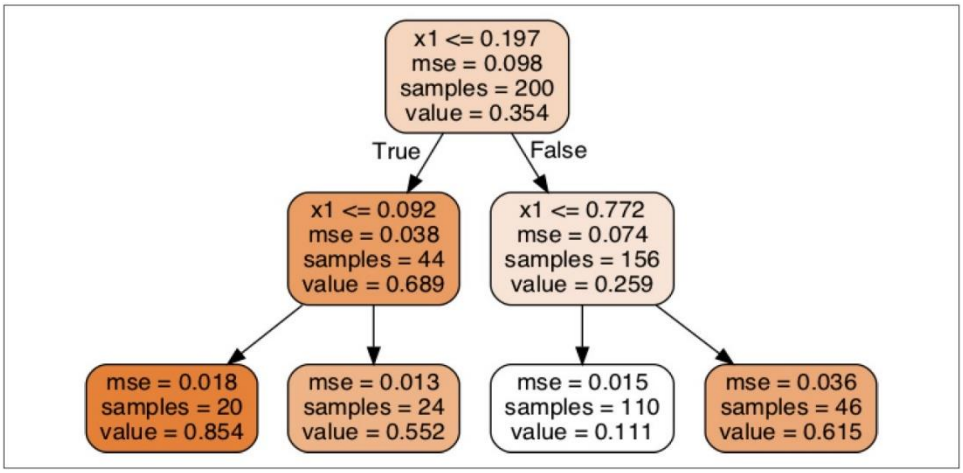
$$H_i = -\sum_{\substack{k=1 \\ P_{i,k} \neq 0}}^{n} P_{i,k} log_2\left(P_{i,k}\right) \qquad \text{(Σχέση 5.10)}$$

Για παράδειγμα, θα γίνει υπολογισμός από την Εικόνα 5.4 για την τιμή entropy του κόμβου με βάθος 2 αριστερά: $-\left(\frac{49}{54}\right) log_2\left(\frac{49}{54}\right) - \left(\frac{5}{54}\right) log_2\left(\frac{5}{54}\right) \approx 0.445.$

Οπότε ποια είναι η καλύτερη επιλογή; Η διαφορές μεταξύ των δυο μεθόδων είναι ελάχιστες, άρα τις περισσότερες φορές δεν κάνει μεγάλη διαφορά, καθώς οδηγούν σε παρόμοια δέντρα. Η μέθοδος καθορισμού Gini είναι ελαφρώς πιο γρήγορα για υπολογισμό, άρα την κάνει αυτομάτως μια πολύ καλή επιλογή. Παρόλο αυτά, εκεί που διαφέρουν είναι πως η μέθοδος Gini τείνει να απομονώνει την πιο συχνή κλάση σε ένα δικό της κλαδί του δέντρου, ενώ από την άλλη μεριά η μέθοδος καθορισμού Entropy τείνει να παράγει ελαφρώς πιο ισορροπημένα δέντρα.

5.3.2 Παλινδρόμηση – Regression

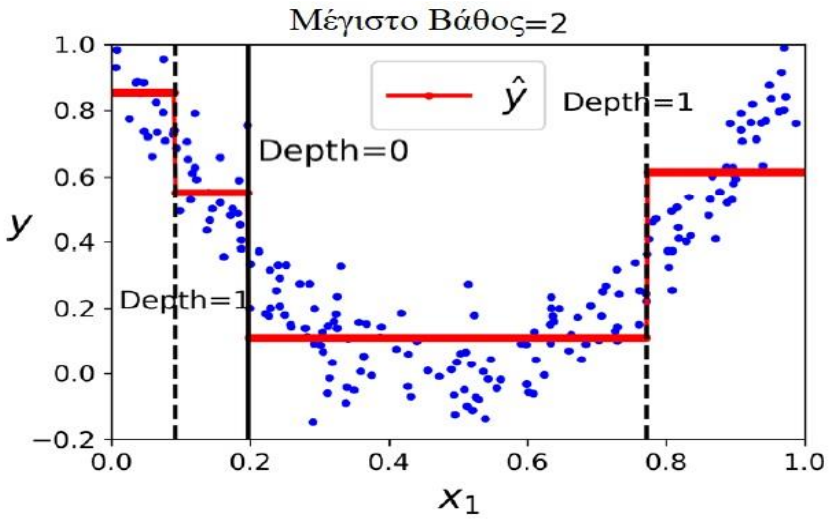
Τα δέντρα απόφασης είναι αρκετά ικανά να εκτελέσουν και εργασίες για παλινδρόμηση (Regression).



Εικόνα 5.5: Δέντρο Απόφασης για παλινδρόμηση (Regression). [5]

Το συγκεκριμένο δέντρο απόφασης μοιάζει αρκετά με αυτό της κατηγοριοποίησης (Classification) Εικόνα 5.4 καθώς, η λογική και η διαδικασία τους είναι ίδια. Η διαφορά τους είναι πως αντί να προβλέπει μια κλάση σε κάθε κόμβο προβλέπει μια τιμή (value). Στην συνέχεια θα γίνει πρόβλεψη για μια υποθετική εγγραφή με $x_1 = 0.6$. Ξεκινώντας από τον ριζικό κόμβο γίνεται ο πρώτος έλεγχος στο x_1 και συμπεράνει πως είναι μεγαλύτερο από αυτό του ελέγχου, άρα συνεχίζει τον έλεγχο του στον κόμβο με βάθος 1 δεξιά και τέλος στον κόμβο με βάθος 2 αριστερά. Οπότε προβλέπει την τιμή 0.111 για αυτή τη περίπτωση. Αυτή η πρόβλεψη που πραγματοποιήθηκε είναι η μέση τιμή που έχει οριστεί ως στόχος από το σύνολο των 110 περιπτώσεων που εκπαιδεύτηκαν σε αυτόν τον κόμβο. Τέλος, γίνεται πρόβλεψη για το Μέσο Τετραγωνικό Σφάλμα (MSE – Mean Square Error) που είναι ίσο με 0.015 για αυτά τα 110 δείγματα που χρησιμοποιήθηκαν.

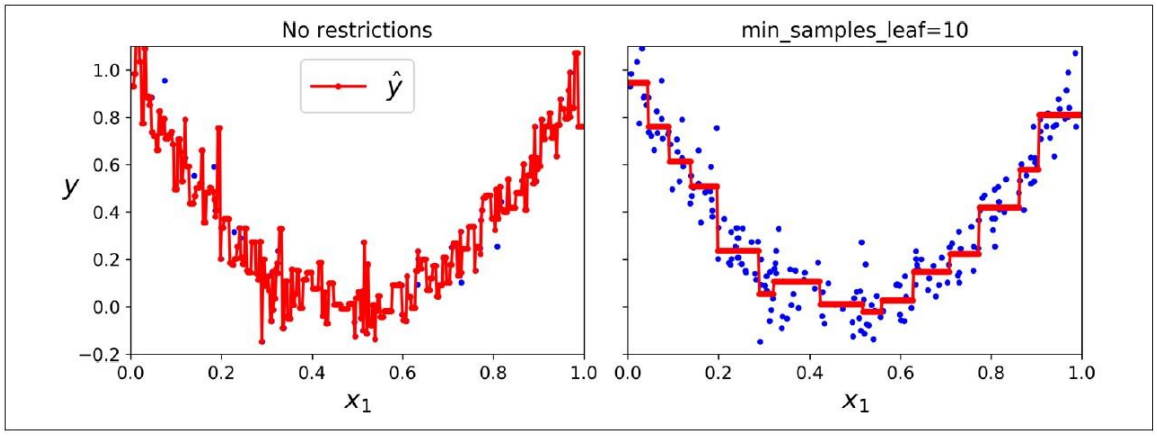
Οι προβλέψεις αυτού του μοντέλου απεικονίζονται στην Εικόνα 5.6 για μέγιστο βάθος κόμβων 2.



Εικόνα 5.6: Πρόβλεψη του μοντέλου Decision Tree Regression. [5]

Γίνεται η παρατήρηση πως το predicted value για κάθε περιοχή είναι το average target value (δηλαδή η μέση τιμή στόχος) των περιπτώσεων για την κάθε περιοχή. Ο αλγόριθμος αυτό καταφέρνει να χωρίσει την κάθε περιοχή έτσι ώστε να κάνει το training όσο το δυνατόν πιο κοντά στην τιμή πρόβλεψης.

Όπως κάθε αλγόριθμος έτσι και τα δέντρα απόφασης με παλινδρόμηση είναι επιρρεπή σε overfitting (3.5.1). Αυτό μπορεί να λυθεί με την **ομαλοποίηση** (regularization), δηλαδή αλλάζοντας τους προεπιλεγμένους υπερπαραμέτρους. Ένα καλό παράδειγμα για την λύση του overfitting σε αυτήν την περίπτωση είναι η ρύθμιση του min_samples_leaf ίσο με 10 (με αυτόν τον τρόπο γίνεται η επιλογή των δειγμάτων πιο λογική) έτσι τα δεδομένα δεν θα υπερβαίνουν το training set και θα υπάρξει σαν αποτέλεσμα ένα πιο λογικό μοντέλο.



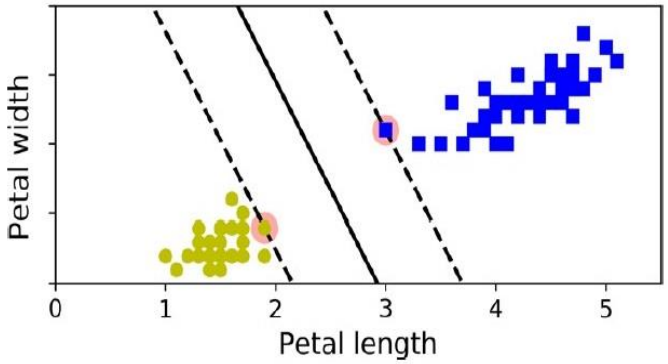
Εικόνα 5.7: Πρόβλεψη του δέντρου απόφασης χωρίς κάποιον περιορισμό (αριστερά), με ρύθμιση ενός υπερπαραμέτρου (δεξιά). [5]

5.4 Μηχανές Διανυσμάτων Υποστήριξης - Support Vector Machines (SVM)

Το Support Vector Machine είναι ένα ισχυρό και ευέλικτο μοντέλο της μηχανικής μάθησης ικανό να εκτελέσει γραμμική (linear) ή μη γραμμική (nonlinear) κατηγοριοποίηση (classification), παλινδρόμηση (regression) ακόμα και outlier detection. Τα μοντέλα SVM είναι πιο κατάλληλα για την σύνθετων μικρών ή μεσαίων σε σύνολο βάσης δεδομένων.

5.4.1 Γραμμικό μοντέλο SVM κατηγοριοποίησης – Linear SVM Classification

Για την καλύτερη περιγραφή του μοντέλου αυτού θα ληφθούν δυο υποθετικές περιπτώσεις γραμμικών διαχωρίσιμων κλάσεων (οι οποίες δηλαδή μπορούν να διαχωριστούν εύκολα με μια ευθεία γραμμή). Είναι όμως αναμενόμενο πως δεν υπάρχει μόνο μια λύσει στο πρόβλημα αυτό (δηλαδή τα δεδομένα αυτά μπορούν να διαχωριστούν σε αμέτρητες ευθείες). Έτσι στο πρόβλημα αυτό δίνει την λύση το κριτήριο αξιολόγησης των λύσεων το οποίο ονομάζεται **περιθώριο κατηγοριοποίησης** (margin classification – Εικόνα 5.8). Επίσης, ο διαχωρισμός των γραμμών γίνεται ώς εξής: Σε 2 διαστάσεις η γραμμική συνάρτηση διαχωρισμού είναι μια διαχωριστική ευθεία, σε 3ης διαστάσεις είναι ένα διαχωριστικό επίπεδο και σε περισσότερες από 3ης διαστάσεις είναι ένα διαχωριστικό υπερεπίπεδο (**hyperplane**).



Εικόνα 5.8: Κατηγοριοποίηση μεγάλου περιθωρίου (Large margin classification). [5]

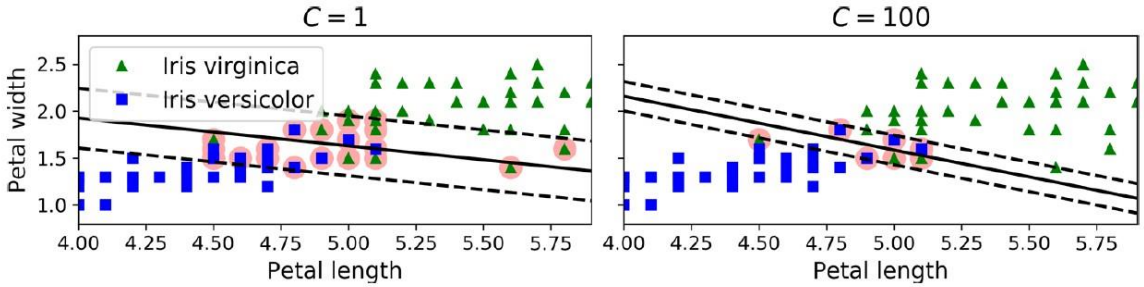
Η σταθερή γραμμή στο κέντρο στην Εικόνα 5.8 αντιπροσωπεύει το όριο του αποτελέσματος ενός μοντέλου κατηγοριοποίησης SVM. Αυτή η γραμμή είναι η πιο ιδανική για αυτά τα δεδομένα, καθώς διαχωρίζει τις δυο κλάσεις στο έμπρακτο και είναι αρκετά «μακριά» από τα training.

5.4.1.1 Κατηγοριοποίηση Soft ή Hard Margin

Για να μπορεί να λειτουργήσει σωστά η μέθοδος της κατηγοριοποίηση σκληρού περιθωρίου (Hard Margin) θα πρέπει τα δεδομένα είναι διαχωρισμένα μόνο γραμμικά, επίσης είναι αρκετά ευαίσθητη σε ακραίες τιμές (outliers).

Οπότε για να αποφευχθεί ένα πρόβλημα όπως τα λάθη που μπορεί να γίνουν από ακραίες τιμές, γίνεται χρήση ενός πιο ευέλικτου μοντέλου. Ο στόχος του είναι να βρεθεί μια καλή ισορροπία μεταξύ της διατήρησης του δρόμου (δηλαδή της μεσαίας γραμμής) όσο το δυνατόν μεγαλύτερο και του περιορισμού των παραβιάσεων των περιθωρίων – margin violations (όπως είναι τιμές που καταλήγουν στη μέση του δρόμου ή ακόμα και στην λάθος πλευρά). Αυτή η μέθοδος ονομάζεται μαλακή κατηγοριοποίηση περιθωρίων (soft margin classification).

Κατά την δημιουργία ενός μοντέλου SVM μπορεί να καθοριστεί ένας υπερπαράμετρος (hyperparameter) με όνομα C και να βοηθήσει στον περιορισμό των παραβιάσεων του περιθωρίου.



Εικόνα 5.9: Εναλλαγή του υπερπαραμέτρου C σε μικρή τιμή (αριστερά) ή σε μεγάλη τιμή (δεξιά). [5]

Χρειάζεται ιδιαίτερη προσοχή ο ορισμός αυτής της τιμής, καθώς αν οριστεί πολύ χαμηλά τότε υπάρχει πιθανότητα να μην έχει τα επιθυμητά αποτελέσματα. Αν όμως οριστεί με υψηλή τιμή τότε υπάρχει πιθανότητα να υπάρχουν αρκετά κακά χωρισμένες παραβιάσεις περιθωρίου (όπως είναι στην στα δεξιά στην Εικόνα 5.9). Η καλύτερη εκπαίδευση ανάμεσα σε αυτές τις δυο (από την Εικόνα 5.9) είναι αυτή στα αριστερά, καθώς μπορεί να αρκετές παραβιάσεις περιθωρίου, όμως υπάρχει μια καλύτερη γενίκευση στον διαχωρισμό. Τέλος, εάν το μοντέλο SVM έχει ενδείξεις ότι έχει πάθει overfitting τότε η καλύτερη επιλογή είναι η μείωση της τιμής στο C.

5.4.2 Μη γραμμική μέθοδος κατηγοριοποίησης SVM

Αν και το γραμμικό μοντέλο κατηγοριοποίησης SVM (5.4.1) είναι αρκετά εύκολο στην χρήση και λειτουργεί εκπληκτικά σε πάρα πολλές περιπτώσεις, σε πολλά data set όμως δεν είναι το ιδανικότερο καθώς δεν πλησιάζουν καν κάποιον γραμμικό διαχωρισμό. Μια καλή προσέγγιση για τον χειρισμό αυτών των μη γραμμικών συνόλων δεδομένων είναι η προσθήκη περισσότερων παραμέτρων στον αλγόριθμο, όπως είναι οι πολυωνυμικές παράμετροι. Σε αρκετές περιπτώσεις αυτή η τακτική μπορεί να οδηγήσει σε έναν γραμμικό διαχωρισμό αυτών των δεδομένων.

5.4.2.1 Polynomial Kernel

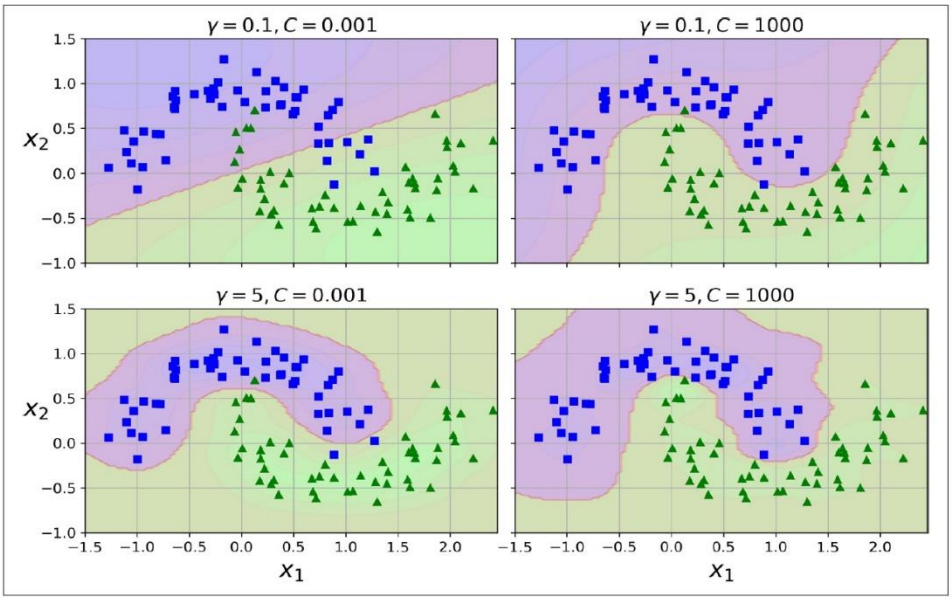
Η προσθήκη πολυωνυμικών παραμέτρων είναι κάτι απλό για να εφαρμοστεί, όμως μπορεί να βοηθήσει οποιονδήποτε τύπο αλγορίθμων μηχανικής μάθησης εκτός από το SVM. Εάν ο βαθμός του πολυωνύμου όμως είναι χαμηλός αυτή η μέθοδος δεν μπορεί να αντιμετωπίσει πολύ περίπλοκα σε σύνολο δεδομένα, ενώ με υψηλό βαθμό του πολυωνύμου τότε δημιουργεί έναν τεράστιο αριθμό παραμέτρων, με αποτέλεσμα το μοντέλο να είναι υπερβολικά αργό.

Την λύση σε αυτό το πρόβλημα όταν χρησιμοποιείτε η μέθοδος SVM την δίνει μια μαθηματική τεχνική, η οποία ονομάζεται **kernel trick**. Το κόλπο αυτό καταφέρνει να παρέχει καλά αποτελέσματα σαν να έχει γίνει η χειροκίνητη προσθήκη πολλών πολυωνυμικών παραμέτρων, ακόμα και με υψηλού βαθμού πολυώνυμα (Σε έναν αλγόριθμο αυτό το κόλπο χρησιμοποιείται όταν γίνεται εισαγωγή η κλάση SVC και λειτουργεί αλλάζοντας τον υπερπαράμετρο r, επηρεάζοντας έτσι γραμμή διαχωρισμού των δεδομένων).

Ένα kernel trick είναι η συνάρτηση **Gaussian RBF** (Radial Basis Function). Η συνάρτηση αυτή στην ουσία λειτουργεί με τα σημεία x (landmarks) της [Σχέσης 5.11], δηλαδή η προσέγγιση της είναι να δημιουργηθεί ένα σημείο για την κάθε περίπτωση από το σύνολο των δεδομένων. Αυτό έχει σαν αποτέλεσμα να δημιουργηθούν πολλές διαστάσεις (dimensions) και να αυξηθούν οι πιθανότητες για το μετασχηματισμένο training set να είναι γραμμικά διαχωρίσιμο. Στην συνάρτηση αυτή ο μαθηματικός της ορισμός είναι ο εξής:

$$\Phi_\gamma(x, l) = exp(-\gamma\|x - l\|^2) \qquad \text{(Σχέση 5.11)}$$

Στην οποίο το γ είναι η υπερπαράμετρος του μοντέλου η οποία εάν είναι αυξημένη καθιστά την καμπύλη που είναι σε σχήμα καμπάνας (bell-shaped) αρκετά πιο στενή (Εικόνα 5.10 αριστερό μέρος), με αποτέλεσμα το εύρος επιρροής της κάθε τιμής να είναι μικρότερο. Από την άλλη μεριά μια μειωμένη τιμή του γ καθιστά την καμπύλη bell-shaped μεγαλύτερη (Εικόνα 5.10 μέρος δεξιά), το οποίο έχει σαν αποτέλεσμα οι τιμές να έχουν μεγαλύτερο εύρος επιρροής και το όριο αποφάσεων τους (decision boundary) να καταλήγει πιο ομαλό.



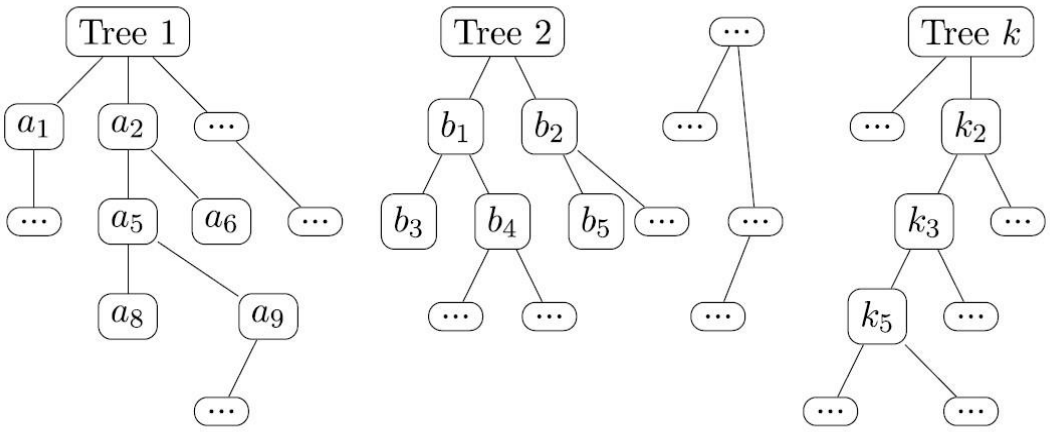
Εικόνα 5.10: Μοντέλο κατηγοριοποίησης SVM χρησιμοποιώντας RBF kernel. [5]

Με αυτόν τον τρόπο, το γ λειτουργεί σαν υπερπαράμετρος κανονικοποίησης. Τέλος, εάν το μοντέλο εμφανίζει overfitting η καλύτερη λύση είναι να μειωθεί η τιμή στο γ, ενώ εάν εμφανίζει underfitting θα πρέπει να αυξηθεί.

5.5 Τυχαίο Δάσος - Random Forest

Το Random Forest όπως φαίνεται και από το όνομα του, αποτελείται από ένα σύνολο με δέντρα απόφασης. Το οποίο εκπαιδεύεται μέσω της μεθόδου τοποθέτησης (Bagging method, ή άλλες φορές με την μέθοδο της επικόλλησης «pasting»), συνήθως με τα μέγιστα δείγματα τα οποία έχουν οριστεί από το training set. Γενικότερα, το Random Forest είναι αρκετά κοινό με το Decision Tree όσο στην δομή του όσο και στους υπερπαραμέτρους του.

Το μοντέλο αυτό εισάγει ένα επιπλέον χαρακτηριστικό κατά την καλλιέργεια (growing) των δέντρων απόφασης, το οποίο είναι το **randomness**. Στην ουσία, αντί να ψάχνει το καλύτερο χαρακτηριστικό για τον διαχωρισμό του κόμβου, αναζητά το καλύτερο χαρακτηριστικό ανάμεσα σε ένα τυχαίο υποσύνολο αυτών. Ο αλγόριθμος αυτός για να μπορέσει να αποδώσει ένα καλύτερο μοντέλο, ανταλλάσσει το υψηλό bias για μια μικρότερη διακύμανση (variance).



Εικόνα 5.11: Η βασική ιδέα του Random Forest. [11]

5.5.1 Extra-Trees

Στην καλλιέργεια ενός δέντρου μέσα στο Random Forest, σε κάθε κόμβο (node) του δέντρου μόνο ένα τυχαίο υποσύνολο των χαρακτηριστικών μπορεί να θεωρηθεί πως χωρίζεται. Επίσης, είναι εφικτό τα δέντρα αυτά να γίνουν ακόμα πιο τυχαία κάνοντας χρήση των τυχαίων κατώτατων ορίων (random thresholds) αντί να γίνεται η εύρεση των καλύτερων δυνατών ορίων (όπως συμβαίνει με το Decision Tree). Ένα δάσος το οποίο εμπεριέχει τέτοια τυχαία δέντρα ονομάζεται: Άκρως Τυχαιοποιημένα Δέντρα (Extremely Randomized Trees ή Extra Trees για συντομία).

Τα Extra-Trees είναι μια αρκετά συνηθισμένη επιλογή καθώς εκπαιδεύονται πιο γρήγορα από τα κανονικά τυχαία δάση, επειδή η εύρεση του καλύτερου δυνατού ορίου για κάθε

χαρακτηριστικό σε κάθε κόμβο είναι μια από τις πιο χρονοβόρες εργασίες στην ανάπτυξη ενός δέντρου.

5.5.2 Εκπαίδευση του Random Forest

Σε ένα σύνολο χαρακτηριστικών D (διαστάσεις του συνόλου training set), γίνεται χρήση ενός παραμέτρου m για να χωριστούν τα δεδομένα σε κάθε διαχωρισμό αποφάσεων (decision split). Ο προσδιορισμός του m γίνεται με τον ακόλουθο τρόπο (στρογγυλεμένο προς τον πλησιέστερο ακέραιο):

$$m = \left[\sqrt{D}\right] \text{ ή } m = [\log_2 D] \qquad \text{Σχέση 5.12}$$

Για κάθε υποσύνολο χαρακτηριστικών m τα δείγματα εκπαίδευση (training samples) επιλέγονται τυχαία από το αρχικό σύνολο δεδομένων με την μέθοδο της αντικατάστασης. Στην συνέχεια, το κάθε υποσύνολο τροφοδοτείται σε έναν κατηγοριοποιήτη του δέντρου απόφασης (decision tree classifier) και η κατηγοριοποίηση του βασίζεται στην πλειοψηφία του συνόλου αυτού. Τέλος, κάθε δέντρο δίνει μόνο μια «ψήφο» για την πιο δημοφιλή κλάση σε ένα δεδομένο διάνυσμα χαρακτηριστικών που έχουν εισαχτεί.

5.5.2.1 Περιπτώσεις Overfitting

Παρόλο που τα τυχαία δάση είναι πολύ δύσκολο να εμφανίσουν overfitting, ωστόσο αυτό γίνεται όλο και πιο πιθανό όταν τα δεδομένα εμπεριέχουν αρκετό **θόρυβο**. Ώστε να μπορέσει να λυθεί αυτό το πρόβλημα γίνεται η χρήση ορισμένων υπερπαραμέτρων. Για παράδειγμα ο πιο διαδεδομένος υπερπαράμετρος είναι τα πόσα δέντρα θα καλλιεργηθούν σε ένα δάσος. Συνήθως χρησιμοποιούνται μερικές δεκάδες έως και εκατοντάδες δέντρα για να γίνει σωστή η εκτίμηση (τα πιο συνηθής σύνολα είναι το 64, το 128 και το 256).

Β) Μέρος Δεύτερο

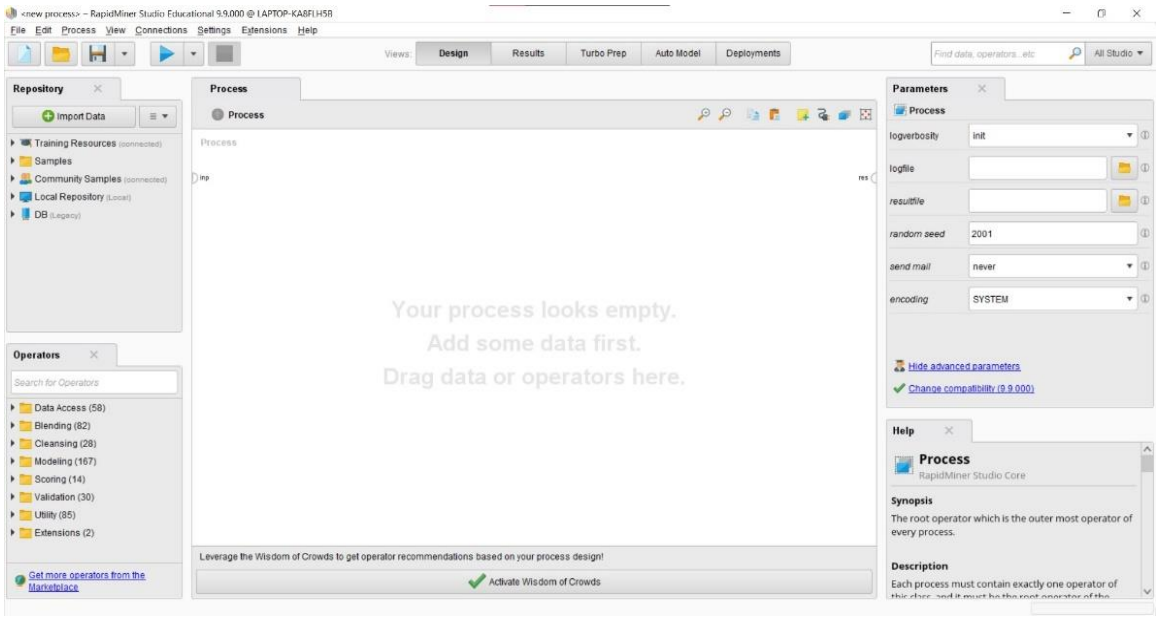
Κεφάλαιο 6: Προγραμματιστικά Εργαλεία

Για να είναι εφικτή η επεξεργασία ενός data set όπως και η δημιουργία των μοντέλων μετέπειτα, γίνεται αναγκαία η χρήση ορισμένων προγραμματιστικών εργαλείων. Υπάρχει πολύ μεγάλη ποικιλία από αυτά τα εργαλεία, άλλα βασίζονται σε πλατφόρμες (όπως είναι το RapidMiner, το KNIME, το Oracle και άλλα πολλά) και άλλα βασίζονται σε γλώσσες προγραμματισμού (όπως είναι η Python με το Jupyter Project (6.2) ή σε γλώσσα R με το RStudio).

Σε αυτήν την διπλωματική εργασία έγινε χρήση του **RapidMiner** (6.1) και της γλώσσα **Python** [με το Jupyter Notebook (6.2.1)] στα οποία γίνεται μια ανάλυση παρακάτω.

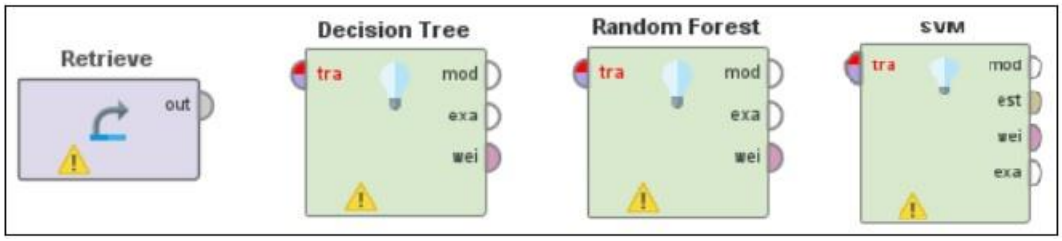
6.1 RapidMiner

Το RapidMiner είναι μια πλατφόρμα λογισμικού Data mining η οποία παρέχει ένα ολοκληρωμένο περιβάλλον για την προετοιμασία των δεδομένων, το Machine learning, το Deep learning και τις δημιουργίας προγνώσεων. Τέλος, το RapidMiner είναι αναπτυγμένο σε μοντέλο ανοιχτού πυρήνα (open core model).



Εικόνα 6.1: Το βασικό περιβάλλον τους προγράμματος.

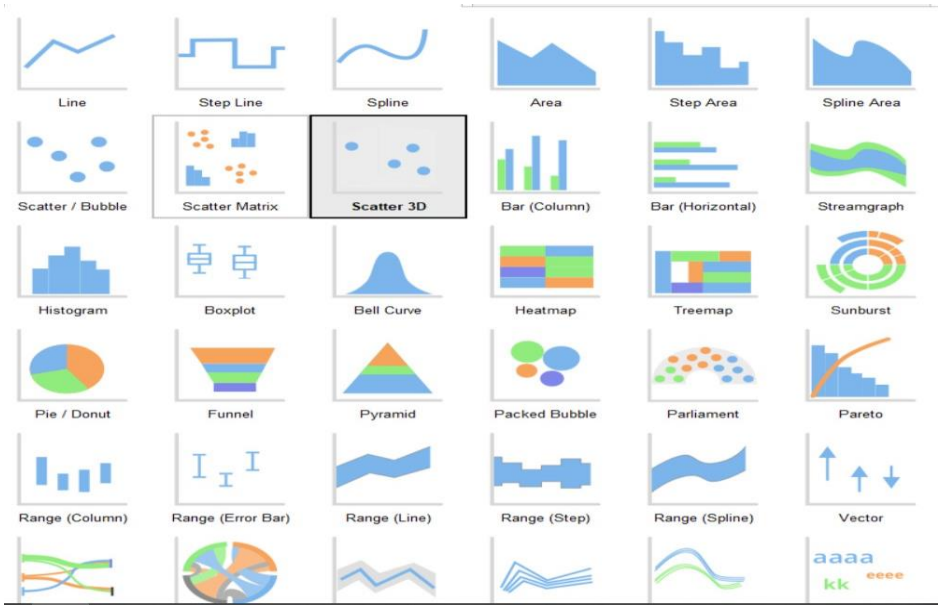
Η Εικόνα 6.1 φανερώνει το βασικό περιβάλλον του RapidMiner, στο οποίο διαδραματίζεται το Design (Σχεδίαση) όπου δημιουργούνται τα μοντέλα. Η δημιουργία επιτυγχάνεται με την χρήση διάφορων **Operators** (Χειριστήρια). Οι Operators (Εικόνα 6.2) εμπεριέχουν τον έτοιμο (σε Python ή R) κώδικα που με την βοήθεια του Drag & Drop ο διαχειριστής του μοντέλου μπορεί να σχεδιάσει ότι αυτός επιθυμεί.



Εικόνα 6.2: RapidMiner Operators.

Η σύνδεση των Operators γίνεται με καλώδια από μια έξοδο (δεξιά) σε μια είσοδο (αριστερά). Επίσης, μπορεί να γίνει σύνδεση σε περισσότερες από μια εισόδους με μια κοινή έξοδο, με την βοήθεια του χειριστηρίου Multiply.

Αφού γίνει η σχεδίαση του μοντέλου και εκτελεστεί η διαδικασία που έχει σχεδιαστεί, τότε στην καρτέλα Results (Αποτελέσματα) θα εμφανιστούν τα αποτελέσματα του μοντέλου του δημιουργήθηκε.



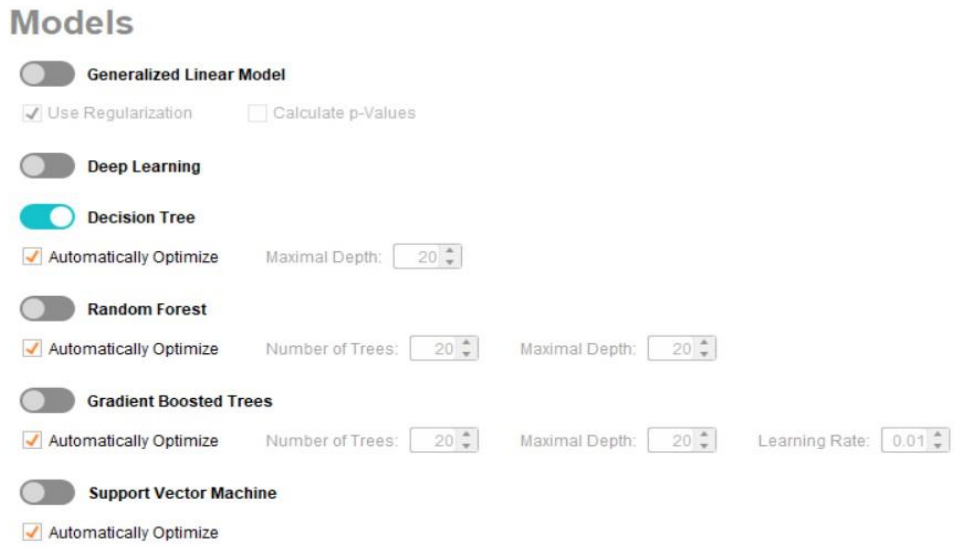
Εικόνα 6.3: RapidMiner chart selection.

Τέλος, δίνει την δυνατότητα από μια πληθώρα επιλογών (όπως φαίνεται και στην Εικόνα 6.3) για το διάγραμμα που θα επιλέξει να δημιουργήσει ο χρήστης για την κάθε περίσταση της ανάλυσης που επιθυμεί.

6.1.1 Auto Model

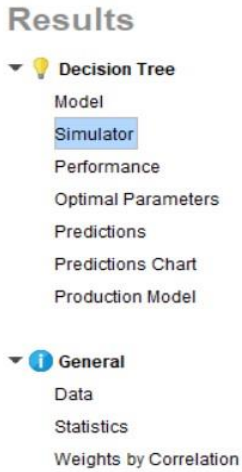
Η πλατφόρμα του RapidMiner παρέχει την δυνατότητα στον σχεδιαστή του μοντέλου έναν πιο γρήγορο και εύκολο τρόπο (όμως χωρίς δυνατότητες εναλλαγής αρκετών από τους υπερπαραμέτρους) για την δημιουργία ενός μοντέλου, αυτό είναι το **αυτόματο μοντέλο** (Auto Model).

Όπως σε όλα τα μοντέλα κατά την δημιουργία τους, αφού γίνει η εισαγωγή των δεδομένων και η επιλογή της μεταβλητής που θα γίνει στόχος (label) τότε επιλέγεται το μοντέλο που θα χρησιμοποιηθεί. Το Auto Model του RapidMiner προσφέρει μια μεγάλη ποικιλία από αυτά (Εικόνα 6.4).



Εικόνα 6.4: Επιλογή μοντέλου στο Auto Model.

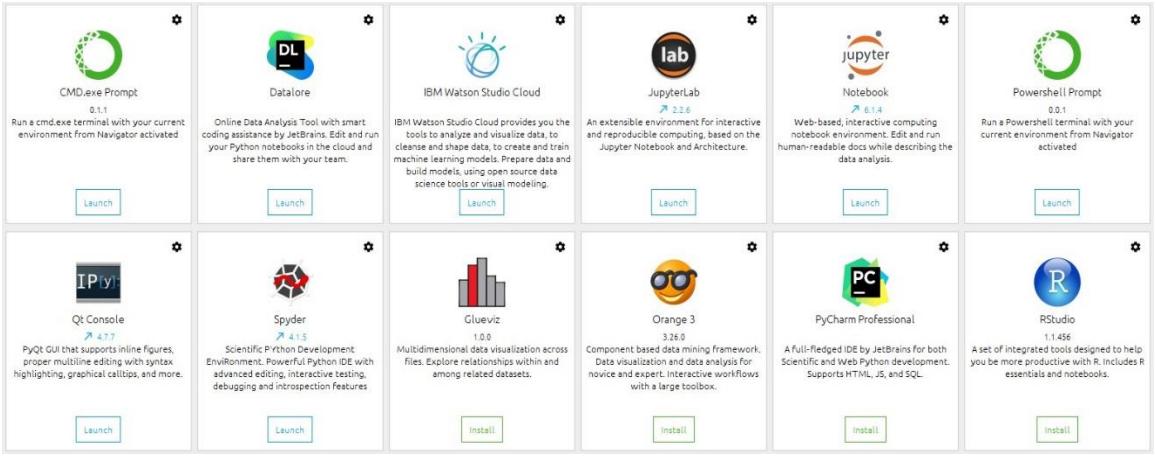
Αφού επιλεχτεί το μοντέλο και εκτελεστεί η διαδικασία, το Auto Model φανερώνει ένα πλήθος αποτελεσμάτων, όπως είναι το Performance, το Prediction Chart και διάφορα άλλα στατιστικά (Εικόνα 6.5).



Εικόνα 6.5: Καρτέλα αποτελεσμάτων του Auto Model.

6.2 Project Jupyter

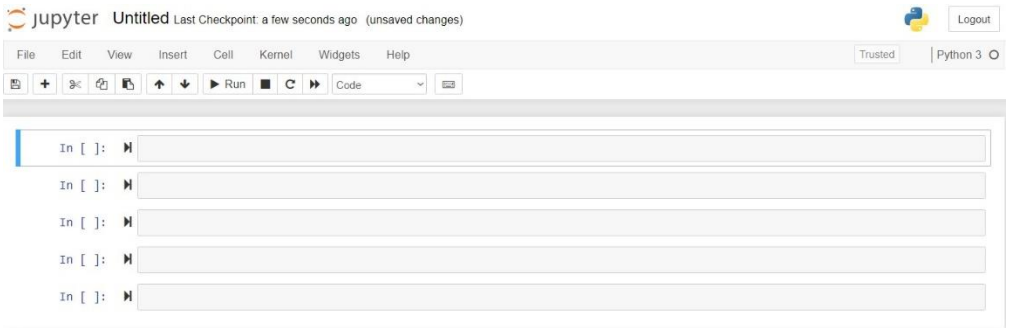
Το Project Jupyter είναι ένα project με στόχο την ανάπτυξη λογισμικού ανοιχτού κώδικα (open - source), ανοιχτών προτύπων (open - standards) και διάφορες άλλες υπηρεσίες σε δεκάδες γλώσσες προγραμματισμού. Το όνομα του προέρχεται από τις τρεις πιο βασικές γλώσσες προγραμματισμού που υποστηρίζονται από αυτό την Julian, την **Python** και την R. Τα κύρια προϊόντα που έχει αναπτύξει το project είναι το **Jupyter Notebook** (6.2.1), το JupyterHub και το JupyterLab.



Εικόνα 6.6: Jupyter Project προϊόντα χρησιμοποιώντας το πρόγραμμα Anaconda.

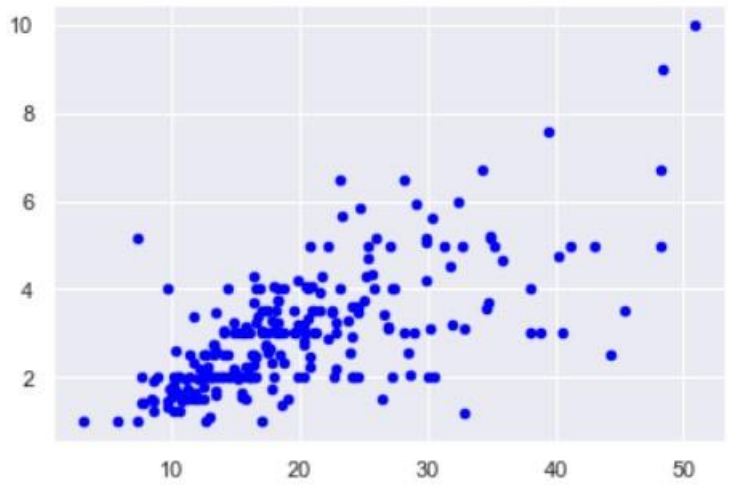
6.2.1 Jupyter Notebook

Το Jupyter Notebook είναι ένα διαδραστικό υπολογιστικό περιβάλλον βασισμένο σε web (ιστοσελίδα) για την δημιουργία αλγορίθμων.



Εικόνα 6.7: Το περιβάλλον του Jupyter Notebook.

Όπως παρουσιάζεται και στην Εικόνα 6.7 το περιβάλλον του αποτελείτε από κελία (cells) εισόδου / εξόδου τα οποία μπορούν να περιέχουν κώδικα, κείμενο, μαθηματικά ακόμα και γραφικές παραστάσεις (Εικόνα 6.8). Για να μπορέσουν να λειτουργήσουν όλα τα παραπάνω το πρόγραμμα παρέχει μια πληθώρα επιλογή βιβλιοθηκών, οι οποίες μπορούν να χρησιμοποιηθούν για την οποιαδήποτε απαιτούμενη χρήση του αλγορίθμου.



Εικόνα 6.8: Γράφημα από το Jupyter Notebook. [14]

Τέλος, με τον κατάλληλο κώδικα μπορεί να εξάγει τα αποτελέσματα σε διάφορες μορφές (όπως είναι σε HTML, PDF, EXEL, CVC, LaTeX και πολλά άλλα).

Κεφάλαιο 7: Προετοιμασία των Δεδομένων – Preprocessing

Για να μπορέσουν τα μοντέλα να δώσουν την καλύτερη δυνατή τους λύση και σε λιγότερο χρόνο, πρέπει να γίνουν ορισμένες ενέργειες στην βάση δεδομένων την οποία πρέπει να διαχειριστούν. Αυτές οι ενέργειες γίνονται στο κομμάτι της προετοιμασίας των δεδομένων. Όπως γίνεται αναφορά και στο Κεφάλαιο 4ο ακόμα και η πιο «τέλεια» βάση δεδομένων χρειάζεται να γίνει μια προετοιμασία, έτσι ώστε να αποτραπεί η χρήση των τιμών NaN (Not a Number – Χωρίς αριθμό), όπως και η αφαίρεση μη χρήσιμων μεταβλητών για τα μοντέλα.

7.1 Για το Data Set που χρησιμοποιήθηκε

Η βάση δεδομένων που χρησιμοποιήθηκε παρέχει μετρήσεις από 180 αισθητήρων (μεταβλητών) ενός πλοίου, σε βάθος χρόνου από τις 13 Φεβρουαρίου 2020 και ώρα 00:00:00 έως και τις 26 Ιουλίου 2020 και ώρα 00:00:00 (οι αισθητήρες λάμβαναν μετρήσεις ανά ένα λεπτό για όλο αυτό το διάστημα, οι οποίες δημιούργησαν ένα σύνολο 236.161 γραμμών).

7.2 Η διαδικασία της προετοιμασίας

Η διαδικασία της προετοιμασίας των δεδομένων εκτελέστηκε με χρήση του Jupyter Notebook (6.2.1) για την γλώσσα Python. Σε αυτήν την διπλωματική εργασία είναι **απαιτούμενο** η βάση δεδομένων που έχει χρησιμοποιηθεί να θεωρηθεί «τέλεια», δηλαδή η προεπεξεργασία να έχει ως στόχο την μείωση των εγγραφών και όχι των μεταβλητών.

Αρχικά, πρέπει να γίνει η εισαγωγή (και εκτέλεση) των βιβλιοθηκών (Εικόνα 7.1), από τις οποίες θα χρησιμοποιηθούν ορισμένες συναρτήσεις.

```
from sklearn import preprocessing
import seaborn as sns
import pandas as pd
```

Εικόνα 7.1 Εισαγωγή βιβλιοθηκών.

Στην συνέχεια γίνεται η εισαγωγή του Excel (Εικόνα 7.2).

```
df = pd.read_excel(r'C:\praktiko\Raw_data.xlsx')
```

Εικόνα 7.2: Εισαγωγή του excel στο πρόγραμμα

7.2.1 Εντολές κατανόησης

Αφού γίνουν αυτά τα δυο βασικά βήματα, στην συνέχεια θα πρέπει να εκτελεστούν ορισμένες εντολές, οι οποίες θα έχουν ως αποτέλεσμα την καλύτερη κατανόηση της βάσης δεδομένων τόσο στην δομή της όσο και στα περιεχόμενα της.

Ξεκινώντας με την εντολή head (Εικόνα 7.3) η οποία δείχνει ένα συγκεκριμένο ποσό γραμμών (το default του είναι οι 5 γραμμές όμως μπορεί να αλλάξει με οποιονδήποτε αριθμό μέσα στην παρένθεση) από το data set.

```
pd.set_option("display.max_columns", None) # thats a setting to display all columns
df.head()
```

	TIME	TIME_NUM	Longitudinal_Water_Speed	Transverse_Water_Speed	Longitudinal_Ground_Speed	Transverse_Ground_Speed	Stern_Transverse_W
0	13-Feb-20 00:00:00	43874.000000	0.44	0.00	0.00	-0.01	
1	13-Feb-20 00:01:00	43874.000694	0.12	-0.11	-0.01	0.02	
2	13-Feb-20 00:02:00	43874.001389	0.04	-0.03	0.01	0.01	
3	13-Feb-20 00:03:00	43874.002083	0.05	-0.19	0.00	-0.01	
4	13-Feb-20 00:04:00	43874.002778	0.02	-0.01	-0.01	0.03	

Εικόνα 7.3: Εντολή ανάλυσης των δεδομένων head.

Αυτή η εντολή βοηθάει στην ενημέρωση του δημιουργού του μοντέλου, ως προς τι δεδομένα υπάρχουν μέσα στις μεταβλητές. Ένας άλλος τρόπος να γίνει αυτό είναι με την εντολή info η οποία δείχνει τα τέσσερα πρώτα δεδομένα και τα τέσσερα τελευταία από κάθε μεταβλητή.

Για να εμφανιστεί το μέγεθος των γραμμών και της βάσης χρησιμοποιείται η εντολή shape (Εικόνα 7.4).

```
df.shape
```

```
(236161, 180)
```

Εικόνα 7.4: Μέγεθος γραμμών και μεταβλητών

Στην συνέχεια με την βοήθεια της εντολής columns (Εικόνα 7.5) εμφανίζονται στην έξοδο οι μεταβλητές ονομαστικά και το μέγεθος τους.

```
df.columns
```

```
Index(['TIME', 'TIME_NUM', 'Longitudinal_Water_Speed',
       'Transverse_Water_Speed', 'Longitudinal_Ground_Speed',
       'Transverse_Ground_Speed', 'Stern_Transverse_Water_Speed',
       'Stern_Transverse_Ground_speed', 'Total_Cumulative_Water_Distance',
       'Water_Distance_Since_Reset',
       ...
       'SG_Wind_direction', 'SG_Mean_Wave_Direction',
       'SG_Wind_Sea_Wave_Height', 'SG_Wind_Sea_wave_Period',
       'SG_Wind_Sea_Direction', 'SG_Swell_Wave_Height', 'SG_Swell_Wave_Period',
       'SG_Swell_Wave_Direction', 'SG_Current_Velocity',
       'SG_Current_Direction'],
      length=180)
```

Εικόνα 7.5: Εντολή columns.

Εάν χρειάζεται να είναι φανερές όλες οι μεταβλητές τότε μπορεί να γίνει η χρήση της παρακάτω εντολής (Εικόνα 7.6), η οποία εκτυπώνει στην έξοδο του κελιού όλες τις μεταβλητές.

```
print(df.columns.tolist())
```

Εικόνα 7.6: Εντολή για να εμφανιστούν όλες τις μεταβλητές.

Τέλος, με την χρήση της εντολής dtypes (Εικόνα 7.7) εμφανίζονται όλοι οι τύποι των μεταβλητών (float, object, int). Αυτή θα χρειαστεί καθώς στα μοντέλα γίνεται μόνο χρήση του Regression οπότε όλοι οι τύποι εκτός των αριθμητικών δεν χρειάζονται.

```
df.dtypes
```

```
TIME                        object
TIME_NUM                   float64
Longitudinal_Water_Speed   float64
Transverse_Water_Speed     float64
Longitudinal_Ground_Speed  float64
                             ...
SG_Swell_Wave_Height       float64
SG_Swell_Wave_Period       float64
SG_Swell_Wave_Direction    float64
SG_Current_Velocity        float64
SG_Current_Direction       float64
Length: 180, dtype: object
```

Εικόνα 7.7: Εντολή dtypes.

7.2.2 Εντολές Επεξεργασίας

Όπως έχει αναφερθεί και παραπάνω (7.2) η βάση δεδομένων που γίνεται χρήση είναι απαιτούμενο να θεωρείται ως τέλεια. Αυτό έχει σαν αποτέλεσμα να μην γίνει χρησιμοποιηθεί καμία μέθοδος η οποία να αλλάζει τα δεδομένα, έτσι, ούτε και η αλλαγή των outliers είναι αποδεκτή. Το μόνο που είναι αποδεκτό είναι η μείωση του μεγέθους της βάσης δεδομένων.

Επομένως, οι μέθοδοι που χρησιμοποιήθηκαν σε αυτήν την διαδικασία είναι από την μέθοδο Data Cleaning (4.3.4):

- Η πρώτη μέθοδος είναι η **διαγραφή ολόκληρης της μεταβλητής**. Για μεταβλητές οι οποίες δεν είναι καθόλου «χρήσιμες» (όπως είναι μια ημερομηνία) και για μεταβλητές οι οποίες είναι συνέχεια σταθερές χωρίς καμία αλλαγή (7.2.2.1).
- Η δεύτερη μέθοδος που χρησιμοποιήθηκε είναι η **αγνόηση ολόκληρης της γραμμής** (7.2.2.2).

7.2.2.1 Διαγραφή μεταβλητών

Αρχικά, θα πρέπει να γίνει η διαγραφή των μεταβλητών οι οποίες είναι τύπου object. Στην βάση αυτή η μόνη μεταβλητή που έχει αυτόν τον τύπο είναι η μεταβλητή TIME (1η μεταβλητή) και περιέχει ακριβή ημερομηνία (όπως επίσης είναι και τύπος object Εικόνα 7.7). Αυτό γίνεται με την ακόλουθη εντολή.

```
df.drop('TIME', inplace=True, axis=1) #delete a column
```

Εικόνα 7.8: Διαγραφή μεταβλητής TIME.

Στην εντολή αυτή η παράμετρος inplace έχει ως σκοπό να κάνει την διαγραφή «μόνιμη» ώστε να μην εμφανίζεται για τις επόμενες εντολές και το axis είναι 1 (ή columns) όταν αυτό που διαγράφετε είναι μεταβλητή, ενώ είναι 0 όταν αυτό που διαγράφετε είναι δείκτης.

Η επόμενη μεταβλητή που πρέπει να διαγραφεί είναι το Year, καθώς παρατηρώντας την βάση δεδομένων αυτή η μεταβλητή έχει μια μοναδική τιμή η οποία είναι το 2020 (Εικόνα 7.9).

Year
2020
2020
2020
2020
2020
2020
2020
2020
2020

Εικόνα 7.9: Περιεχόμενα μεταβλητής Year.

Οπότε, με τον ίδιο ακριβώς τρόπο με την μεταβλητή TIME γίνεται και η διαγραφή της μεταβλητής Year (Εικόνα 7.10).

```
df.drop('Year', inplace=True, axis=1)
```

Εικόνα 7.10: Διαγραφή της μεταβλητής Year.

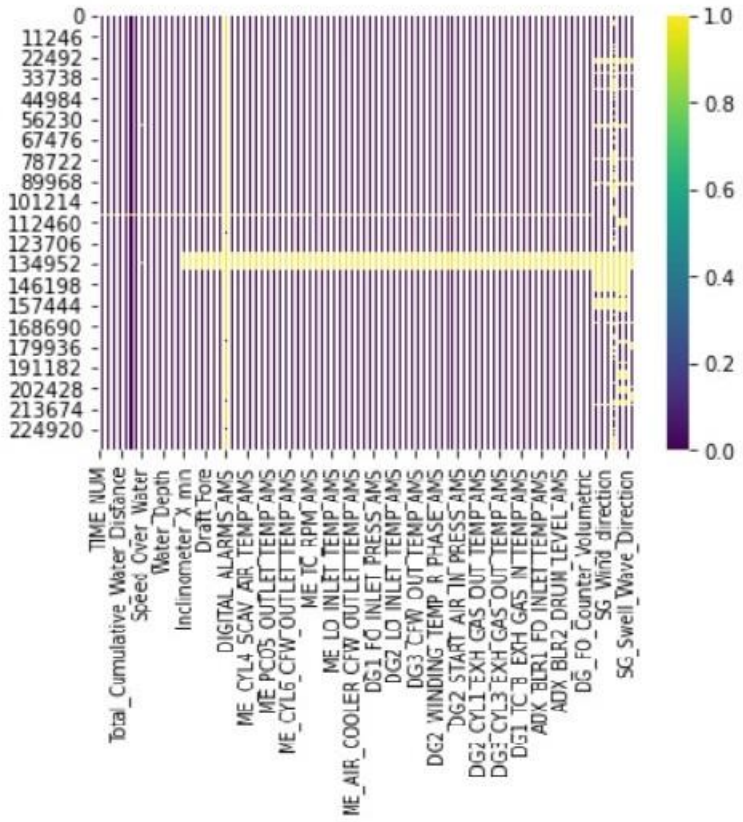
Αφού διαγραφεί η μεταβλητή Year θα πρέπει να αφαιρεθούν και οι μεταβλητές Month, Day και Time καθώς θεωρούνται «άχρηστες» χωρίς την μεταβλητή Year.

7.2.2.2 Διαγραφή της γραμμής

Αφού γίνει η διαγραφή των «άχρηστων» μεταβλητών το επόμενη βήμα είναι να βρεθούν οι γραμμές που έχουν missing values και να αφαιρεθούν τελείως από το data set. Για να μπορέσει να γίνει αυτό πρέπει πρώτα να εντοπιστούν τα Missing Values, το οποίο γίνεται πιο ξεκάθαρο σε ένα διάγραμμα (Εικόνα 7.11).

```
sns.heatmap(df.isnull(), cmap='viridis') #finding the NaN values
```

```
<AxesSubplot:>
```



Εικόνα 7.11: Εντολή για εντοπισμό των NaN τιμών.

Το διάγραμμα αυτό φανερώνει με κίτρινο χρώμα όλες τις τιμές μέσα στις μεταβλητές που εμπεριέχουν Missing Values.

Στην Εικόνα 7.11 γίνεται η παρατήρηση πως η μεταβλητή DIGITAL_ALARMS_AMS η οποία περιέχει τα ψηφιακά σήματα συναγερμού να είναι γεμάτη από Missing Values. Οπότε το πιο «ασφαλές» είναι να γίνει η αφαίρεση της (Εικόνα 7.12), εάν δεν γίνει αυτό τότε η επεξεργασμένη βάση δεδομένων θα είναι υπερβολικά μικρή, με αποτέλεσμα το νέο data set να μην αντικατοπτρίζεται ως μια μικρότερη βάση της αρχικής.

```
df.drop('DIGITAL_ALARMS_AMS', inplace=True, axis=1)
```

Εικόνα 7.12: Διαγραφή της μεταβλητής Digital Alarms Ams.

Επομένως, το τελευταίο βήμα της επεξεργασίας είναι να γίνει η αφαίρεση όλων των γραμμών οι οποίες εμπεριέχουν έστω και **ένα** missing value (αυτό μπορεί να αλλάξει ανάλογα με μια παράμετρο, παραδείγματος χάρη: με την παράμετρο thresh=2 ο αλγόριθμος αγνοεί όλες τις γραμμές που έχουν δυο missing values αντί για ένα). Αυτό πραγματοποιείται με την ακόλουθη εντολή (Εικόνα 7.13).

```
df.dropna(inplace=True)
df
```

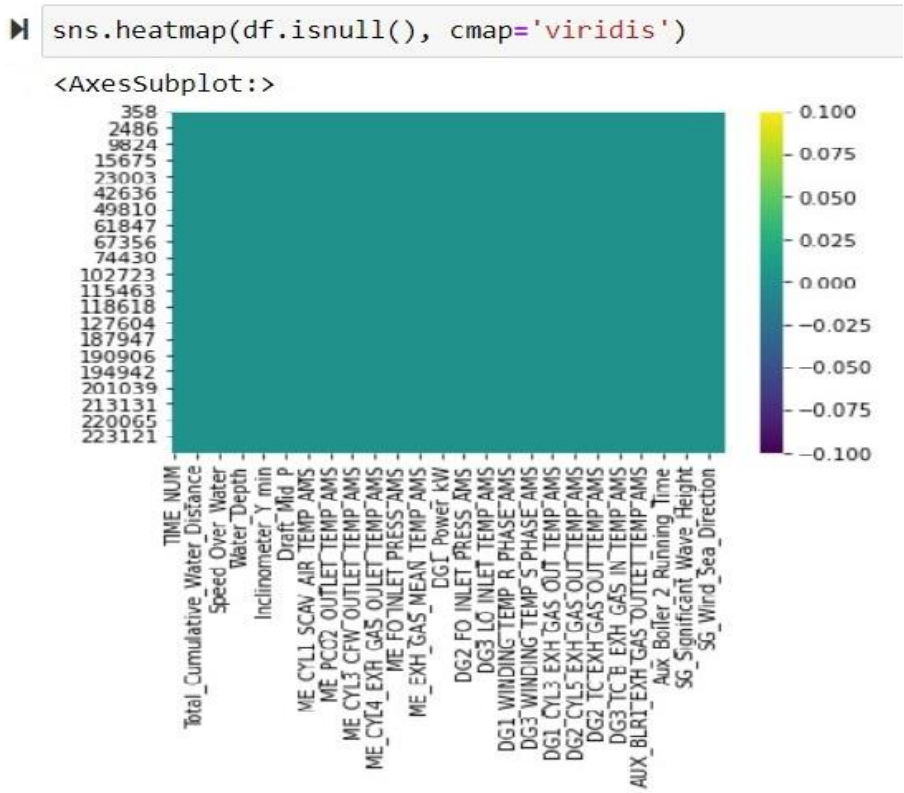
	TIME_NUM	Longitudinal_Water_Speed	Transverse_Water_Speed	Longitudinal_Ground_Speed	Transverse_Ground_Speed
358	43874.248611	-0.06	-0.01	0.01	0.03
359	43874.249306	-0.01	-0.23	0.00	-0.01
360	43874.250000	-0.18	-0.22	0.00	-0.01
361	43874.250694	-0.13	-0.20	0.00	0.00
362	43874.251389	0.11	-0.17	0.00	0.00
...	...	...	...	...	...
228596	44032.747222	10.13	0.25	10.13	0.25
228597	44032.747917	9.88	0.45	9.88	0.45
228598	44032.748611	9.79	0.31	9.79	0.31
228599	44032.749306	9.65	0.32	9.65	0.32
228600	44032.750000	9.67	0.30	9.67	0.30

39475 rows × 174 columns

Εικόνα 7.13: Αφαίρεση των γραμμών με τιμές NaN.

Όπως φαίνεται και στην Εικόνα 7.13 στο κάτω αριστερά μέρος, το νέο μέγεθος τις βάσης δεδομένων είναι στις 39.475 γραμμές και στις 174 μεταβλητές.

Οπότε, αφού ξαναγίνει ο έλεγχος (Εικόνα 7.14) για τα Missing Values και διαπιστωθεί ότι δεν υπάρχει καμία τιμή που να λείπει (άρα δεν υπάρχει και κανένα κίτρινο σημείο), τότε μπορεί να γίνει η εξαγωγή της βάσης δεδομένων.



Εικόνα 7.14: Έλεγχος για Missing Values.

Η εξαγωγή της νέας βάσης δεδομένων σε ένα νέο Excel πραγματοποιείται με την εξής εντολή (Εικόνα 7.15):

```
df.to_excel(r'C:\praktiko\Data_woNaN.xlsx', index = False)  #export the new data set
```

Εικόνα 7.15: Εξαγωγή της βάσης δεδομένων.

7.3 Εναλλακτική λύση

Παρόλο που η λύση που χρησιμοποιήθηκε με το Data Cleaning (4.3.4) και την αγνόηση της γραμμής (όπου υπάρχουν Missing Values) είναι αρκετά αποτελεσματική και γρήγορη, χάνει αρκετά σε ακρίβεια στα αποτελέσματα των μοντέλων. Μια καλή λύση για να γίνει η ακρίβεια όσο το δυνατόν καλύτερη είναι να γίνει η επιλογή της μεθόδου του PCA (Principal Components Analysis 4.3.5.1).

Η διαδικασία που θα έπρεπε να γίνει είναι η ακόλουθη:

Αρχικά, θα έπρεπε να γίνει με την μέθοδο του data normalization (4.3.2) μια αναδιάταξη στα missing values, είτε με min-max normalization (4.3.2.1), είτε με Z-Score normalization (4.3.2.2) (η πιο ασφαλείς επιλογή θα ήταν το Z-Score για την αποφυγή λάθους από ακραίες τιμές).

Έπειτα, η μέθοδος του PCA θα επιλέξει ποιες μεταβλητές είναι οι πιο σημαντικές για να κρατήσει και ποιες οι πιο ασήμαντες για να αφαιρέσει. Όμως, για να έχει μεγαλύτερη ακρίβεια, μέσα στις μεταβλητές αυτές θα υπάρχουν όλες οι γραμμές από το αρχικό data set. Αυτό έχει ως αποτέλεσμα τα μοντέλα να καταναλώνουν πολύ περισσότερο χρόνο και επεξεργαστική ισχύ, οπότε η συγκεκριμένη βάση δεδομένων το καθιστούσαν (το PCA) ανέφικτο ως λύση.

Κεφάλαιο 8: Δημιουργία των Μοντέλων

Αυτό το κεφάλαιο είναι το πιο σημαντικό σε όλη την εργασία, καθώς θα γίνει η δημιουργία και η ανάλυση ορισμένων από των κυριότερων μοντέλων που επικρατούν στην μηχανική μάθηση.

Ο **στόχος** του κεφαλαίου είναι να δημιουργηθούν και να εκπαιδευτούν διάφορα μοντέλα μηχανικής μάθησης με σκοπό την καλύτερη βελτιστοποίηση των μοντέλων από τους υπερπαραμέτρους τους.

Τα μοντέλα που θα δημιουργηθούν είναι τα εξής: Το Δέντρο Απόφασης (8.1), το Generalized Linear Model (8.2) και το Random Forest (8.3). Όλα τα μοντέλα δημιουργήθηκαν και επεξεργάστηκαν με την βοήθεια του RapidMiner (6.1).

Όλα τα μοντέλα (του Supervised Learning που χρησιμοποιήθηκαν) για να μπορέσουν να κάνουν **πρόβλεψη** (prediction) και να έχουν τα καλύτερα δυνατά αποτελέσματα, πρέπει να επιλεγεί μια μεταβλητή η οποία θα πάρει τον τύλο του Label (δηλαδή θα είναι η μεταβλητή «στόχος»). Από τη βάση δεδομένων με την οποία διαδραματίζονται τα μοντέλα, οι πιο σημαντικές μεταβλητές για τον σκοπό αυτό είναι το Fuel _ Rack _ Position _ AMS (δηλαδή η Θέση του κανόνα πετρελαίου), το ME _ FO _ Flow _ Mass (δηλαδή η Μέτρηση της ροής του πετρελαίου της κύριας μηχανής και έχει μονάδα μέτρησης το kgr/min) και το Speed _ Over _ Ground (δηλαδή η ταχύτητα του πλοίου ως προς το νερό – μονάδα μέτρησης είναι το Knots). Αυτές οι τρεις μεταβλητές είναι οι πιο σημαντικές καθώς ο στόχος της πρόβλεψης είναι η μείωση της κατανάλωσης καυσίμου του πλοίου. Η μεταβλητή η οποία επιλέχτηκε για να είναι ως στόχος σε όλα τα μοντέλα είναι η **ME _ FO _ Flow _ Mass**.

Η ανάλυση των μοντέλων που δημιουργήθηκαν γίνεται από τα εξής μέρη:

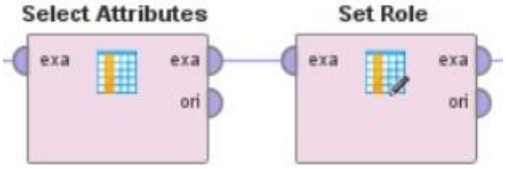
- Στον χρόνο (Execution time) που χρειάζεται να γίνει η εκτέλεσης τους.
- Από το Performance (πιο συγκεκριμένα από το testing set) του κάθε μοντέλου. Δηλαδή γίνεται ανάλυση στα Errors όπως είναι το **Root Mean Square Error** (**RMSE** - μέσο τετραγωνικό σφάλμα, με τύπο $RMSE = \sqrt{\frac{\sum_{i=1}^{N}(Predicted_i - Actual_i)^2}{n}}$) το οποίο όσο μεγαλύτερη είναι η τιμή του τόσο υπάρχει η πιθανότητα να είναι και η διαφορά στην προβλεπόμενη τιμή. Επίσης, στα Errors ανήκει και το **Absolute Error** (**AE** – απόλυτο σφάλμα), με την διαφορά τους να είναι στο αποτέλεσμα (το ένα είναι απόλυτο [AE] ενώ το άλλο τετραγωνικό [RMSE]) και στην χρήση τους (το AE είναι πιο αποδοτικό κριτήριο όταν υπάρχουν ελάχιστες ή καθόλου outliers τιμές, αντίθετα με το RMSE). Τέλος, το τελευταίο κριτήριο που γίνεται ανάλυση από το performance είναι το Prediction Average (Μέσος όρος πρόβλεψης).
- Στο prediction Chart του αυτόματου μοντέλου.

8.1 Δημιουργία ενός Decision Tree

Η δημιουργία του Δέντρου Απόφασης (5.3) στην πλατφόρμα RapidMiner ξεκινάει εισάγοντας τα δεδομένα (τα δεδομένα αυτά είναι αποτελούνται από την νέα επεξεργασμένη βάση) στην καρτέλα Design, το οποίο γίνεται με το ακόλουθο χειριστήριο (Εικόνα 8.1):

Εικόνα 8.1: Operator Εισαγωγής δεδομένων.

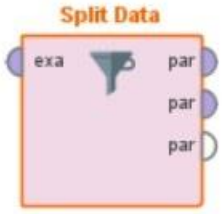
Αφού, γίνει η εισαγωγή της βάσης θα πρέπει στην συνέχεια να επιλεχτούν οι μεταβλητές που θα χρησιμοποιηθούν, όπως και η μεταβλητή που θα πάρει τον ρόλο του στόχου (Label) (Εικόνα 8.2).



Εικόνα 8.2: Operators επιλογής μεταβλητών.

Για το συγκεκριμένο μοντέλο επιλέχτηκαν όλες οι μεταβλητές (μέσα στο select Attributes).

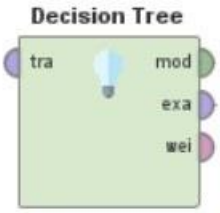
Στην συνέχεια, γίνεται η δημιουργία του βασικού μέρους του μοντέλου, η οποία ξεκινάει με τον διαχωρισμό των δεδομένων σε training set και testing set (Εικόνα 8.3).



Εικόνα 8.3: Operator διαχωρισμού δεδομένων.

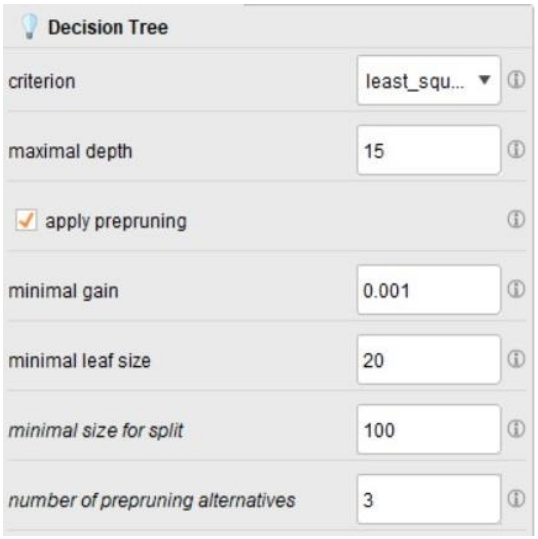
Ο διαχωρισμός πραγματοποιείται μέσα στην παράμετρο partitions. Το ποσοστό του training set είναι στο 70% και του testing set είναι στο 30%.

Έπειτα, ακολουθεί η δημιουργία του δέντρου μέσα από το χειριστήριο Decision Tree:



Εικόνα 8.4: Operator Decision Tree.

Σε αυτό το χειριστήριο (Εικόνα 8.4) επιλέγονται οι κυριότεροι υπερπαράμετροι που χρειάζεται το δέντρο απόφασης:



Εικόνα 8.5: Υπερπαράμετροι του Δέντρου απόφασης.

8.1.1 Επιλογή των υπερπαραμέτρων του μοντέλου

Η κάθε παράμετρος (Εικόνα 8.5) έχει το δικό της μοναδικό σκοπό και βοηθάει στην καλύτερη βελτιστοποίηση του δέντρου.

Criterion: Αυτή η παράμετρος χρησιμοποιείται για να επιλεχτεί το κριτήριο με το οποίο θα διαχωριστούν οι επιλεγμένες μεταβλητές. Η τιμή που επιλέχτηκε για αυτήν την παράμετρο είναι το **least_square,** καθώς με βάση την επιλεγμένη μεταβλητή (η μεταβλητή label) ελαχιστοποιεί την τετραγωνική απόσταση του μέσου όρου των τιμών στον κόμβο σε σχέση με την πραγματική τους τιμή.

Maximal depth: Η συγκεκριμένη παράμετρος αντιπροσωπεύει το μέγιστο βάθος κόμβων που θα δημιουργηθεί στο δέντρο. Η τιμή σε αυτή τη παράμετρο επιλέχτηκε αφού εκτελέστηκε για πρώτη φορά το μοντέλο με την τιμή -1 (με αυτήν την τιμή το χειριστήριο του δέντρου δημιουργεί όσους κόμβους στο βάθος χρειάζεται, έτσι ώστε να φτάσει από

μόνο του στο μέγιστο βάθος) διαπιστώθηκε από το διάγραμμα του δέντρου πως το μέγιστο βάθος που χρειάζεται είναι το 15.

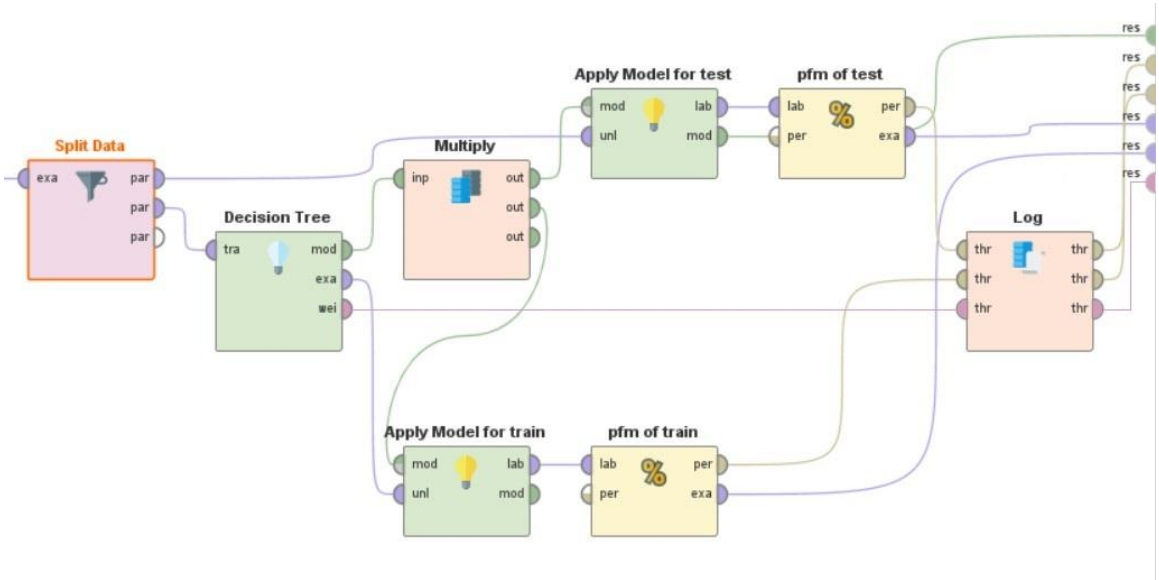
Minimal gain: Η παράμετρος αυτή θέτει το ελάχιστο κέρδος που θα έχει ο κάθε κόμβος. Εάν το ελάχιστο κέρδος έχει ως τιμή μια υψηλή τότε οι κόμβοι δεν διασπώνται αρκετά (το κέρδος του κόμβου υπολογίζεται αυτόματα πριν τον διαχωρισμό του), με αποτέλεσμα να δημιουργείτε ένα μικρότερο δέντρο (στην περίπτωση που η τιμή είναι υπερβολικά υψηλή τότε το δέντρο που θα δημιουργηθεί θα είναι με μοναδικό κόμβο).

Minimal leaf size: Αυτή η παράμετρος καθορίζει το ελάχιστο μέγεθος των τιμών των οποίων εμπεριέχονται μέσα στο κάθε φύλο. Όσο πιο μεγάλη είναι η τιμή (με βάση το data set που χρησιμοποιήθηκε) σε αυτήν την παράμετρο τόσο πιο πολύ αυξάνονται τα Errors (RMSE και AE).

Minimal size for split: Η τελευταία παράμετρος που χρησιμοποιήθηκε για τη δημιουργία του δέντρου είναι το ελάχιστο μέγεθος των τιμών μέσα στον κόμβο για τον διαχωρισμό του. Ο κόμβος διαχωρίζεται όταν το μέγεθος το σύνολο των τιμών είναι μεγαλύτερο ή ίσο με αυτό της τιμής της παραμέτρου.

8.1.2 Αποτελέσματα του Decision Tree

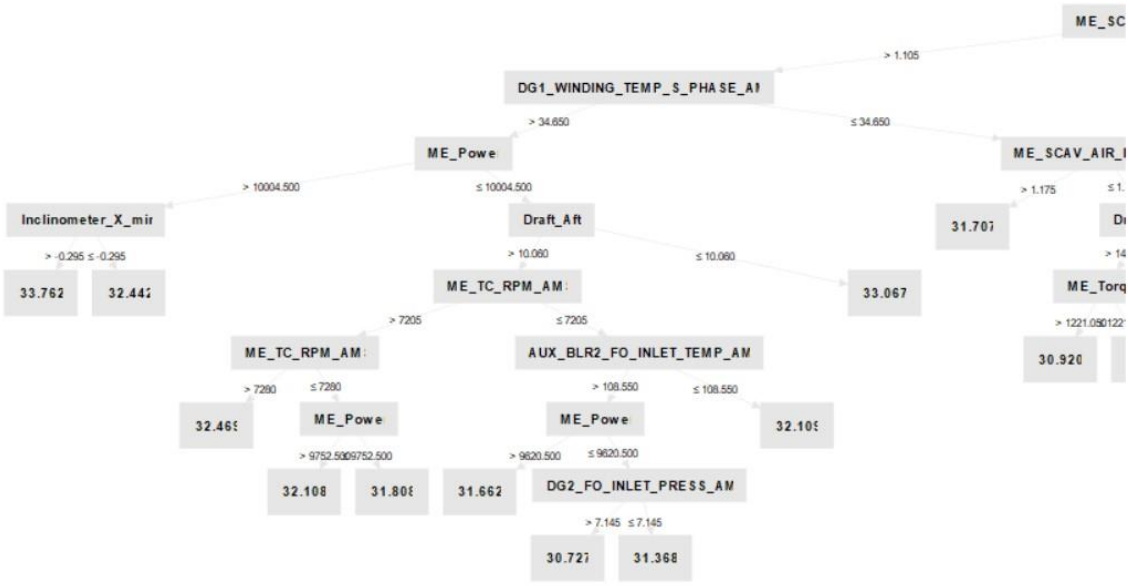
Η τελική μορφή του μοντέλου στην καρτέλα design είναι η εξής (Εικόνα 8.6):



Εικόνα 8.6: Τελικός σχεδιασμός του μοντέλου decision tree.

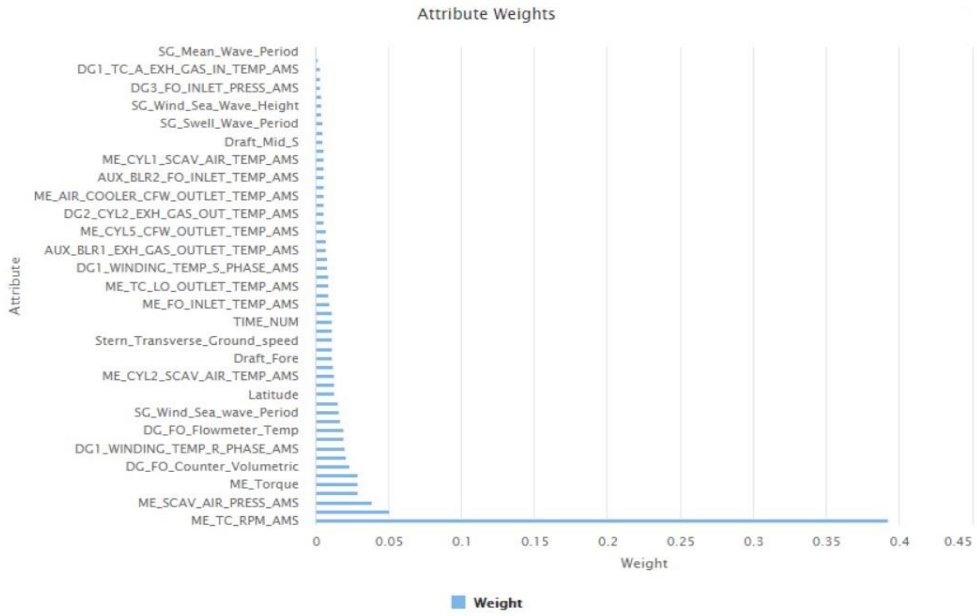
Μέσα στο performance (pfm) γίνεται η επιλογή όλων των κριτήριων απόδοσης του μοντέλου.

Αφού γίνει η εκτέλεση της διαδικασίας το δέντρο που δημιουργείται παίρνει την ακόλουθη μορφή (Εικόνα 8.7):



Εικόνα 8.7: Ένα μέρος από το διάγραμμα του δέντρου απόφασης.

Στην συνέχεια, δημιουργείται ένα γράφημα ανάμεσα στις μεταβλητές και τα βάρη που χρησιμοποιήθηκαν (Γραφική 8.1).



Γραφική 8.1: Τα βάρη του μοντέλου Decision Tree.

Τα βάρη έχουν έναν σημαντικό ρόλο στην απόδοση του μοντέλου, καθώς επηρεάζουν αρκετά το prediction value.

Έπειτα, από την καρτέλα του Performance (Εικόνα 8.8) δημιουργείται η απόδοση του decision tree για το training και το testing set.

PerformanceVector Trainig set

```
PerformanceVector:
root_mean_squared_error: 0.741 +/- 0.000
absolute_error: 0.412 +/- 0.616
squared_correlation: 0.996
prediction_average: 17.379 +/- 12.064
```

PerformanceVector Testing set

```
PerformanceVector:
root_mean_squared_error: 0.812 +/- 0.000
absolute_error: 0.437 +/- 0.684
squared_correlation: 0.995
prediction_average: 17.542 +/- 12.006
```

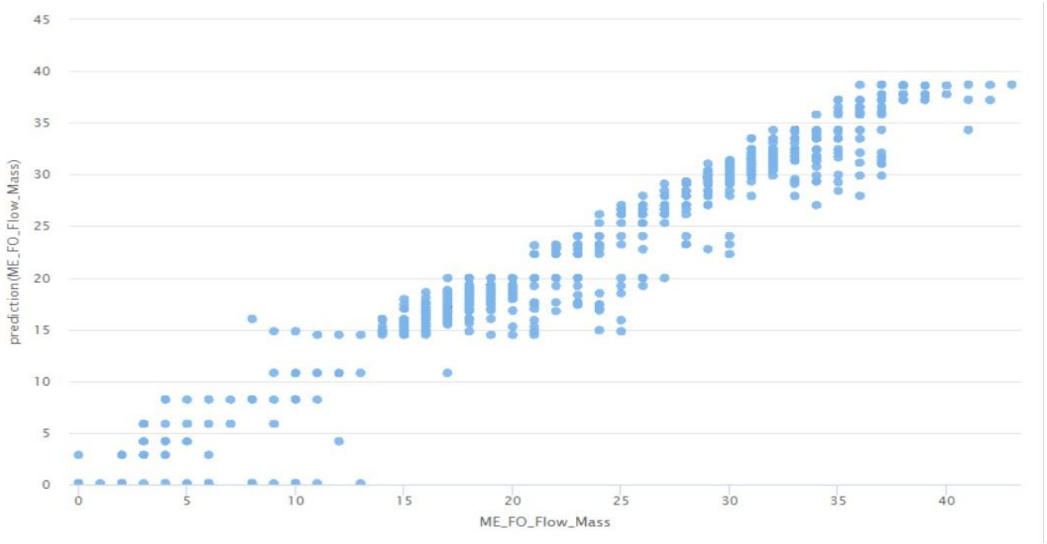
Εικόνα 8.8: Performance Training set (αριστερά) και Testing set (Δεξιά).

Ο πρώτος έλεγχος που γίνεται στο Performance είναι στο Squared Correlation, καθώς είναι το κριτήριο που αποδεικνύει εάν το μοντέλο έχει υποστεί overfitting κατά τον διαχωρισμό και την εκπαίδευση των δεδομένων. Στην Εικόνα 8.8 το κριτήριο αυτό είναι σχεδόν ίδιο και στις δυο περιπτώσεις, άρα βγαίνει το συμπέρασμα πως το μοντέλο δεν έχει υποστεί overfitting (στην περίπτωση που το Squared Correlation στο testing set ήταν μεγαλύτερο τότε το μοντέλο θα είχε υποστεί underfitting).

Στην συνέχεια, για να βρεθεί η πιο μειωμένη τιμή για το RMSE στο testing set και ταυτόχρονα να μην έχει μεγάλη απόσταση από την αντίστοιχη τιμή στο training set, πραγματοποιήθηκαν πολλές εναλλαγές στην παράμετρο του Minimal Leaf Size (8.1.1).

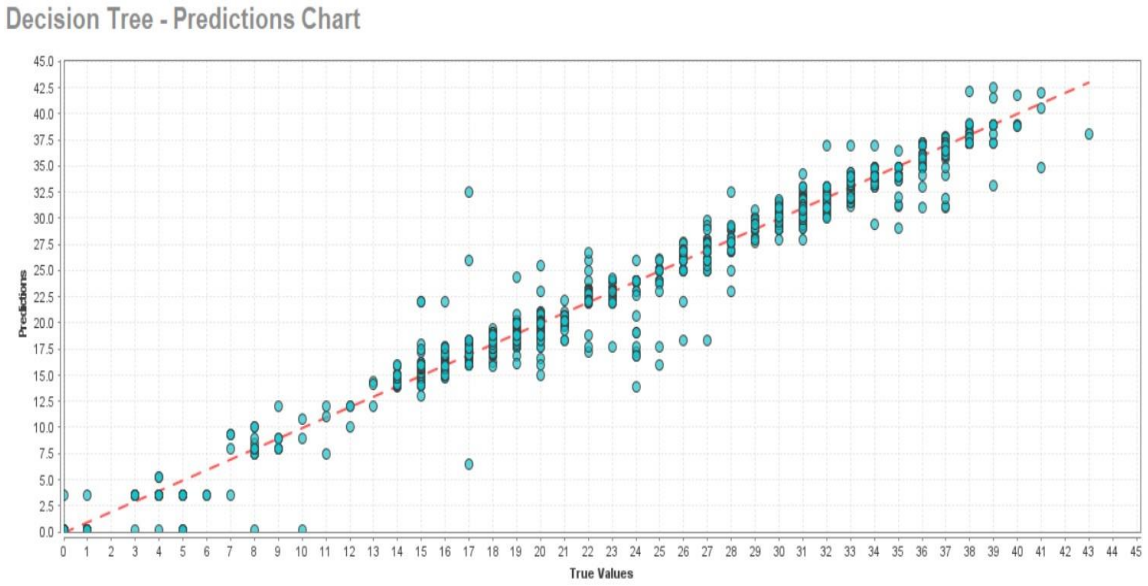
Η τιμή του Prediction Average αναφέρεται στην μέση τιμή από την μεταβλητή της προβλεπόμενης τιμής (Prediction Value).

Από το example (exa) του testing set βγαίνει το αποτέλεσμα της προβλεπόμενης τιμής και για περισσότερη κατανόηση της δημιουργήθηκε το ακόλουθο γράφημα (Γραφική 8.2).



Γραφική 8.2: Γραφική της προβλεπόμενης τιμής (Prediction) με την κανονική τιμή (ME FO Flow Mass).

Έπειτα, με την βοήθεια του Auto Model (6.1.1) αφού δημιουργηθεί το δέντρο απόφασης με τους ίδιους παραμέτρους, παράγει ως έξοδο την εξής γραφική (Γραφική 8.3) με σκοπό την βεβαίωση πως το μοντέλο (στο Design) έχει εκπαιδευτεί και δημιουργηθεί σωστά:



Γραφική 8.3: Γραφική της προβλεπόμενης τιμής (Predictions) με της τιμής της μεταβλητής (True Values).

Γίνεται η παρατήρηση πως οι τιμές που έχουν πολύ μεγάλη απόσταση (όπως είναι οι τιμές στο 17 στον άξονα X - True Values) από την κόκκινη ευθεία (Regression) μπορούν να θεωρηθούν ώς outliers.

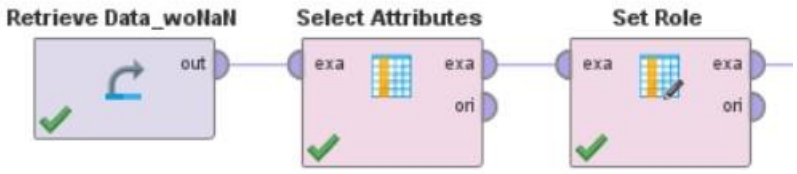
Συγκρίνοντας της δυο γραφικές (8.2 και 8.3) σημειώνεται πως υπάρχει μικρή διαφορά (θεωρείται αναμενόμενο καθώς στο Auto Model δεν επεξεργάστηκαν όλες οι παράμετροι όπως το μοντέλο στο Design).

Τέλος, ο **χρόνος εκτέλεσης** (Execution time) του μοντέλου κατατάσσεται στο 1 second.

8.2 Γενικευμένο Γραμμικό Μοντέλο - Generalized Linear Model (GLM)

Το γενικευμένο γραμμικό μοντέλο (GLM) επιλέχθηκε έναντι του απλού γραμμικού μοντέλου (Linear Model), καθώς με βάση την στατιστική το GLM είναι μια ευέλικτη προσέγγιση του γραμμικού μοντέλου παλινδρόμησης. Το GLM στην ουσία γενικεύει τη γραμμική παλινδρόμηση (linear regression) επιτρέποντας στο γραμμικό μοντέλο να αποκτήσει μια συσχέτιση (link) με την μεταβλητή απόκρισης (response variable – label), με αποτέλεσμα να επιτρέπεται στο μέγεθος της διακύμανσης κάθε μέτρησης (του data set) να είναι συνάρτηση για την προβλεπόμενη τιμή.

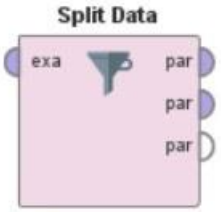
Η δημιουργία του γενικευμένου γραμμικού μοντέλου ξεκινάει όπως και στο δέντρο απόφασης (8.1) με την εισαγωγή των δεδομένων και την επιλογή των μεταβλητών (Εικόνα 8.9).



Εικόνα 8.9: Εισαγωγή των δεδομένων και επιλογή μεταβλητών.

Για το μοντέλο αυτό στο χειριστήριο Select Attributes επιλέχτηκαν όλες οι διαθέσιμες μεταβλητές της βάσης δεδομένων.

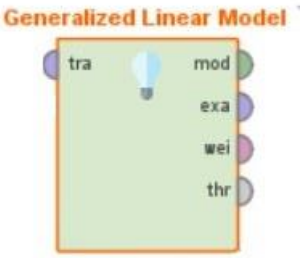
Αφού γίνει η επιλογή των μεταβλητών (εισαγωγής και στόχου) τότε ακολουθεί ο διαχωρισμός των δεδομένων από το χειριστήριο Split Data (Εικόνα 8.10).



Εικόνα 8.10: Διαχωρισμός των δεδομένων.

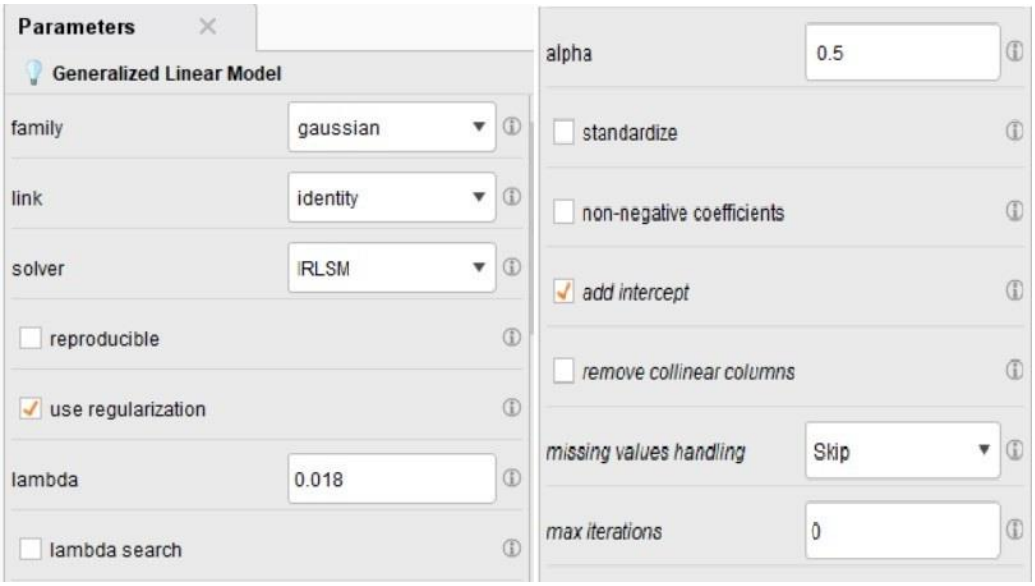
Τα ποσοστά του διαχωρισμού στην παράμετρο partitions είναι το 70% για το training set και το 30% για το testing set.

Στην συνέχεια, εισάγετε στην σχεδίαση το κυριότερο χειριστήριο. Το χειριστήριο Generalized Linear Model (Εικόνα 8.11) με το οποίο πραγματοποιείται η επεξεργασία των παραμέτρων του.



Εικόνα 8.11: Χειριστήριο του GLM.

Μέσα στο χειριστήριο αυτό επιλέγονται όλες οι υπερπαράμετροι (Εικόνα 8.12) που χρειάστηκαν να επεξεργαστούν για την καλύτερη βελτιστοποίηση του μοντέλου.



Εικόνα 8.12: Οι υπερπαράμετροι του μοντέλου (δεξιά η αρχή και αριστερά η συνέχεια).

8.2.1 Επιλογή των υπερπαραμέτρων του μοντέλου

Family: Η παράμετρος αυτή επιλέγει τον τύπου με τον οποίο θα λειτουργήσει το μοντέλο είτε αυτός είναι κατηγοριοποίηση (Classification), είτε αυτός είναι παλινδρόμηση (Regression). Η τιμή **Gaussian** που έχει επιλεχτεί δηλώνει πως το μοντέλο θα χρησιμοποιηθεί για παλινδρόμηση και τα δεδομένα θα είναι αριθμητικά.

Link: Αυτή η παράμετρος είναι μια συνάρτηση η οποία συνδέει το linear predictor με τη συνάρτηση κατανομής. Η τιμή **identity** επιλέχτηκε ως η πιο βέλτιστη (για το Performance) και γρήγορη από τις άλλες δυο πιθανές τιμές (το log και το inverse).

Solver: Η χρήση της παραμέτρου αυτής είναι να επιλέγει την μέθοδο της λύσης που θα χρησιμοποιήσει. Οι πιθανές τιμές που μπορεί να δεχτεί αυτή η παράμετρος είναι δυο το IRLSM και το L_BFGS. Παρόλου που η λύση της L_BFGS είναι καλύτερη για βάσεις δεδομένων με μεγάλο αριθμό μεταβλητών, χρησιμοποιήθηκε το **IRLSM** καθώς μείωνε αρκετά τον χρόνο εκτέλεσης του μοντέλου.

Regularization: Το regularization επιλέχτηκε καθώς δίνει την δυνατότητα στον διαχειριστή του μοντέλου να μπορεί να εισάγει τιμές στις παραμέτρους lambda και alpha με σκοπό να κανονικοποιηθεί το αποτέλεσμα.

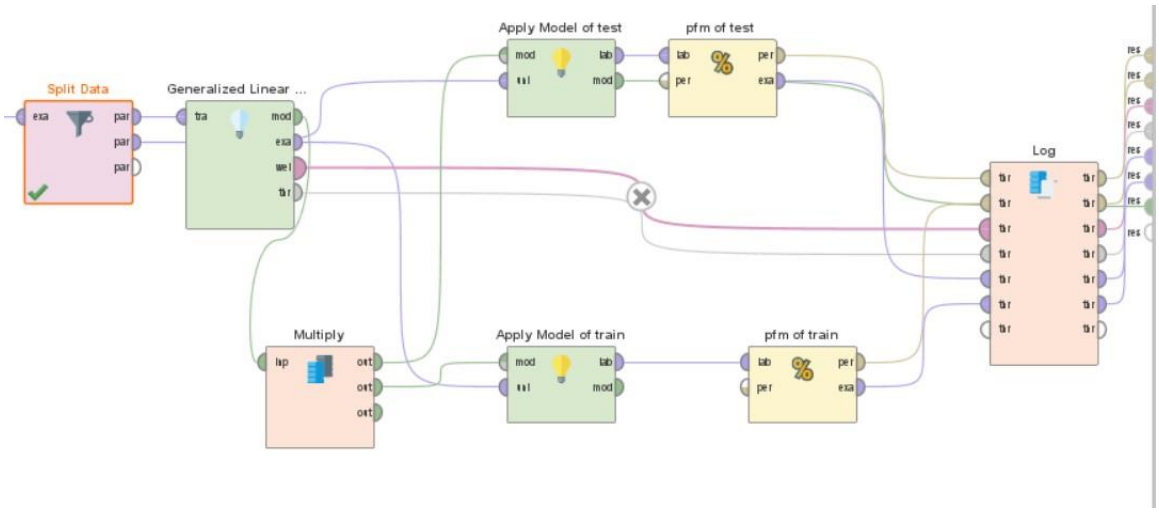
Lambda: Μια από τις πιο σημαντικές παραμέτρους καθώς ελέγχει το «ποσοστό» της κανονικοποίησης που θα εφαρμοστεί. Η τιμή 0.018 επιλέχτηκε καθώς ήταν η καλύτερη τιμή (θα αναπτυχθεί περαιτέρω στα αποτελέσματα) από ένα εύρος τιμών 0.0 έως 1.79769.

Alpha: Η παράμετρος alpha επιλέγει την κατανομή μεταξύ των κυρώσεων L1 (Lasso – με τιμή 1.0) και L2 (Ridge regression – με τιμή 0.0). Η τιμή του 0.5 επιλέχτηκε καθώς το IRLSM του Solver έχει μόνο αυτήν την τιμή ως αποδεκτή.

Missing value handling: Τελευταία παράμετρος που θα αναλυθεί είναι η αντιμετώπιση των missing values. Το χειριστήριο δίνει την δυνατότητα να γίνει η αντικατάσταση τους με την μέση τιμή. Επειδή, όμως η αντιμετώπιση τους έχει γίνει στο κομμάτι του preprocessing (Κεφάλαιο 7), οπότε δεν χρειάζεται να επιβαρυνθεί παραπάνω το μοντέλο. Έτσι, παίρνει την τιμή skip.

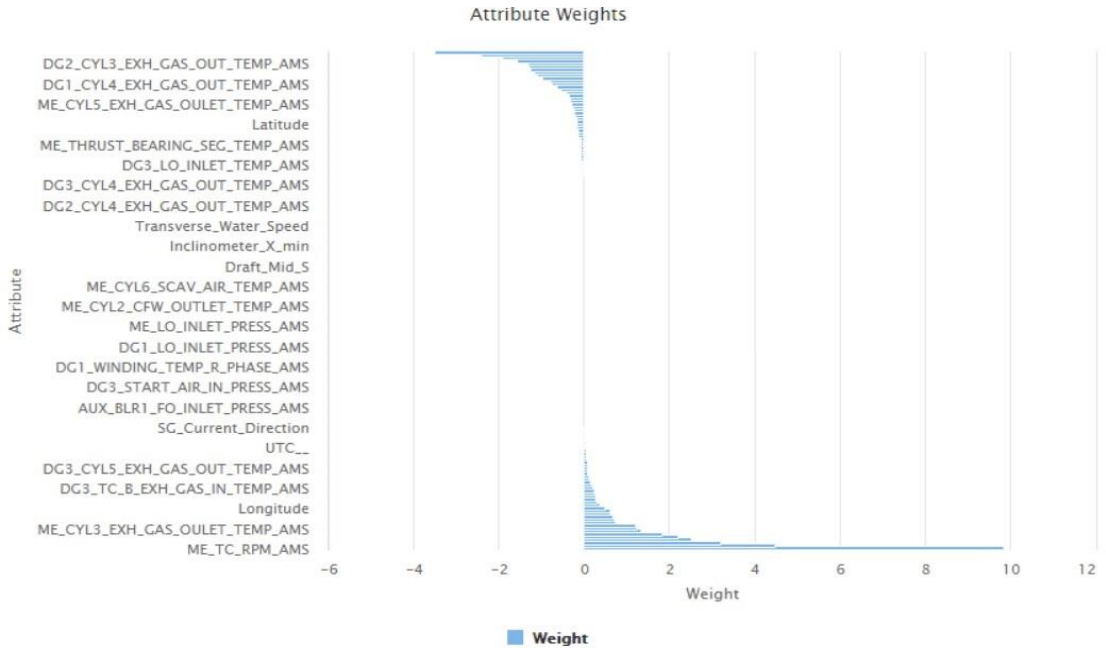
8.2.2 Αποτελέσματα του GLM

Το μοντέλο διαμορφώνεται στην καρτέλα του design με την ακόλουθη μορφή (Εικόνα 8.13):



Εικόνα 8.13: Τελική μορφή του μοντέλου GLM.

Μετά την εκτέλεση της διαδικασίας το πρώτο που παρατηρείται είναι ο πίνακας με τα βάρη για την καλύτερη απεικόνιση του θα δημιουργηθεί ένα γράφημα με τις μεταβλητές και τα βάρη (Γραφική 8.4):



Γραφική 8.4: Τα βάρη του μοντέλου.

Γίνεται η παρατήρηση πως, όσο περισσότερο χρησιμοποιείτε μια μεταβλητή τόσο πιο μεγάλη είναι η τιμή της στο weight (ακόμα και αυτές που φαίνονται να ανήκουν στο μηδέν χρησιμοποιούνται).

Στην συνέχεια, γίνεται η ανάλυση του πιο σημαντικού μέρους, της καρτέλας Performance για το testing set και το training set (Εικόνα 8.14):

PerformanceVector

Training set

```
PerformanceVector:
root_mean_squared_error: 0.785 +/- 0.000
absolute_error: 0.474 +/- 0.626
squared_correlation: 0.996
prediction_average: 17.444 +/- 12.062
```

PerformanceVector

Testing set

```
PerformanceVector:
root_mean_squared_error: 0.792 +/- 0.000
absolute_error: 0.479 +/- 0.631
squared_correlation: 0.996
prediction_average: 17.421 +/- 12.041
```

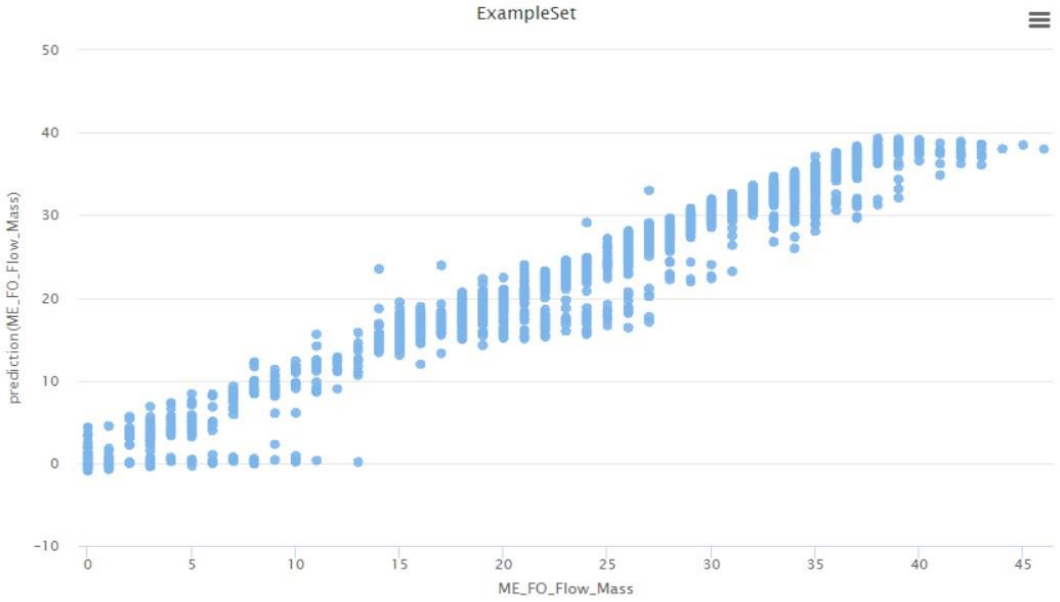
Εικόνα 8.14: Performance για το training set (αριστερά) και για το testing set (δεξιά).

Στο Performance παρατηρείται πως το Squared correlation είναι ίδιο όσο στο training set τόσο και στο testing set, με αποτέλεσμα η πρώτη όψη να είναι πως έχει δημιουργηθεί το «τέλειο» μοντέλο.

Για να βρεθεί το καλύτερο RMSE πραγματοποιήθηκαν αρκετές εναλλαγές στην υπερπαράμετρο του Solver και του Lambda. Έπειτα, παρατηρήθηκε πως το IRLSM ως

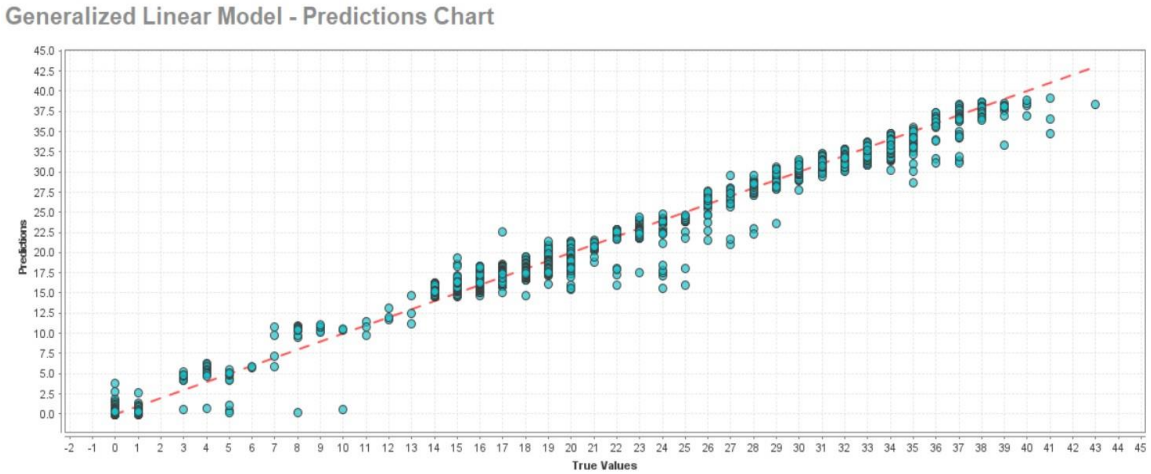
τιμή του Solver, συνδυαστικά με το χαμηλό Lambda έδιναν το καλύτερο αποτέλεσμα στο RMSE (το Lambda όταν πήρε την τιμή του 0.017 τότε το μοντέλο είχε πρόβλημα Underfitting, δηλαδή το Squared correlation γινόταν μηδέν).

Στην συνέχεια από την έξοδο exa (example) του χειριστήριού του Performance (του testing set) παράγεται το prediction attribute, για την καλύτερη σύγκριση του δημιουργείται το επόμενο γράφημα (Γραφική 8.5).



Γραφική 8.5: Γραφική του prediction attribute και της μεταβλητής της βάσης δεδομένων.

Ωστόσο, για καλύτερη επιβεβαίωση των αποτελεσμάτων γίνεται σύγκριση με των αποτελεσμάτων του Auto Model για το GLM.



Γραφική 8.6: Γραφική μεταξύ του Prediction και του True value από Auto Model.

Παρατηρείται πως αν συγκριθούν οι δυο γραφικές (8.5 και 8.6) του μοντέλου η γραφική 8.5 είναι πιο σωστά καταμερισμένη από την γραφική 8.6. Ένα ακόμα κριτήριο είναι και η σύγκριση ανάμεσα στο Performance (Πίνακας 8.1) του αποτελέσματος του Auto Model με το αντίστοιχο του μοντέλου που σχεδιάστηκε.

Root Mean Squared Error	0.920 +/- 0.075
Absolute Error	0.587 +/- 0.022
Square Correlation	0.996

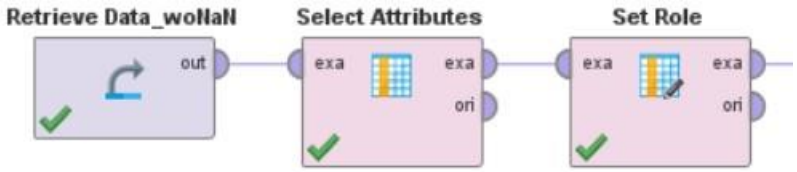
Πίνακας 8.1: Performance Auto Model.

Όπως φαίνεται και από το Performance η διαφορά είναι αρκετά μεγάλη στο RMSE, με αποτέλεσμα να επιβεβαιώνεται πως το μοντέλο στο Design έχει δημιουργηθεί και εκπαιδευτεί στην πιο βέλτιστη του μορφή.

Τέλος, ο **χρόνος εκτέλεσης** (Execution time) του μοντέλου κατατάσσεται στα 4 second.

8.3 Δημιουργία του Τυχαίου Δάσους – Random Forest

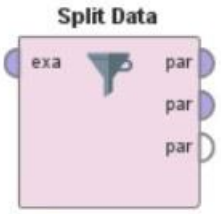
Η δημιουργία του Random Forest ξεκινάει όπως και σε όλα τα προηγούμενα μοντέλα με την εισαγωγή των δεδομένων και την επιλογή των μεταβλητών (Εικόνα 8.15).



Εικόνα 8.15: Εισαγωγή των δεδομένων και επιλογή μεταβλητών εισαγωγής και ρόλου.

Στο χειριστήριο select Attributes επιλέχτηκαν όλες οι μεταβλητές για εισαγωγή στην δημιουργία του μοντέλου.

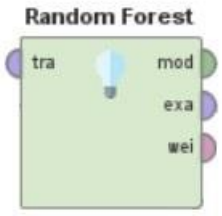
Έπειτα, γίνεται ο διαχωρισμός των δεδομένων για το training set και το testing set από το χειριστήριο Split Data (Εικόνα 8.16):



Εικόνα 8.16: Χειριστήριο διαχωρισμού δεδομένων.

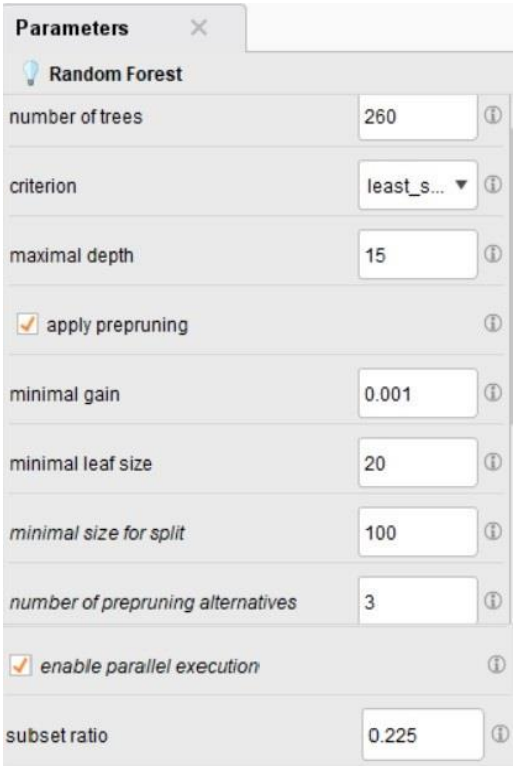
Τα ποσοστά του διαχωρισμού στην παράμετρο partitions είναι το 70% για το training set και το 30% για το testing set.

Στην συνέχεια το Split Data συνδέεται με το σημαντικότερο χειριστήριο το οποίο δημιουργεί το μοντέλο. Το χειριστήριο αυτό είναι το Random Forest (Εικόνα 8.17).



Εικόνα 8.17: Χειριστήριο Random Forest

Στο χειριστήριο αυτό γίνονται όλες οι επιλογές για τους υπερπαραμέτρους βελτιστοποίησης του μοντέλου (Εικόνα 8.18).



Εικόνα 8.18: Επιλογή υπερπαραμέτρων Random Forest.

8.3.1 Επιλογή των υπερπαραμέτρων του μοντέλου

Number of trees: Η παράμετρος αυτή ξεκαθαρίζει πόσα τυχαία δέντρα (Random Trees) θα δημιουργηθούν.

Οι υπερπαράμετροι του Criterion, maximal depth, minimal gain, minimal leaf size, minimal size for split και number of prepruning alternatives επιλέχτηκαν όπως ήταν επιλεγμένες κατά τη δημιουργία του Decision Tree (8.1.1), καθώς έχουν θεωρηθεί ως οι

πιο βέλτιστες για το δέντρο απόφασης (σε αυτήν την περίπτωση οι υπερπαράμετροι αυτοί αναφέρονται στην δημιουργία του κάθε δέντρου μέσα στο τυχαίο δάσος).

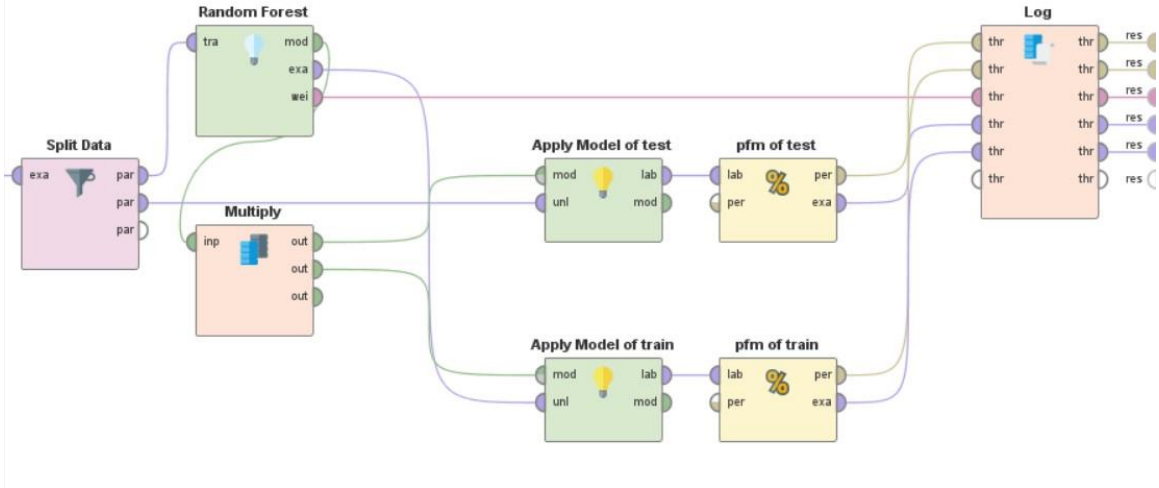
Enable parallel execution: Η χρήση αυτής της παραμέτρου δίνει τη δυνατότητα στο τυχαίο δάσος να μπορεί να εκτελέσει την αναπαραγωγή των δέντρων πιο γρήγορα, καθώς τα αναπαράγει παράλληλα. Όμως, αυτό έχει ως αποτέλεσμα την κατανάλωση περισσότερης επεξεργαστικής ισχύς και μνήμης.

Subset ratio: Η τελευταία και από τις πιο σημαντικές υπερπαραμέτρους που χρησιμοποιήθηκαν, είναι το subset ratio. Αυτή η υπερπαράμετρος χρησιμοποιείται για να επιλέγει την αναλογία (ratio) των **τυχαίων** μεταβλητών που θα επιλεχτούν για το test. Η τιμή του καθορίζεται από τον ακόλουθο τύπο: $int(\log(m) + 1)$ όπου το m είναι οι μεταβλητές.

Οι τιμές για τις δυο υπερπαραμέτρους που επεξεργάστηκαν αναλύονται στα αποτελέσματα (8.3.2).

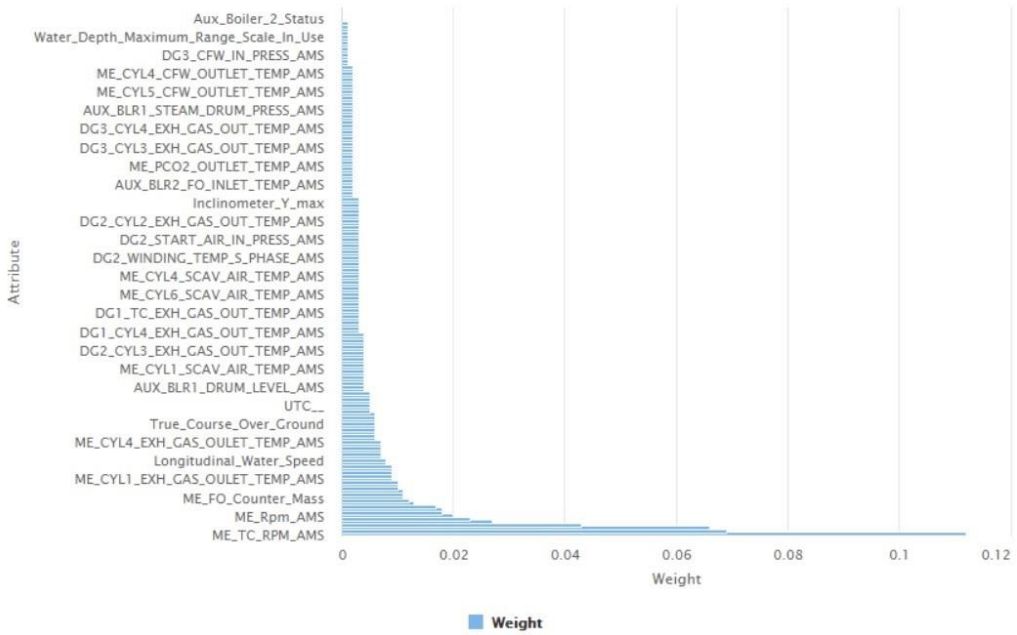
8.3.2 Αποτελέσματα του Random Forest

Η σχεδίαση του μοντέλου Random Forest στην καρτέλα Design παίρνει την ακόλουθη μορφή (Εικόνα 8.19):



Εικόνα 8.19: Τελική μορφή του μοντέλου Random Forest.

Αφού γίνει η εκτέλεση της διαδικασίας με τους επιλεγμένους υπερπαραμέτρους το πρώτο που παρατηρείται είναι το γράφημα του weight (Γραφική 8.7):



Γραφική 8.7: Γραφική ανάμεσα στα βάρη και στις μεταβλητές.

Τα βάρη μπορούν να αλλάξουν ανάλογα με την τιμή που εισάγεται στην υπερπαράμετρο subset ratio. Μεγαλώνοντας την τιμή αυτή τα βάρη αυξάνονται για όλες τις μεταβλητές έτσι αυξάνεται και το Squared Correlation, όμως το RMSE αυξάνεται και αυτό κάτι που δεν ήταν επιθυμητό. Οπότε, η επιλογή της τιμής 0.225 έγινε ως η πιο βέλτιστη (μειώνοντας περισσότερο την τιμή διακινδυνεύεται αρκετά το μοντέλο να έχει πρόβλημα underfitting).

Ακολουθεί η ανάλυση της καρτέλας του Performance για το training set και το testing set (Εικόνα 8.20):

PerformanceVector

Training set

```
PerformanceVector:
root_mean_squared_error: 0.702 +/- 0.000
absolute_error: 0.387 +/- 0.586
squared_correlation: 0.997
prediction_average: 17.470 +/- 12.023
```

PerformanceVector

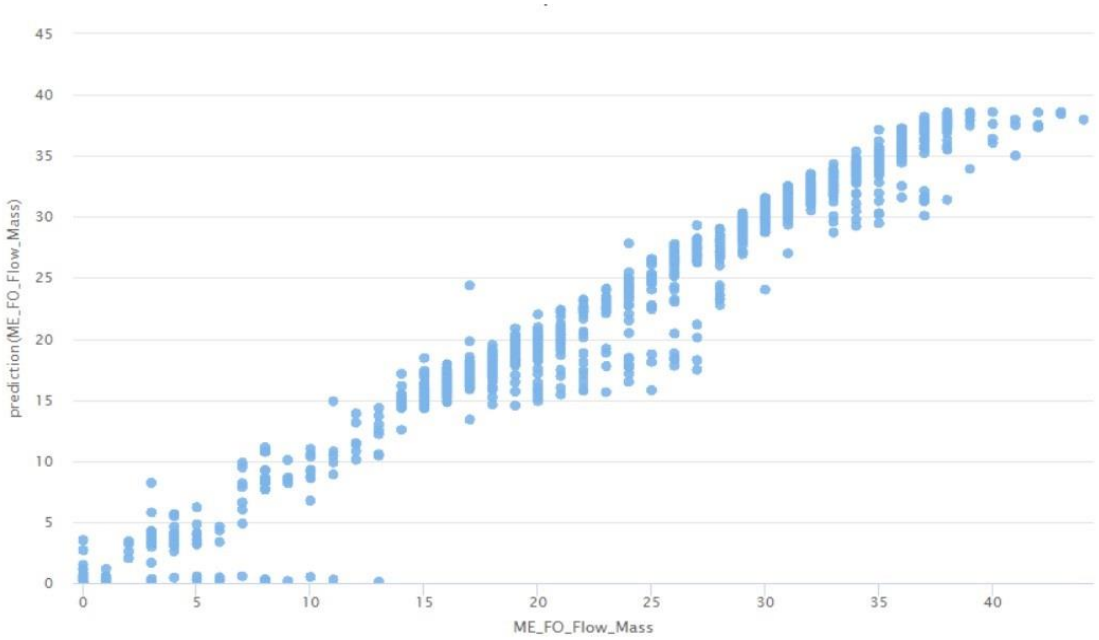
Testing set

```
PerformanceVector:
root_mean_squared_error: 0.740 +/- 0.000
absolute_error: 0.411 +/- 0.616
squared_correlation: 0.996
prediction_average: 17.329 +/- 12.103
```

Εικόνα 8.20: Performance του Random Forest για το training set (αριστερά) και του testing set (δεξιά).

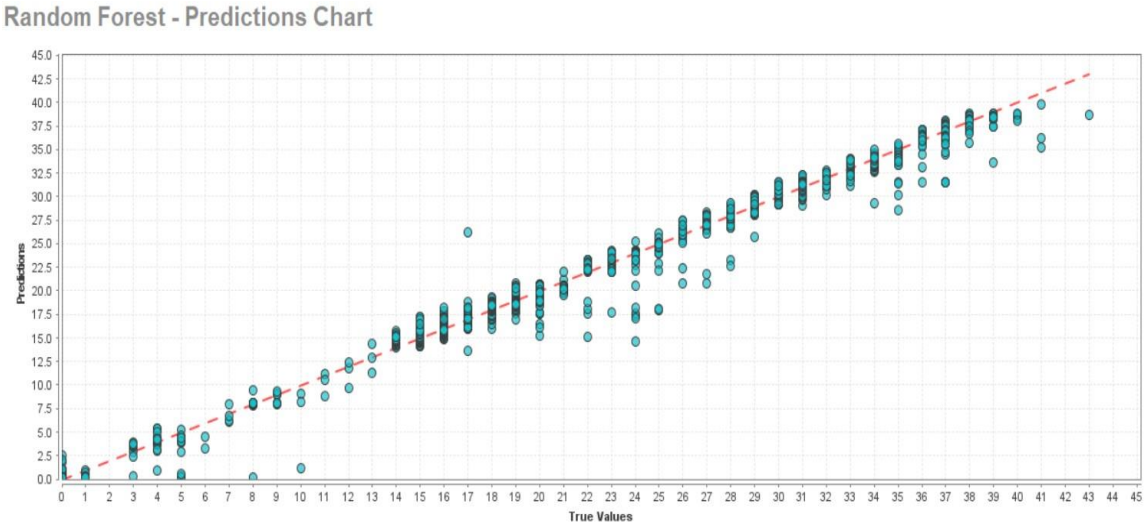
Όπως παρατηρείται και στο Squared Correlation δεν είναι απόλυτα ίσο μεταξύ του training set και του testing set (αναφέρθηκε και παραπάνω), την λύση σε αυτό την έδωσε η υπερπαράμετρος number of trees. Αυξάνοντας τον αριθμό των δέντρων που θα δημιουργηθούν (από 100 που ήταν το default σε 200) παρατηρήθηκε πως το S.C. ανέβηκε και αυτό, επομένως βρέθηκε το βέλτιστο που ήταν ανάμεσα στα 250 με 270 δέντρα.

Έπειτα, από την έξοδο exa (example) του χειριστήριού του Performance (του testing set) παράγεται το prediction attribute, για την καλύτερη σύγκριση του παράγεται το ακόλουθο γράφημα (Γραφική 8.8).



Γραφική 8.8: Γραφική του prediction attribute μεταξύ του True Value.

Επομένως, για υπάρχει καλύτερη επιβεβαίωση πως το μοντέλο έχει δημιουργηθεί και εκπαιδευτεί σωστά θα γίνει σύγκριση με τα αποτελέσματα από το Auto Model (με παραμέτρους το 260 στην αναπαραγωγή των δέντρων και 15 στο βάθος του κάθε δέντρου).



Γραφική 8.9: Γραφική του Prediction με το True Values.

Παρατηρείται πως οι δυο γραφικές (8.8 και 8.9) είναι σχεδόν όμοιες μεταξύ τους. Οι διαφορές που φαίνονται είναι στο πλήθος των τιμών, καθώς στο Auto Model έγινε χρήση ενός εύρος των πέντε χιλιάδων γραμμών, ενώ στο testing set του design το εύρος των γραμμών που χρησιμοποιήθηκαν είναι περίπου στις έντεκα χιλιάδες.

Το τελευταίο που συγκρίνεται ανάμεσα στο μοντέλο του Design και στο μοντέλο από το Auto Model είναι από την καρτέλα του Performance (Πίνακας 8.2).

Root Mean Square Error	0.741 +/- 0.085
Absolute Error	0.413 +/- 0.620
Square Correlation	0.998

Πίνακας 8.2: Απόδοση του μοντέλου από το Auto Model.

Γίνεται η παρατήρηση πως παρόλο που το μοντέλο από το Auto Model έχει εκπαιδευτεί καλύτερα (Square Correlation 0.998 έναντι του 0.996) το αποτέλεσμα του στο κριτήριο του RMSE είναι περίπου το ίδιο. Συνεπάγεται, πως το μοντέλο έχει εκπαιδευτεί και δημιουργηθεί σωστά.

Τέλος, ο **χρόνος εκτέλεσης** (Execution Time) του μοντέλου κατατάσσεται στα 34 second.

Γ) ΜΕΡΟΣ ΤΡΙΤΟ

i) Συμπεράσματα

Αφού έχει γίνει η δημιουργία, η εκπαίδευση και η ανάλυση των μοντέλων στο προηγούμενο μέρος της εργασίας, σε αυτό το μέρος θα γίνει η σύγκριση μεταξύ τους ώστε να βρεθεί το καλύτερο για το κάθε ζητούμενο.

Αρχικά δημιουργείται ένας πίνακας με το Performance (με μόνο το κριτήριο του RMSE) και τον χρόνο εκτέλεσης από κάθε μοντέλο που δημιουργήθηκε.

Όνομα Μοντέλου	**Rout Mean Square Error**	**Absolute Error**	**Χρόνος εκτέλεσης**
Decision Tree	0.812	0.437	1 second
Generalized Linear Model	0.792	0.479	4 second
Random Forest	0.740	0.411	34 second

Πίνακας σύγκρισης των μοντέλων.

Όπως φαίνεται και στον παραπάνω πίνακα, κάθε μοντέλο είναι πιο εξιδεικευμένο ανάλογα με τις απαιτήσεις που ζητούνται. Για καλύτερη κατανόηση των απαιτήσεων θα γίνει ανάπτυξη σε τρία παραδείγματα.

- Εάν το ζητούμενο είναι η ακρίβεια τότε το καλύτερο μοντέλο είναι το Random Forest, καθώς έχει την χαμηλότερη τιμή στο Rout Mean Square Error.
- Εάν το ζητούμενο είναι η ταχύτητα και δεν είναι τόσο σημαντική η ακρίβεια τότε το καλύτερο μοντέλο είναι το Decision Tree, καθώς μπορεί να έχει την υψηλότερη τιμή στο RMSE όμως είναι αρκετά κοντά με το GLM και είναι και το πιο γρήγορο.
- Τέλος, εάν το ζητούμενο είναι και η ταχύτητα και η ακρίβεια, τότε για τα δεδομένα που χρησιμοποιήθηκαν το καλύτερο μοντέλο είναι το GLM.

Συμπερασματικά το καλύτερο μοντέλο από τα τρία (Decision Tree, GLM, Random Forest) είναι το **Random Forest**. Παρόλο που έχει τον περισσότερο χρόνο εκτέλεσης έχει μεγάλη διαφορά στα κριτήρια απόδοσης (RMSE και AE) από τα υπόλοιπα μοντέλα.

ii) Δυσκολίες που αντιμετωπίστηκαν

Αρχικά, η πρώτη δυσκολία που αντιμετωπίστηκε ήταν στο κομμάτι του preprocessing (Κεφάλαιο 7), καθώς η βάση δεδομένων ήταν πολύ μεγάλη για να χρησιμοποιηθεί η λύση του PCA όπως αναφέρθηκε στο Κεφάλαιο 7 (7.3). Οπότε, έγινε η επιλογή να χρησιμοποιηθεί η μέθοδος της αγνόησης ολόκληρης της γραμμής (7.2.2.2) η οποία είχε ως αποτέλεσμα ένα πιο προσιτό data set, για να μειωθεί ο χρόνος εκτέλεσης και η κατανάλωση της επεξεργαστικής ισχύς των μοντέλων (αργότερα), όμως με γνώμονα πως μπορεί να υπάρχει κίνδυνος στην ακρίβεια των αποτελεσμάτων του κάθε μοντέλου.

Η επόμενη δυσκολία που αντιμετωπίστηκε ήταν στο μοντέλο του Random Forest (8.3). Στο μοντέλο αυτό παρατηρήθηκε πως όταν υπήρχε μεγάλος αριθμός στην υπερπαράμετρο number of trees (8.3.1), τότε η διαδικασία της εκτέλεσης του μοντέλου (όπως ήταν αναμενόμενο, καθώς έπρεπε να δημιουργήσει πάρα πολλά δέντρα απόφασης) χρειαζόταν περισσότερο χρόνο και καταλάμβανε υπερβολικό ποσοστό από την επεξεργαστική ισχύ, με αποτέλεσμα να υπήρχε κίνδυνος να μην μπορεί να ολοκληρωθεί η διαδικασία. Αυτός ο κίνδυνος μειώθηκε (μέχρι την εξάλειψη του) όταν βρέθηκαν οι κατάλληλες τιμές για τους υπερπαραμέτρους του μοντέλου.

Τέλος, να σημειωθεί πως για λόγο τις έλλειψης πόρων (επεξεργαστικής ισχύς και μνήμης RAM) όλα τα μοντέλα δεν μπορούσαν να είχαν την «τέλεια» ακρίβεια.

Βιβλιογραφία

[1] Mehmed Kantardzic. “DATA MINING Concepts, Models, Methods, and Algorithms” (Third Edition). IEEE PRESS WILEY, 2020.

[2] Jiawei Han, Micheline Kamber, Jian Pei. “DATA MINING Concepts and Techniques” (Third Edition). Morgan Kaufmann, USA, 2012.

[3] Oracle. Database defined. [Online].

Available: https://www.oracle.com/database/what-is-database/

[4] Κωνσταντίνος Διαμαντάρας, Δημήτρης Μπότσης. ”Μηχανική Μάθηση”. Κλειδάριθμος, Αθήνα, Ιούλιος 2019.

[5] Aurélien Géron. “Hands – On Machine Learning with Scikit – Learn, Keras, and TensFlow” (Second Edition). O’REILLY, September 2019.

[6] Salvador García, Julián Luengo, Francisco Herrera. “Data Preprocessing in Data Mining”. Spinger, Switzerland, 2015.

[7] Yulia Garvilova, Olga Bolgurtseva. “What is Data Preprocessing in ML?”. [Online]. Available: https://serokell.io/blog/data-preprocessing

[8] Deepak Jain. “Data Preprocessing in data mining”. [Online], September 2019. Available: https://www.geeksforgeeks.org/data-preprocessing-in-data-mining/

[9] Shai Shalev – Shwartz, Shai Ben – David. “UNDERSTANDING MACHINE LEARNING From Theory to Algorithms”. Cambridge University Press, 2014.

[10] Wittern H. Ian, Frank Eibe, Hall A. Mark, Pal J. Christopher. “Data Mining Practical Machine Learning Tools and Techniques” (Fourth Edition). Morgan Kaufmann, Elsevier, USA, 2017.

[11] Xin – She Yang. “Introduction to Algorithms for Data Mining and Machine Learning”. Elsevier, United Kingdom, 2019.

[12] Russell J. Stuart, Norvig Peter. “Artificial Intelligence A modern Approach” (Third Edition). Pearson, New Jersey, 2010.

[13] RapidMiner Inc. [Online]. Available: https://rapidminer.com

[14] Project Jupyter Community. [Online]. Available: https://jupyter.org/community

[15] IBM. “Generalized Linear Models”. [Online]. Available:

https://www.ibm.com/docs/de/spss-statistics/24.0.0?topic=option-generalized-linear-models

[16] Markus Hofmann, Ralf Klinkenberg. “RapidMiner: Data Mining Use Cases and Business Analytics Applications” (First Edition). Chapman, Hall/CRC, March 2014.

Ιστοσελίδες Προγραμμάτων

RapidMiner. [Online]. Available: https://rapidminer.com

Anaconda (Jyputer Notebook). [Online]. Available: https://www.anaconda.com