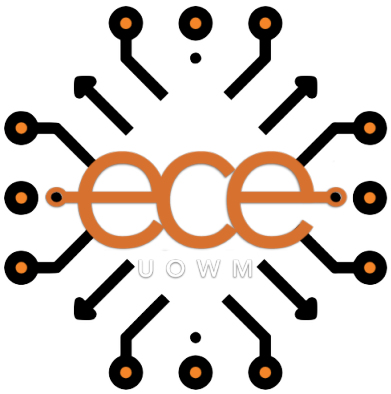


Πανεπιστήμιο Δυτικής Μακεδονίας

Τμήμα Ηλεκτρολόγων Μηχανικών & Μηχανικών Πληροφορικής

Τεχνικές προεπεξεργασίας δεδομένων στη μηχανική μάθηση

Μάριο Καντρίου

Επιβλέπων Καθηγητής: Νικόλαος Πλόσκας

15 Φεβρουαρίου 2021

Περίληψη

Η προεπεξεργασία δεδομένων αποτελεί μέθοδο εξόρυξης δεδομένων που περιλαμβάνει τη μετατροπή στοιχείων σε πιο αντιληπτή μορφή. Στόχος της παρούσας διπλωματικής είναι η εξέταση συνόλων δεδομένων και η εξαγωγή αποτελεσμάτων ακρίβειας πριν και μετά την προεπεξεργασία δεδομένων. Για να επιτευχθεί αυτό γίνεται χρήση πέντε αλγορίθμων κατηγοριοποίησης και οκτώ αλγορίθμων παλινδρόμησης. Στα πλαίσια των υπολογιστικών πειραμάτων έγιναν όλοι οι δυνατοί συνδυασμοί σε είκοσι διαφορετικά σύνολα δεδομένων (δέκα για κατηγοριοποίηση και δέκα για παλινδρόμηση). Κατά τη διάρκεια των υπολογιστικών πειραμάτων γίνεται η σύγκριση όλων των αλγορίθμων, οι οποίοι χρησιμοποιούν διαφορετικούς αλγόριθμους αντιμετώπισης ελλιπών τιμών (imputers) καθώς επίσης και αλγόριθμους κλιμάκωσης (scalers). Το τελικό συμπέρασμα που εξάγεται είναι η εύρεση του αλγορίθμου μηχανικής μάθησης που σε συνδυασμό με κάποιον αλγόριθμο κλιμάκωσης είναι ο καταλληλότερος και καλύτερος, δηλαδή θα έχει το μεγαλύτερο ποσοστό ακρίβειας σε σχέση με τους υπόλοιπους συνδυασμούς μετά την προεπεξεργασία των δεδομένων του κάθε συνόλου δεδομένων.

Λέξεις κλειδιά: μηχανική μάθηση, προεπεξεργασία δεδομένων, κατηγοριοποίηση, παλινδρόμηση, ελλιπείς τιμές, κλιμάκωση

Abstract

Data preprocessing is a method of data mining that involves converting data into a more perceptible form. The aim of the thesis is to examine datasets and extract accurate results before and after data processing. To achieve this, five classification algorithms and eight regression algorithms are used. As part of the computational experiments, all possible combinations were made in twenty different datasets (ten for classification and ten for regression). During the computational experiments, all combinations that use different algorithms for dealing with missing values (imputers) as well as scaling algorithms are compared (scalers). The final conclusion that is drawn is the machine learning algorithm which in combination with a scaling algorithm is the most appropriate and accurate, i.e., it will have the highest accuracy compared to the other combinations after the preprocessing of each dataset.

Keywords: machine learning, data preprocessing, classification, regression, imputing, scaling

Δήλωση Πνευματικών Δικαιωμάτων

Δηλώνω ρητά ότι, σύμφωνα με το άρθρο 8 του Ν. 1599/1986 και τα άρθρα 2,4,6 παρ. 3 του Ν. 1256/1982, η παρούσα Διπλωματική Εργασία με τίτλο "Τεχνικές προεπεξεργασίας δεδομένων στη μηχανική μάθηση" καθώς και τα ηλεκτρονικά αρχεία και πηγαίοι κώδικες που αναπτύχθηκαν ή τροποποιήθηκαν στα πλαίσια αυτής της εργασίας και αναφέρονται ρητώς μέσα στο κείμενο που συνοδεύουν, και η οποία έχει εκπονηθεί στο Τμήμα Ηλεκτρολόγων Μηχανικών & Μηχανικών Πληροφορικής του Πανεπιστημίου Δυτικής Μακεδονίας, υπό την επίβλεψη του μέλους του Τμήματος κ. Πλόσκα Νικόλαου αποτελεί αποκλειστικά προϊόν προσωπικής εργασίας και δεν προσβάλλει κάθε μορφής πνευματικά δικαιώματα τρίτων και δεν είναι προϊόν μερικής ή ολικής αντιγραφής, οι πηγές δε που χρησιμοποιήθηκαν περιορίζονται στις βιβλιογραφικές αναφορές και μόνον. Τα σημεία όπου έχω χρησιμοποιήσει ιδέες, κείμενο, αρχεία ή / και πηγές άλλων συγγραφέων, αναφέρονται ευδιάκριτα στο κείμενο με την κατάλληλη παραπομπή και η σχετική αναφορά περιλαμβάνεται στο τμήμα των βιβλιογραφικών αναφορών με πλήρη περιγραφή.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα. Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον συγγραφέα και μόνο.

Υπογραφή Φοιτητή

Περιεχόμενα

1 Εισαγωγή **10**

1.1 Τι είναι η προεπεξεργασία δεδομένων 10

1.2 Κίνητρο υλοποίησης της εργασίας 10

1.3 Στόχος και συμβολή της εργασίας 11

1.4 Επισκόπηση εργασίας 12

2 Μηχανική Μάθηση **13**

2.1 Τι είναι η μηχανική μάθηση 13

2.2 Μη εποπτευόμενη μάθηση 14

2.3 Εποπτευόμενη μάθηση 15

2.4 Ενισχυτική μάθηση 16

2.5 Ταυτόχρονη μηχανική μάθηση 16

2.6 Μαζική μάθηση 17

2.7 Προκλήσεις της μηχανικής μάθησης 18

2.7.1 Δεδομένα εκπαίδευσης που δεν είναι αντιπροσωπευτικά 18

2.7.2 Χαρακτηριστικά που δεν είναι σχετικά 19

2.7.3 Ανεπαρκής ποσότητα δεδομένων εκπαίδευσης 20

2.7.4 Δεδομένα κακής ποιότητας 20

2.8 Επικύρωση και δοκιμές 21

3 Προεπεξεργασία Δεδομένων **22**

3.1 Ελλιπή δεδομένα 22

3.1.1 Κατηγορίες τυχαιότητας ελλιπών δεδομένων 22

3.1.2 Τρόποι αντιμετώπισης ελλιπών δεδομένων 23

3.1.3 Μέθοδοι καταλογισμού (imputation) 24

3.1.4 Καταλογισμός χρησιμοποιώντας τον αλγόριθμο πλησιέστερων γειτόνων 25
3.1.5 Αντιμετώπιση ελλιπών δεδομένων με το CN2 και C4.5 26
3.1.6 Χρήση του αλγορίθμου KNN σε συνδυασμό με τον CN2 και C4.5 26
3.1.7 Πολλαπλός καταλογισμός (imputation) 28
3.1.8 Καταλογισμός με τη χρήση της μεθόδου Deep Learning - DataWig 29
3.2 Κωδικοποίηση χαρακτηριστικών 30
3.2.1 Μέθοδος κωδικοποίησης One-Hot 31
3.2.2 Μέθοδος κωδικοποίησης ετικετών 31
3.2.3 Μέθοδος ενσωμάτωσης κωδικοποίησης 32
3.3 Ομαλοποίηση ή τυποποίηση χαρακτηριστικών 33
3.4 Επιλογή των κατάλληλων χαρακτηριστικών 35
3.4.1 Μείωση διαστάσεων 35
3.4.2 Επιλογή χαρακτηριστικών μέσω γενετικών αλγορίθμων 37

4 Αλγόριθμοι Κλιμάκωσης και Αντιμετώπισης Ελλιπών Τιμών 38
4.1 Αλγόριθμοι κλιμάκωσης 38
4.2 Αλγόριθμοι αντιμετώπισης ελλιπών τιμών 41

5 Υπολογιστικά αποτελέσματα 43
5.1 Σύντομη περιγραφή του πειράματος 43
5.2 Η συλλογή των συνόλων δεδομένων 44
5.3 Τα εργαλεία που χρησιμοποιήθηκαν για την ανάπτυξη του πειράματος 50
5.4 Αλγόριθμοι που χρησιμοποιήθηκαν στην κατηγοριοποίηση και στην παλινδρόμηση 52
5.4.1 Μηχανές διανυσμάτων υποστήριξης 52
5.4.2 Δέντρα απόφασης 52
5.4.3 Πολυστρωματικό αντίληπτρο - νευρωνικά δίκτυα 53
5.4.4 Τυχαία δάση 54
5.4.5 k-πλησιέστερος γείτονας 54
5.5 Αλγόριθμοι που χρησιμοποιήθηκαν μόνο στην παλινδρόμηση 55
5.5.1 Γραμμική παλινδρόμηση 55
5.5.2 Παλινδρόμηση κορυφογραμμής 56

5.5.3 Παλινδρόμηση Lasso 56
5.6 Περιγραφή κώδικα 57
5.7 Σύγκριση αποτελεσμάτων για κάθε σύνολο δεδομένων 66
5.7.1 Σύγκριση αποτελεσμάτων για σύνολα δεδομένων κατηγοριοποίησης 66
5.7.2 Σύγκριση αποτελεσμάτων για σύνολα δεδομένων παλινδρόμησης 71
5.8 Συγκεντρωτική σύγκριση αποτελεσμάτων 78

6 Συμπεράσματα **81**

References **83**

Κατάλογος σχημάτων

2.1 Παράδειγμα ομαδοποίησης (*Fast Threshold Clustering Algorithm*, 2013) . 15
2.2 Ροή εργασίας μαζικής μάθησης (Bora, 2021) 18
2.3 Αντιπροσωπευτικό δείγμα εκπαίδευσης (Géron, 2019) 19
2.4 Η σημασία των δεδομένων σε σχέση με τους αλγόριθμους (*Distributed model training using Dask and Scikit-learn - Datafoam*, 2020) 20

3.1 Χρήση διάφορων μεθόδων στη διαχείριση μεγάλων συνόλων δεδομένων (Batista, Monard, et al., 2002) 28
3.2 Παράδειγμα κωδικοποίησης ετικετών (Panda & Misra, 2021) 32
3.3 Αρχιτεκτονική επιλογής χαρακτηριστικών (Khalid, Khalil, & Nasreen, 2014) 36

4.1 Απεικόνιση των δεδομένων μετά την κλιμάκωση (Zheng & Casari, 2018) 39
4.2 Σύνολο δεδομένων με ελλιπείς τιμές (Vikram, 2020) 41

5.1 Συγκεντρωτικά αποτελέσματα αλγορίθμων κατηγοριοποίησης 79
5.2 Συγκεντρωτικά αποτελέσματα αλγορίθμων παλινδρόμησης 80

Κατάλογος πινάκων

5.1 Αποτελέσματα συνόλου δεδομένων cars 66
5.2 Αποτελέσματα συνόλου δεδομένων abalone 66
5.3 Αποτελέσματα συνόλου δεδομένων bloodTransfusion 67
5.4 Αποτελέσματα συνόλου δεδομένων ticTacToe 68
5.5 Αποτελέσματα συνόλου δεδομένων balanceScale 68
5.6 Αποτελέσματα συνόλου δεδομένων bestSellersBooks 69
5.7 Αποτελέσματα συνόλου δεδομένων haberman 69
5.8 Αποτελέσματα συνόλου δεδομένων hayesRoth 70
5.9 Αποτελέσματα συνόλου δεδομένων lenses 70
5.10 Αποτελέσματα συνόλου δεδομένων drugTypes 71
5.11 Αποτελέσματα συνόλου δεδομένων aquaticToxicity 72
5.12 Αποτελέσματα συνόλου δεδομένων slumpTest 72
5.13 Αποτελέσματα συνόλου δεδομένων drivePoints 73
5.14 Αποτελέσματα συνόλου δεδομένων southGermanCredit 73
5.15 Αποτελέσματα συνόλου δεδομένων carPriceAssignment 74
5.16 Αποτελέσματα συνόλου δεδομένων insurance 75
5.17 Αποτελέσματα συνόλου δεδομένων realEstate 75
5.18 Αποτελέσματα συνόλου δεδομένων winequalityRed 76
5.19 Αποτελέσματα συνόλου δεδομένων diabetes 77
5.20 Αποτελέσματα συνόλου δεδομένων advertising 78

Κατάλογος απεικονίσεων

5.1 Δομή αρχείου classification.py 57
5.2 Δομή συνάρτησης init 58
5.3 Δομή συνάρτησης get_results 60
5.4 Δομή συνάρτησης find_best_scale_method 61
5.5 Δομή συνάρτησης get_reg_algorithms 62

Κεφάλαιο 1

Εισαγωγή

1.1 Τι είναι η προεπεξεργασία δεδομένων

Η προεπεξεργασία δεδομένων στον τομέα της μηχανικής μάθησης αποτελεί ένα σημαντικό κομμάτι της αλγοριθμικής ροής, διότι βοηθά στη βελτίωση της ποιότητας των στοιχείων ενός συνόλου δεδομένων προκειμένου να γίνει η εξαγωγή αρκετά χρήσιμων πληροφοριών από τα δεδομένα. Οργανώνει και καθαρίζει τα στοιχεία που είναι πρωτογενή για να είναι σε θέση να δοθούν ως είσοδος σε μοντέλα μηχανικής μάθησης και να τα εκπαιδεύσει. Αρκετές φορές τα δεδομένα στον πραγματικό κόσμο είναι ασυνεπή, ελλιπή και μπορεί να κρύβουν ακραίες τιμές μέχρι και σφάλματα. Για τους λόγους που προαναφέρθηκαν, η προεπεξεργασία δεδομένων έρχεται να βοηθήσει στην κατηγοριοποίηση, τη μορφοποίηση καθώς επίσης και στο καθάρισμα των στοιχείων που δεν έχουν υποστεί κάποιου είδους επεξεργασία προκειμένου να είναι σε θέση να χρησιμοποιηθούν από μοντέλα και αλγόριθμους μηχανικής μάθησης. Μερικά βήματα για να επιτευχθεί η προεπεξεργασία δεδομένων είναι η αξιολόγηση του συνόλου δεδομένων που πρόκειται να εξεταστεί, να εξαχθούν συμπεράσματα και να γίνει η εισαγωγή στο προγραμματιστικό περιβάλλον. Στη συνέχεια, θα γίνει η αναγνώριση και η διαχείριση των τιμών που αγνοούνται, η κωδικοποίηση των στοιχείων που είναι κατηγορηματικά σε αριθμητικά, ο διαχωρισμός του συνόλου στοιχείων και τέλος, η κλιμάκωση χαρακτηριστικών.

1.2 Κίνητρο υλοποίησης της εργασίας

Η προεπεξεργασία δεδομένων διαδραματίζει σημαντικό ρόλο διότι μέσω αυτής της διαδικασίας γίνεται η μετατροπή των δεδομένων σε μορφή κατανοητή για τον

υπολογιστή. Οι βάσεις δεδομένων αποτελούνται από ελλιπή στοιχεία και στοιχεία με θόρυβο, έχοντας ως συνέπεια να προκαλέσουν σύγχυση και αποτελέσματα που δεν είναι επιθυμητά. Τα χαρακτηριστικά των βάσεων δεδομένων χωρίζονται σε κατηγορηματικά και αριθμητικά. Κατηγορηματικά μπορούν να χαρακτηριστούν για παράδειγμα οι μήνες ενός χρόνου: [Ιανουάριος, Φεβρουάριος, Μάρτιος, Απρίλιος, Μάιος, Ιούνιος, Ιούλιος, Αύγουστος, Σεπτέμβριος, Οκτώβριος, Νοέμβριος, Δεκέμβριος] που αποτελούν μια κατηγορία διότι η τιμή εξάγεται από αυτό το σύνολο. Ως αριθμητικά χαρακτηρίζονται οι τιμές που είναι ακέραιες ή συνεχείς. Εκπροσωπούνται από αριθμούς και έχουν τις παρόμοιες ιδιότητες με αυτούς. Το κίνητρο της παρούσας διπλωματικής εργασίας είναι η αντιμετώπιση των παραπάνω προβλημάτων χρησιμοποιώντας τεχνικές προεπεξεργασίας δεδομένων όπως είναι η ενσωμάτωση δεδομένων, ο καθορισμός δεδομένων, η μείωση δεδομένων και ο μετασχηματισμός τους. Οι μετασχηματισμοί των δεδομένων μπορούν να βελτιώσουν αισθητά την αποτελεσματικότητα αλλά και την ακρίβεια ενός αλγορίθμου μηχανικής μάθησης. Ο καθαρισμός δεδομένων τροποποιεί τις τιμές που λείπουν και αφαιρεί τις ακραίες τιμές.

1.3 Στόχος και συμβολή της εργασίας

Προκειμένου να επιτευχθεί με σωστό τρόπο η προεπεξεργασία δεδομένων ενός συγκεκριμένου συνόλου δεδομένων θα πρέπει να γίνει η κατάλληλη αξιολόγηση και εξέταση του περιεχομένου και των στοιχείων του. Η πειραματική διαδικασία έχει ως σκοπό αφού πρώτα λάβει ως είσοδο τα σύνολα δεδομένων κατηγοριοποίησης και παλινδρόμησης να εκτελέσει και να εξάγει ως αποτέλεσμα για κάθε σύνολο έναν ή περισσότερους αλγόριθμους ως τους καλύτερους σύμφωνα με το ποσοστό ακρίβειας που συγκεντρώνουν. Εκτός αυτού θα εξαχθούν και τα συμπεράσματα για το ποιος ή ποιοι αλγόριθμοι σε συνδυασμό με έναν αλγόριθμο κλιμάκωσης ήταν οι καλύτεροι συγκεντρωτικά σε όλα τα σύνολα δεδομένων, ποιοι αλγόριθμοι ευνοήθηκαν περισσότερο έπειτα από την προπεξεργασία δεδομένων (αντιμετώπισης ελλιπών τιμών καθώς επίσης και η κωδικοποίηση τυχόν κατηγορηματικών τιμών) και τη συμμετοχή κάποιου αλγόριθμου κλιμάκωσης. Τέλος, θα πραγματοποιείται και η εύρεση του καλύτερου αλγορίθμου αντιμετώπισης ελλιπών τιμών συγκεντρωτικά σε όλα τα σύνολα.

1.4 Επισκόπηση εργασίας

Η παρούσα διπλωματική εργασία είναι δομημένη σε έξι κεφάλαια. Στο δεύτερο κεφάλαιο περιγράφεται ο ορισμός της μηχανικής μάθησης και οι τεχνικές εφαρμογής της όπως για παράδειγμα είναι η εποτευόμενη μάθηση, η ενισχυτική μάθηση καθώς επίσης και οι διάφορες προκλήσεις της που πρέπει να αντιμετωπιστούν όπως χαρακτηριστικά που είναι κακής ποιότητας και είναι άσχετα μεταξύ τους. Στο τρίτο κεφάλαιο αναλύονται έννοιες και τρόποι αντιμετώπισης ελλιπών δεδομένων, κωδικοποίησης, ομαλοποίησης καθώς επίσης και επιλογής κατάλληλων χαρακτηριστικών. Το τέταρτο κεφάλαιο περιγράφει τους αλγόριθμους κλιμάκωσης και αντιμετώπισης ελλιπών τιμών που χρησιμοποιήθηκαν στα πειράματα της διπλωματικής εργασίας. Στο πέμπτο κεφάλαιο επεξηγείται η πειραματική διαδικασία. Πιο συγκεκριμένα, περιγράφονται τα εργαλεία που βοήθησαν στην ανάπτυξη του πειράματος, οι συλλογές συνόλων δεδομένων και οι αλγόριθμοι μηχανικής μάθησης που χρησιμοποιήθηκαν. Επιπλέον πραγματοποιείται η επεξήγηση του τρόπου ανάπτυξης της πειραματικής διαδικασίας και παρουσιάζονται τα αποτελέσματα σύγκρισης των αλγορίθμων τόσο μεμονωμένα όσο και συγκεντρωτικά για σύνολα δεδομένων κατηγοριοποίησης και παλινδρόμησης. Στο έκτο και τελευταίο κεφάλαιο επισημαίνονται τα συμπεράσματα που εξήχθησαν από τη διπλωματική εργασία.

Κεφάλαιο 2

Μηχανική Μάθηση

2.1 Τι είναι η μηχανική μάθηση

Η μηχανική μάθηση είναι ο τρόπος με τον οποίο μπορεί να ενσωματωθεί τεχνητή νοημοσύνη σε έναν υπολογιστή. Αποτελεί υποπεδίο στην επιστήμη των υπολογιστών και δημιουργήθηκε μέσω της υπολογιστικής θεωρίας και της αναγνώρισης προτύπων. Με λίγα λόγια μέσα από αυτήν μπορεί να πραγματοποιηθεί η εξαγωγή πληροφοριών από δεδομένα και οι υπολογιστές μπορούν να εκπαιδεύονται χωρίς να έχουν προγραμματιστεί με ρητό τρόπο. Χρησιμοποιείται σε καθημερινή βάση για να προτείνει τα τρόφιμα που θα πρέπει να αγοραστούν, ποια ταινία θα μπορούσε να δει κάποιος, μέχρι και την αναγνώριση φίλων σε φωτογραφίες. Ιστοσελίδες όπως είναι το Amazon και το Youtube κρύβουν από πίσω μοντέλα μηχανικής μάθησης (Müller & Guido, 2016). Συνήθως κάποια προβλήματα που έχουν λυθεί απαιτούν αρκετές παραμέτρους και εκεί ένας αλγόριθμος μηχανικής μάθησης μπορεί να κάνει τον κώδικα απλούστερο έχοντας ως στόχο την καλύτερη απόδοση του. Μπορεί να βρεθεί λύση για κάποια προβλήματα που είναι δύσκολα και πολύπλοκα. Συστήματα μηχανικής μάθησης μπορούν να είναι εύκολα προσαρμόσιμα και να μάθουν πληροφορίες από μεγάλες ποσότητες νέων δεδομένων. Τύποι μηχανικής μάθησης μπορεί να είναι αυτοί που εκπαιδεύονται με την επίβλεψη του ανθρώπου (μη εποπτευόμενη, εποπτευόμενη μάθηση), αυτοί που μπορούν να μαθαίνουν σταδιακά και αυτοί που είτε κάνουν συγκρίσεις ανάμεσα σε γνωστά και νέα σημεία στοιχείων είτε κατασκευάζουν μοντέλα πρόβλεψης αφού πρώτα εξετάσουν τα μοτίβα στοιχείων που σχετίζονται με αυτά της εκπαίδευσης. Πρέπει να σημειωθεί πως αυτές οι τεχνικές δε λειτουργούν αποκλειστικά μόνες τους αλλά μπορούν να συνδυαστούν

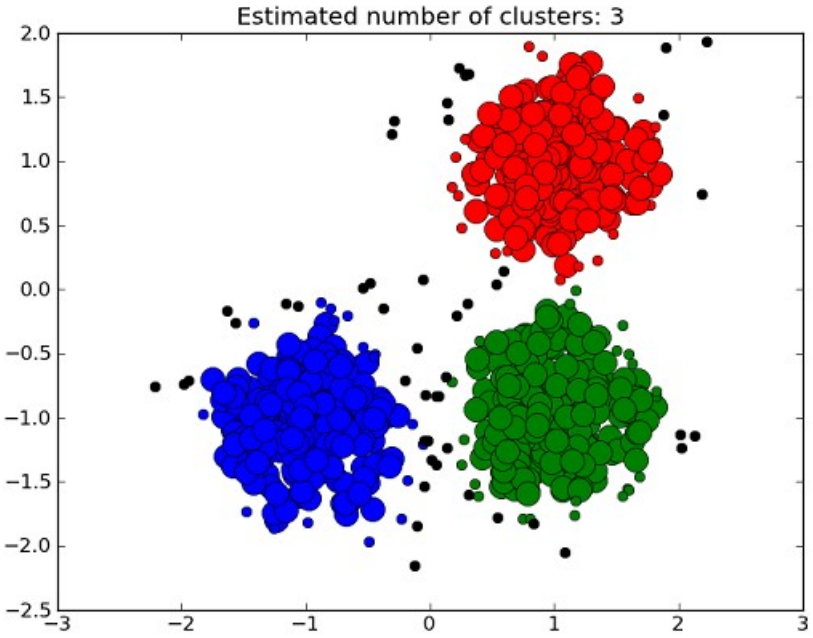
μεταξύ τους.

2.2 Μη εποπτευόμενη μάθηση

Η πρώτη κατηγορία μηχανικής μάθησης είναι η μη εποπτευόμενη μάθηση όπου η έξοδος της δεν είναι γνωστή από την αρχή. Στην κατηγορία αυτή γίνεται μόνο ο καθορισμός των δεδομένων στην είσοδο και ο αλγόριθμος της μηχανικής μάθησης καλείται να εξάγει ως αποτέλεσμα διάφορες πληροφορίες. Οι δύο διαφορετικοί τύποι στη μη εποπτευόμενη μάθηση είναι η ομαδοποίηση και ο μετασχηματισμός του συνόλου δεδομένων. Οι μετασχηματισμοί ενός συνόλου δεδομένων αποτελούν αλγόριθμους, οι οποίοι φτιάχνουν μια καινούργια αναπαράσταση των πληροφοριών με σκοπό να μπορούν να την καταλάβουν άλλοι αλγόριθμοι μηχανικής μάθησης ή ακόμη και οι άνθρωποι. Συνηθισμένη εφαρμογή της μη εποπτευόμενης μάθησης είναι η μείωση των διαστάσεων (dimensionality reduction) με αποτέλεσμα μια πιο απλουστευμένη μορφή που θα περιέχει λιγότερα δεδομένα, όπως για παράδειγμα είναι η μείωση των πολλών διαστάσεων σε δύο. Επιπλέον μια ακόμη εφαρμογή μπορεί να αποτελέσει και η έρευνα για την ανακάλυψη νέων στοιχείων ή τμημάτων που δημιουργούνται από τα δεδομένα. Οι αλγόριθμοι ομαδοποίησης διαφοροποιούν τα δεδομένα σε μεμονωμένες ομάδες παρόμοιων χαρακτηριστικών. Στο Σχήμα 2.1 απεικονίζεται ένα παράδειγμα ομαδοποίησης. Στη μη εποπτευόμενη μάθηση αποτελεί πρόκληση η αξιολόγηση που απαιτείται για να σιγουρευτεί κάποιος κατά πόσο ένας αλγόριθμος έμαθε κάποια σημαντική πληροφορία διότι η εφαρμογή των μη επιτηρούμενων αλγόριθμων εκμάθησης εφαρμόζονται σε δεδομένα όπου δεν περιλαμβάνουν πληροφορίες σχετικά με την ετικέτα εξόδου. Επίσης, οι αλγόριθμοι μη εποπτευόμενης μάθησης χρησιμοποιούνται ως ένα βήμα προεπεξεργασίας για τους αλγόριθμους εποπτευόμενης μάθησης. Η εκμάθηση μιας καινούργιας αναπαράστασης δεδομένων ενδέχεται ορισμένες φορές να βελτιώσει αρκετά το ποσοστό ακρίβειας των αλγόριθμων εποπτευόμενης μάθησης ή και ακόμη να ελαττώσει την κατανάλωση του χώρου και της μνήμης. Μερικοί αλγόριθμοι μη εποπτευόμενης μάθησης για ομαδοποίηση είναι η ιεραρχική ανύλυση συμπλέγματος (Hierarchical Cluster Analysis), η μεγιστοποίηση προσδοκιών (Expectation Maximization) και ο k-Means. Για εκμάθηση κανόνων σύνδεσης είναι ο Eclat και ο Apriori ενώ για μείωση διαστάσεων και οπτικοποίηση είναι ο πυρήνας (kernel) PCA, η τοπικά γραμμική

ενσωμάτωση (locally linear embedding), η ανάλυση κύριων συστατικών (principal component analysis) και ο t-διανομή στοχαστικής ενσωμάτωσης του γείτονα (t-distributed stochastic neighbor embedding) (Géron, 2019).

Σχήμα 2.1: Παράδειγμα ομαδοποίησης (*Fast Threshold Clustering Algorithm*, 2013)



2.3 Εποπτευόμενη μάθηση

Η εποπτευόμενη μάθηση αποτελεί έναν από τους πιο διαδεδομένους τύπους μηχανικής μάθησης. Χρησιμοποιείται για προβλέψεις ενός συγκεκριμένου αποτελέσματος έχοντας όμως από πριν ορισμένα ζεύγη εισόδων και εξόδων τα οποία αποτελούν το εκπαιδευτικό σύνολο όπου ο ίδιος ο άνθρωπος έχει δημιουργήσει. Στην εποπτευόμενη μάθηση υπάρχουν δύο τρόποι επίλυσης προβλημάτων η παλινδρόμηση και η κατηγοριοποίηση. Σκοπός των μεθόδων κατηγοριοποίησης είναι να ταξινομούν στοιχεία σε κατηγορίες με βάση κάποια κριτήρια. Ορισμένες φορές γίνεται ο διαχωρισμός της σε δυαδική κατηγοριοποίηση δηλαδή θα γίνεται η κατηγοριοποίηση μεταξύ δύο κατηγοριών και σε πολλών κλάσεων κατηγοριοποίηση όπου πλέον η κατηγοριοποίηση γίνεται σε περισσότερες από δύο κατηγορίες. Παραδείγματα μεθόδου παλινδρόμησης μπορεί να αποτελέσουν η πρόβλεψη ηλικίας ενός ανθρώπου και η πρόβλεψη του εισοδήματος του κάθε χρόνο. Η διάκριση μεταξύ των μεθόδων κατηγοριοποίησης και παλινδρόμησης μπορεί να γίνει αρκετά

εύκολα διαπιστώνοντας εάν στην έξοδο υπάρχει κάποιο συνεχές μοτίβο. Αρκετές φορές το πρόβλημα της υπερμοντελοποίησης στη μηχανική μάθηση παρουσιάζεται όταν ένα μοντέλο βρίσκεται αρκετά κοντά στις ιδιαιτερότητες του εκπαιδευτικού συνόλου αλλά δεν είναι αποτελεσματική η εφαρμογή του σε νέα δεδομένα. Ένα ακόμη πρόβλημα που μπορεί να προκύψει είναι όταν ένα μοντέλο είναι αρκετά απλουστευμένο με αποτέλεσμα να μην υπάρχει η δυνατότητα να επιτευχθεί σωστά η μεταβλητότητα και η καταγραφή όλων των πληροφοριών του εκπαιδευτικού συνόλου (Müller & Guido, 2016). Μερικοί από τους πιο σημαντικούς αλγόριθμους εποπτευόμενης μάθησης είναι οι εξής: Λογιστική Παλινδρόμηση (Logistic Regression), Υποστήριξη Διανυσματικών Μηχανών (Support Vector Machines), Γραμμική Παλινδρόμηση (Linear Regression), Νευρωνικά Δίκτυα (Neural Networks), k-πλησιέστεροι Γείτονες (k-Nearest Neighbors), Τυχαία Δάση (Random Forests) και Δέντρα Απόφασης (Decision Trees).

2.4 Ενισχυτική μάθηση

Η ενισχυτική μάθηση είναι αρκετά διαφορετική σε σχέση με όσες μεθόδους περιγράφηκαν προηγούμενως. Το εκπαιδευτικό σύστημα που αποκαλείται πράκτορας σε αυτήν την περίπτωση αποτελεί τον παρατηρητή του περιβάλλοντος και έχει ως σκοπό να επιλέξει και να εκτελέσει διεργασίες για να λάβει πίσω αποτελέσματα που θα τον βοηθήσουν να συνεχίσει. Ύστερα θα πρέπει η εκπαίδευση να γίνει από μόνη της προκειμένου να βρεθεί η πιο καλή πολιτική που θα έχει το καλύτερο αποτέλεσμα καθώς περνάει ο χρόνος. Βρίσκοντας την καλύτερη πολιτική θα μπορούσε να γίνει ο καθορισμός για την επιλογή του πράκτορα καθώς αυτός βρίσκεται σε μια κατάσταση που είναι καθορισμένη. Η ενισχυτική μάθηση χρησιμοποιείται κυρίως σε προβλήματα που αφορούν τον σχεδιασμό. Μερικοί αλγόριθμοι ενισχυτικής μάθησης είναι ο Sarsa, ο Q-μάθηση (Q-learning) καθώς επίσης και η Επαναληπτική Πολιτική (Policy Iteration).

2.5 Ταυτόχρονη μηχανική μάθηση

Στην ταυτόχρονη μηχανική μάθηση (online machine learning) η εκπαίδευση του συστήματος πραγματοποιείται σταδιακά τροφοδοτώντας τα δεδομένα με διαδοχικό

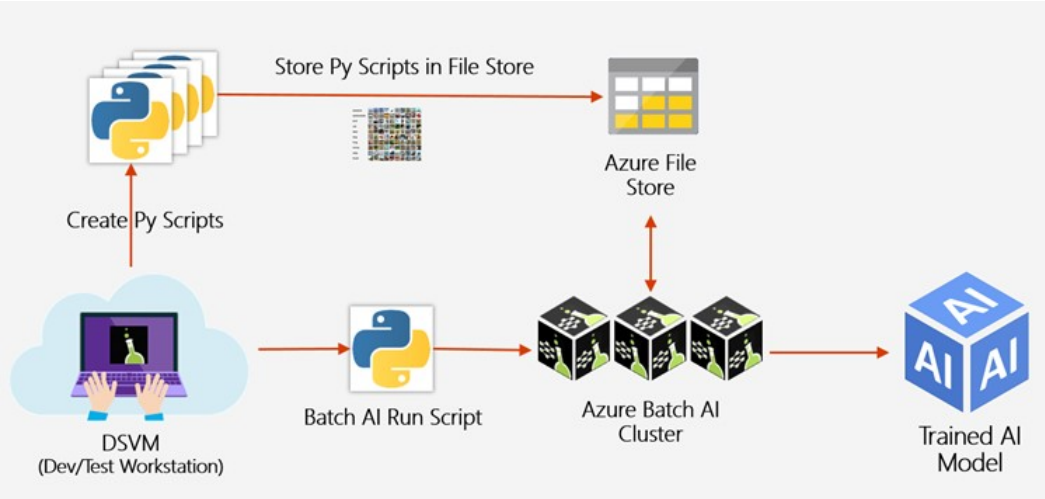
τρόπο, είτε σε μικρές ομάδες τύπου mini-batch είτε ένα-ένα. Η εκμάθηση γίνεται με μικρό κόστος και ταχυτήτα, δηλαδή το σύστημα έχει τη δυνατότητα να μαθαίνει για καινούργια δεδομένα ενώ βρίσκεται σε κατάσταση εκμάθησης. Αποτελεί επίσης μια από τις καλύτερες επιλογές σε περίπτωση που υπάρχει ελάχιστο πλήθος υπολογιστικών πόρων. Για παράδειγμα, ένα σύστημα που σκοπεύει να μάθει για νέα δεδομένα, τα παλαιότερα δεν τα έχει ανάγκη πλέον και μπορεί να τα απορρίψει μόνο στην περίπτωση όμως που δε θέλει να γυρίσει στην προηγούμενη κατάσταση που βρισκόταν. Με αυτόν τον τρόπο μπορεί να επιτευχθεί αρκετά μεγάλη εξοικονόμηση χώρου. Επιπλέον, οι διαδικτυακοί αλγόριθμοι εκμάθησης μπορούν να εφαρμοστούν όταν υπάρχει η επιθυμία να γίνει η εκπαίδευση συστημάτων σε μεγάλο όγκο δεδομένων και δεν είναι εφικτό να προσαρμοστούν στην υπάρχουσα μνήμη του συστήματος. Ο αλγόριθμος είναι υπεύθυνος στη φόρτωση ενός μέρους από τα δεδομένα, στην εφαρμογή σταδίου εκπαίδευσης στα δεδομένα αυτά και στην επανάληψη της διαδικασίας μέχρι αυτή να εφαρμοστεί σε όλο το σύνολο των δεδομένων. Αρκετά σημαντική παράμετρος της εκμάθησης διαδικτυακών συστημάτων αποτελεί η προσαρμογή των δεδομένων που αλλάζουν μορφή, γνωστό και ως ποσοστό εκμάθησης. Αν το ποσοστό εκμάθησης οριστεί αρκετά ψηλά, τότε το σύστημα θα έχει τη δυνατότητα να κάνει προσαρμογή με πιο γρήγορο ρυθμό σε νέα στοιχεία, διαγράφοντας από τη μνήμη του εξίσου γρήγορα τα παλαιότερα στοιχεία. Στην περίπτωση που οριστεί μικρό ποσοστό εκμάθησης θα πρέπει το σύστημα να έχει μεγαλύτερη αδράνεια. Με λίγα λόγια δε θα παρουσιάζει μεγάλη ευαισθησία στον θόρυβο που παρουσιάζουν ακολουθίες στοιχείων ή νέα δεδομένα αλλά θα μαθαίνει με πιο αργό ρυθμό. Μια πρόκληση που καλείται η ταυτόχρονη μηχανική μάθηση να αντιμετωπίσει είναι η περίπτωση που τροφοδοτηθούν κακής ποιότητας δεδομένα στο σύστημα. Προκειμένου να μειωθεί ο κίνδυνος αυτός θα πρέπει να γίνεται συχνή παρακολούθηση του συστήματος και να τερματιστεί αμέσως η εκμάθηση εάν υπάρξει μείωση στην απόδοση.

2.6 Μαζική μάθηση

Η εκπαίδευση του συστήματος στη μαζική μάθηση (batch learning) θα πρέπει να υλοποιηθεί με όλα τα δεδομένα που είναι διαθέσιμα. Επειδή όμως αυτό απαιτεί αρκετούς υπολογιστικούς πόρους και είναι αρκετά χρονοβόρο θα πρέπει να γίνει

σε κατάσταση εκτός σύνδεσης από το διαδίκτυο. Αρχικά, το σύστημα ξεκινάει την εκπαίδευση του και ύστερα εκτελείται ξανά εφαρμόζοντας ό,τι έμαθε. Στην περίπτωση που ένα σύστημα τύπου μαζικής εκμάθησης μάθει πληροφορίες για νέα δεδομένα, θα πρέπει να γίνει εκπαίδευση της καινούργιας έκδοσης του συστήματος από την αρχή σε όλο το φάσμα των δεδομένων και η αντικατάσταση του παλιού με το καινούργιο σύστημα. Αυτή η λύση από τη μια πλευρά είναι μια καλή επιλογή, από την άλλη πλευρά όμως απαιτεί πολύ χρόνο για να γίνει η εκπαίδευση ενός νέου συστήματος ξεκινώντας από 24 ώρες έως και εβδομάδες. Τέλος, στην περίπτωση που ο όγκος των στοιχείων είναι αρκετά μεγάλος τότε η μέθοδος μαζικής μάθησης δεν αποτελεί τη βέλτιστη επιλογή. Μια καλύτερη λύση αποτελούν οι αλγόριθμοι που μπορούν να εκπαιδεύονται με τρόπο σταδιακό (Géron, 2019). Στο Σχήμα 2.2 αποτυπώνεται ένα παράδειγμα ροής εργασίας στη μαζική μάθηση.

Σχήμα 2.2: Ροή εργασίας μαζικής μάθησης (Bora, 2021)

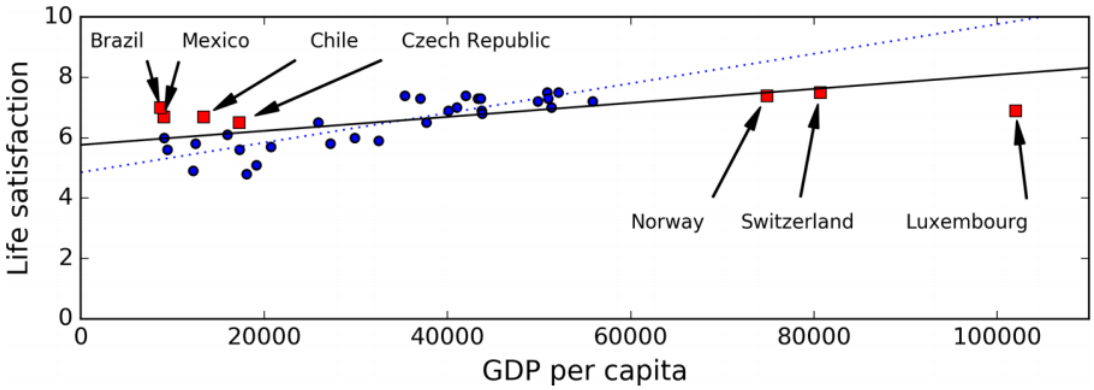


2.7 Προκλήσεις της μηχανικής μάθησης

2.7.1 Δεδομένα εκπαίδευσης που δεν είναι αντιπροσωπευτικά

Αρκετά σημαντικό γεγονός αποτελούν τα δεδομένα που προορίζονται για εκπαίδευση και αφορούν νέες περιπτώσεις να είναι αντιπροσωπευτικά. Στην περίπτωση που εκπαιδευτεί ένα μοντέλο που είναι γραμμικού τύπου με αυτού του είδους τα δεδομένα το αποτέλεσμα που θα ληφθεί είναι μια σταθερή και μια διακεκομμένη γραμμή που παριστάνει το παλαιότερο μοντέλο. Μετά την προσθήκη κάποιων δεδομένων το μοντέλο τροποποιείται σε μεγάλο βαθμό και στην περίπτωση που είναι

Σχήμα 2.3: Αντιπροσωπευτικό δείγμα εκπαίδευσης (Géron, 2019)



αρκετά απλοϊκό υπάρχει μεγάλη πιθανότητα η λειτουργία του να μην είναι τόσο ικανοποιητική. Από το Σχήμα 2.3 θα θεωρούσε κάποιος ότι οι πλουσιότερες χώρες δεν είναι πιο ”ευτυχισμένες” απο τις φτωχότερες. Από την άλλη πλευρά όμως κάποιες από τις φτωχότερες χώρες δείχνουν ότι είναι περισσότερο ”ευτυχισμένες” από μεγάλο αριθμό πλουσιότερων χωρών. Η χρήση ενός εκπαιδευτικού συνόλου που είναι αντιπροσωπευτικό κατάφερε να εκπαίδευσει ένα μοντέλο και να έχει ακριβέστερες προβλέψεις. Γι’ αυτό το λόγο θα πρέπει να δοθεί βάση στην επιλογή ενός πιο αντιπροσωπευτικού εκπαιδευτικού συνόλου για περιπτώσεις που πρέπει να γενικευτούν. Αν γίνει χρήση μεγάλου αριθμού δειγμάτων που δεν είναι αντιπροσωπευτικά τότε η δειγματοληψία δε θα πραγματοποιηθεί με σωστό τρόπο (Géron, 2019).

2.7.2 Χαρακτηριστικά που δεν είναι σχετικά

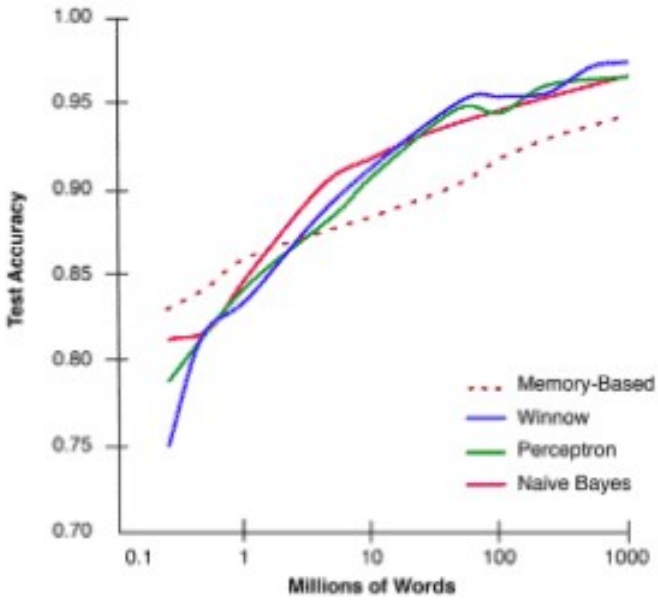
Μεγάλο πάραγοντα αποτελεί το σύστημα να εκπαιδευτεί από δεδομένα που σχετίζονται μεταξύ τους. Αυτή η ανάγκη είναι γνωστή και ως μηχανική χαρακτηριστικών και αποτελεί ένα σημαντικό μέρος προκειμένου ένα μοντέλο μηχανικής μάθησης να εκπαιδευτεί σωστά με τον βέλτιστο δυνατό τρόπο. Η μηχανική χαρακτηριστικών αποτελείται από τρεις διαφορετικές τεχνικές. Η πρώτη τεχνική είναι η κατασκευή καινούργιων χαρακτηριστικών κάνοντας συλλογή νέων δεδομένων. Μια ακομή τεχνική αποτελεί η εξαγωγή των χαρακτηριστικών στην οποία συνδυάζονται τα χαρακτηριστικά που υπάρχουν ηδή προκειμένου να δημιουργηθεί ένας συνδυασμός που θα είναι πιο χρήσιμος. Η τρίτη και τελευταία τεχνική είναι η επιλογή των πιο χρήσιμων χαρακτηριστικών από όλα τα διαθέσιμα προκειμένου η διαδικασία

να προχωρήσει στην εκπαίδευση του μοντέλου.

2.7.3 Ανεπαρκής ποσότητα δεδομένων εκπαίδευσης

Στη μηχανική μάθηση ακόμη και σε απλά προβλήματα υπάρχει η ανάγκη να υπάρχουν αρκετά δεδομένα προκειμένου να εκτελεστούν με σωστό τρόπο οι περισσότεροι αλγόριθμοι. Στο Σχήμα 2.4 αποτυπώνεται η σημασία των δεδομένων σε σχέση με τους αλγόριθμους μηχανικής μάθησης.

Σχήμα 2.4: Η σημασία των δεδομένων σε σχέση με τους αλγόριθμους (*Distributed model training using Dask and Scikit-learn - Datafoam*, 2020)



Υπάρχουν αρκετοί ερευνητές (π.χ., (Halevy, Norvig, & Pereira, 2009)) που υποστηρίζουν πως οι αλγόριθμοι έχουν μικρότερη σημασία σε σχέση με τα δεδομένα. Όμως τα δεδομένα που είναι μέτριου ή μικρού μεγέθους θα συνεχίσουν να παρουσιάζουν ομοιότητες μεταξύ τους και δε θα είναι εφικτό πάντοτε η ανθρωπότητα να κερδίζει μεγάλο αριθμό δεδομένων όποτε δε θα πρέπει να παραμεληθούν οι αλγόριθμοι.

2.7.4 Δεδομένα κακής ποιότητας

Στην περίπτωση που τα δεδομένα εκπαίδευσης περιέχουν λάθη, δηλαδή παρουσιάζουν θόρυβο ή ακραίες τιμές, το σύστημα θα δυσκολευτεί να κάνει σωστό εντοπισμό των υποκείμενων προτύπων με αποτέλεσμα να μην έχει και την καλύτερη δυνατή απόδοση. Γι' αυτό το λόγο θα πρέπει να γίνεται ο καθαρισμός των δεδο-

μένων που προορίζονται για εκπαίδευση. Αν στο σύνολο δεδομένων σε κάποιες περιπτώσεις υπάρχει έλλειψη στοιχείων θα πρέπει να μη ληφθούν υπόψη, να γίνει η εκπαίδευση του μοντέλου χωρίς τα στοιχεία που λείπουν ή να συμπληρωθούν οι τιμές που είναι ελλιπείς. Τέλος, υπάρχει η περίπτωση στο σύνολο δεδομένων να υπάρχουν τιμές που είναι ακραίες και θα πρέπει είτε να διορθωθούν είτε να απορριφθούν.

2.8 Επικύρωση και δοκιμές

Ένα μοντέλο μπορεί να διαπιστωθεί πόσο κάλα γενικεύει νέα δεδομένα αφού πρώτα δοκιμαστεί πάνω σε αυτά. Μια επιλογή αποτελεί το μοντέλο να τεθεί σε στάδιο παραγωγής και να παρακολουθείται ανά χρονικά διαστήματα η πορεία του και πόσο ορθά δουλεύει με τα νέα δεδομένα. Το ποσοστό που σχετίζεται με το σφάλμα των νέων δεδομένων είναι γνωστό και ως σφάλμα γενίκευσης και η εκτίμηση του ποσοστού αυτού γίνεται κάνοντας την επικύρωση του μοντέλου σε ένα σύνολο από διαδοχικές δοκιμές. Αρκετές φορές υπάρχει το ερώτημα ως προς την επιλογή του κατάλληλου μοντέλου δηλαδή θα πρέπει να προτιμηθεί το πολυωνυμικό ή το γραμμικό. Στην περίπτωση που το γραμμικό μοντέλο παρουσιάζει καλύτερη γενίκευση θα πρέπει στη συνέχεια να υποστεί κανονικοποίηση προκειμένου να μην υπάρξει αρκετά μεγάλο ποσοστό υπερμοντελοποίησης. Ένας τρόπος κανονικοποίησης αποτελεί η επιλογή εκατό διαφορετικών μοντέλων κάνοντας χρήση εκατό διαφορετικών τιμών σε μια υπερπαράμετρο (hyperparameter). Η τιμή της υπερπαραμέτρου που βρέθηκε ως καλύτερη σε σχέση με τις υπόλοιπες που θα εξεταστούν ενδέχεται σε επίπεδο παραγωγής να μη δουλέψει όπως θα περίμενε κανείς. Η λύση σε αυτό το πρόβλημα αποτελεί η ύπαρξη ενός συνόλου επικύρωσης. Με αυτόν τον τρόπο μπορεί να γίνει η εκπαίδευση αρκετών μοντέλων με μια πληθώρα από υπερπαραμέτρους κάνοντας χρήση του εκπαιδευτικού συνόλου και η επιλογή των κατάλληλων υπερπαραμέτρων που έχουν τη μέγιστη δυνατή απόδοση στο σύνολο επικύρωσης. Στη συνέχεια, θα πρέπει να εκτελεστεί μια τελική δοκιμή στο δοκιμαστικό σύνολο προκειμένου να εκτιμηθεί το σφάλμα γενίκευσης (Géron, 2019).

Κεφάλαιο 3

Προεπεξεργασία Δεδομένων

3.1 Ελλιπή δεδομένα

Ένα σημαντικό πρόβλημα που αφορά την εξόρυξη των δεδομένων αποτελεί η ύπαρξη των ελλιπών στοιχείων. Τα δεδομένα που λείπουν ενδέχεται να μην υπάρχουν για διάφορους λόγους. Επίσης, συχνό φαινόμενο αποτελεί ένα μέγαλο μέρος των δεδομένων να είναι εσφαλμένο και η μόνη αντιμετώπιση αυτού του προβλήματος είναι η απόρριψη αυτών των στοιχείων. Όμως, παρόλο που υπάρχει η έλλειψη των στοιχείων σε πάρα πολλά σύνολα δεδομένων, οι αλγόριθμοι της μηχανικής μάθησης συνεχίζουν να τα αντιμετωπίζουν με αφελές τρόπο. Αν τα δεδομένα που λείπουν δεν αντιμετωπιστούν με προσεκτικό τρόπο τότε ενδέχεται να προκαλέσουν προβλήματα κατά την εκπαίδευση ενός μοντέλου. Αρκετές φορές είναι συχνό φαινόμενο τα χαρακτηριστικά συνόλων δεδομένων να έχουν σχέση με άλλα χαρακτηριστικά και με αυτόν τον τρόπο οι τιμές που είναι άγνωστες να μπορέσουν να βρεθούν και να είναι πλέον γνωστές. Με τη βοήθεια των αλγόριθμων αντιμετώπισης ελλιπών τιμών (imputers) μπορεί να γίνει η αντικατάσταση των άγνωστων τιμών. Πλεονέκτημα των αλγόριθμων αυτών αποτελεί η επεξεργασία των δεδομένων που είναι άγνωστα και δε σχετίζονται με τον αλγόριθμο εκμάθησης που θα χρησιμοποιηθεί (Batista & Monard, 2003).

3.1.1 Κατηγορίες τυχαιότητας ελλιπών δεδομένων

Η τυχαιότητα των ελλιπών δεδομένων μπορεί να διαχωριστεί σε τρεις κατηγορίες. Η πρώτη κατηγορία ονομάζεται έλλειψη τυχαιότητας. Στην κατηγορία αυτή όταν η πιθανότητα μιας οντότητας έχει τιμή που είναι άγνωστη, ενδέχεται να έχει

στενή σύνδεση με πεδία που περιέχουν γνωστές τιμές. Η δεύτερη κατηγορία αποκαλείται ολοκληρωτική έλλειψη της τυχαιότητας. Είναι εμφανής στην περίπτωση που η πιθανότητα μιας οντότητας έχει τιμή που είναι άγνωστη από ένα χαρακτηριστικό και δεν έχει καμία απολύτως εξάρτηση από πεδία με τιμές που είναι ήδη γνωστά. Αν συμβαίνει αυτό τότε υπάρχει η ελευθερία να εφαρμοστεί χωρίς να υπάρξει κάποιο πρόβλημα που να έχει σχέση με την εμφάνιση του φαινομένου της προκατάληψης οποιαδήποτε μέθοδος τροποποίησης των δεδομένων. Η τρίτη κατηγορία είναι γνωστή ως "μη έλλειψη της τυχαιότητας", δηλαδή η τιμή ενός χαρακτηριστικού που είναι άγνωστη έχει στενή εξάρτηση από την τιμή του ίδιου χαρακτηριστικού (Batista & Monard, 2003).

3.1.2 Τρόποι αντιμετώπισης ελλιπών δεδομένων

Αρκετές μέθοδοι, όπως για παράδειγμα η υποκατάσταση περιπτώσεων, δημιουργήθηκαν προκειμένου να αντιμετωπίσουν ελλιπή δεδομένα τα οποία σχετίζονται με δειγματοληπτικές έρευνες και εμφανίζουν κάποια προβλήματα στην περίπτωση που η εφαρμογή τους γίνεται σε περιβάλλον που αφορά την εξόρυξη δεδομένων. Υπάρχουν και άλλες μέθοδοι όπως η αντικατάσταση των τιμών που είναι άγνωστες και μπορούν να προσδιοριστούν κάνοντας χρήση του αριθμού που εμφανίζεται πιο συχνά σε ένα σύνολο (mode) ή του μέσου όρου (mean), αλλά θα πρέπει η χρήση τους να γίνει αρκετά προσεκτικά προκειμένου να μη δημιουργηθούν προβλήματα προκατάληψης. Οι μέθοδοι επεξεργασίας ελλιπών δεδομένων μπορούν να χωριστούν σε τρεις κατηγορίες. Η πρώτη κατηγορία ονομαζέται εκτίμηση παραμέτρων. Σε αυτήν την κατηγορία γίνεται χρήση διαφόρων διαδικασιών μέγιστης δυνατής πιθανότητας προκειμένου να επιτευχθεί η εκτίμηση των παραμέτρων ενός μοντέλου το οποίο θα οριστεί για δεδομένα που είναι πλήρη. Έχουν τη δυνατότητα να χειρίζονται διαδικασίες πιθανοτήτων που είναι μέγιστες και κάνουν χρήση τροποποιημένου αλγόριθμου, ο οποίος στοχεύει στην εκτίμηση παραμέτρων σε δεδομένα που είναι άγνωστα. Η δεύτερη κατηγορία είναι η απόρριψη και η παράβλεψη δεδομένων. Υπάρχουν δύο τρόποι που απορρίπτουν δεδομένα με τιμές που δεν είναι γνωστές. Ο πρώτος τρόπος είναι η ανάλυση των περιπτώσεων που γίνεται σε πλήρη βαθμό και αποτελεί προεπιλεγμένο τρόπο σε μεγάλο αριθμό λογισμικών αλλά η χρήση του μπορεί να γίνει και σε αρκετά αν όχι όλα τα στατιστικά πακέτα. Ο τρόπος αυτός

απορρίπτει όλα τα πεδία που έχουν άγνωστες τιμές. Ο δεύτερος τρόπος απορρίπτει παρουσίες ή χαρακτηριστικά. Συμμετέχει στον προσδιορισμό του πλήθους των δεδομένων κάποιου χαρακτηριστικού που είναι ελλιπές και στην κατάργηση τους με αρκετά μεγάλα επίπεδα δεδομένων που είναι άγνωστα. Πριν όμως γίνει η κατάργηση ενός χαρακτηριστικού θα πρέπει πρώτα να αξιολογηθεί ο συσχετισμός του με τα υπόλοιπα χαρακτηριστικά. Η τρίτη και τελευταία κατηγορία είναι ο καταλογισμός (imputation). Με τη βοήθεια της μπορούν να συμπληρωθούν τιμές πεδίων που είναι άγνωστα μέσω γνωστών τιμών ή σχέσεων μεταξύ των χαρακτηριστικών του συνόλου (Batista & Monard, 2003).

3.1.3 Μέθοδοι καταλογισμού (imputation)

Οι μέθοδοι καταλογισμού έχουν ως στόχο να αντικαταστήσουν τα πεδία που είναι άγνωστα με τη βοήθεια εκτιμώμενων πληροφοριών που προκύπτουν από το σύνολο δεδομένων. Μια από τις πιο συνηθισμένες μεθόδους είναι ο μέσος καταλογισμός. Χρησιμοποιείται προκειμένου να τροποποιήσει τα δεδομένα που είναι ελλιπή με γνωστές τιμές κάνοντας χρήση είτε του αριθμού που εμφανίζεται πιο συχνά σε ένα σύνολο για ποιοτικά χαρακτηριστικά είτε του μέσου όρου για ποσοτικά χαρακτηριστικά. Δεύτερη μέθοδο αποτελεί η τροποποίηση των περιπτώσεων και η χρήση της γίνεται κυρίως σε δειγματοληπτικές έρευνες. Σκοπός της είναι να αλλάξει τις τιμές που λείπουν με μια τιμή που δεν αντικατοπτρίζει κάποιο δείγμα. Τρίτη μέθοδος αποτελεί το cold and hot deck (κρύο και ζεστό κατάστρωμα). Με τη χρήση της hot deck το πεδίο κάποιου χαρακτηριστικού που είναι άγνωστο αντικαθιστάται από μια εκτίμηση της κατανομής για την τιμή που λείπει από τα υπάρχοντα δεδομένα του συνόλου δεδομένων που είναι γνωστά και αυτό υλοποιείται σε δύο στάδια. Στο πρώτο στάδιο τα δεδομένα του συνόλου διαχωρίζονται σε συμπλέγματα και στο δεύτερο στάδιο κάθε πεδίο που λείπει συνδέεται με μια από τις ομάδες που δημιουργήθηκαν και αυτό επιτυγχάνεται μέσω του αριθμού που εμφανίζεται πιο συχνά σε ένα σύνολο ή του μέσου όρου σε ένα σύμπλεγμα. Το cold deck λειτουργεί με τρόπο παρόμοιο του hot deck με τη διαφορά όμως πως η πηγή του συνόλου δεδομένων θα πρέπει να μην είναι ίδια με την τρέχουσα. Τελευταία μέθοδο αποτελεί το μοντέλο πρόβλεψης το οποίο περιέχει διάφορες προηγμένες διεργασίες για την αντιμετώπιση δεδομένων που λείπουν. Το χαρακτηριστικό που περιέχει δεδομένα που

είναι άγνωστα χρησιμοποιείται ως χαρακτηριστικό κλάσης ενώ τα υπόλοιπα χρησιμοποιούνται ως δεδομένα εισόδου στο μοντέλο που πρόκειται να κάνει προβλέψεις και αυτό είναι χρήσιμο επειδή σε αρκετές περιπτώσεις τα δεδομένα παρουσιάζουν συνδέσεις μεταξύ τους. Με τη βοήθεια των συνδέσεων αυτών μπορούν να κατασκευαστούν προγνωστικά μοντέλα παλινδρόμησης ή κατηγοριοποίησης για ποσοτικά και ποιοτικά ελλιπή δεδομένα. Μειονέκτημα της μεθόδου αυτής αποτελούν οι τιμές που εκτιμούνται και παρουσιάζουν στις περισσότερες περιπτώσεις πιο καλή συμπεριφορά από τις τιμές που είναι πραγματικές. Ένα δεύτερο μειονέκτημα είναι πως υπάρχει η απαίτηση, τα πεδία που είναι άγνωστα να έχουν σχέση με κάποια άλλα χαρακτηριστικά του συνόλου. Στην περίπτωση που δεν υπάρχουν συσχετίσεις με άλλα χαρακτηριστικά τότε το μοντέλο δε θα είναι τόσο ακριβές προκειμένου να βρει τις τιμές που είναι άγνωστες (Batista & Monard, 2003).

3.1.4 Καταλογισμός χρησιμοποιώντας τον αλγόριθμο πλησιέστερων γειτόνων

Η χρήση του αλγορίθμου πλησιέστερων γειτόνων παρέχει τη δυνατότητα να προβλέπονται τόσο τα ποσοτικά όσο και τα ποιοτικά χαρακτηριστικά και δεν είναι υποχρεωτικό να κατασκευαστεί προγνωστικό μοντέλο για κάθε ένα από τα χαρακτηριστικά που είναι άγνωστα. Αυτό που συμβαίνει πραγματικά είναι πως ο αλγόριθμος KNN δε φτιάχνει εξαρχής κάποιο μοντέλο και με αυτόν τον τρόπο αρκετά εύκολα προσαρμόζει τη λειτουργία του έτσι ώστε να δουλεύει βέλτιστα με κάθε χαρακτηριστικό, λαμβάνοντας υπόψη τα χαρακτηριστικά που πρέπει να μεταβληθούν στο μέτρημα της απόστασης αντιμετωπίζοντας όλες τις περιπτώσεις με τιμές που είναι άγνωστες. Μειονέκτημα του αλγόριθμου αυτού είναι η αναζήτηση παρόμοιων περιπτώσεων ανάμεσα σε όλα τα δεδομένα του συνόλου. Ένας τρόπος επίλυσης του προβλήματος αυτού μπορεί να αποτελέσει η κατασκευή μειωμένου εκπαιδευτικού συνόλου για τον πλησιέστερο γείτονα, ο οποίος περιέχει πρωτότυπα δεδομένα κάνοντας χρήση των M-δέντρων (M-tree). Τα M-δέντρα έχουν τη δυνατότητα να κάνουν αναζήτηση και οργάνωση των δεδομένων μέσω του γενικού μετρικού χώρου και μπορούν να ελαττώσουν σε σημαντικό βαθμό τον αριθμό των υπολογισμών μέσω των παρόμοιων ερωτημάτων (Batista & Monard, 2003).

3.1.5 Αντιμετώπιση ελλιπών δεδομένων με το CN2 και C4.5

Οι τιμές που είναι άγνωστες ενδέχεται να υπάρχουν σε οποιοδήποτε χαρακτηριστικό με εξαίρεση το χαρακτηριστικό της κλάσης. Στην περίπτωση που ληφθεί υπόψη ένα εκπαιδευτικό σύνολο που θα έχει το όνομα T, ο αλγόριθμος C4.5 έπειτα από πολλαπλές δοκιμές θα βρίσκει κάνοντας χρήση μόνο ενός χαρακτηριστικού ένα ή και πιο πολλά αποτελέσματα που είναι ομαδικά και συμβολίζονται $O_1, O_2, \cdots, O_n$. Στην περίπτωση που υπάρχουν άγνωστα πεδία σε ένα χαρακτηριστικό με το όνομα X τότε ο C4.5 μέσω του υποσυνόλου που περιέχει γνωστές τιμές θα μπορέσει να βγάλει συμπεράσματα και τιμές που λείπουν στα υπόλοιπα πεδία. Αμέσως μετά την επιλογή μιας από τις δοκιμές, οι οποίες βασίζονται στο χαρακτηριστικό X, ο C4.5 μέσω της πιθανοτικής προσέγγισης θα διαχωρίσει τις υπάρχουσες τιμές που είναι άγνωστες στο χαρακτηριστικό X. Μια παρουσία που περιέχει γνωστές τιμές, αποθηκεύεται σε ένα στοιχείο συνόλου T_i, και συμβολίζεται με ένα, ενώ σε όλα τα άλλα υποσύνολα με μηδέν. Ο αλγόριθμος C4.5 συνδέει σε κάθε παρουσία του T_i ένα βάρος το οποίο δείχνει την πιθανότητα της παρουσίας που υπάρχει στο T_i. Στην περίπτωση που το στιγμιότυπο περιέχει γνωστή τιμή και ικανοποιεί τη δοκιμή με αποτέλεσμα το O_i, τότε η παρουσία είναι αντίστοιχη της T_i με βάρος ένα. Αν όμως η παρουσία αυτή περιέχει τιμή που δεν είναι γνωστή εξαρχής τότε η παρουσία αυτή θα αντιστοιχίζεται με μέρη που το κάθε ένα από αυτά έχει διαφορετικό βάρος. Η εκτίμηση της πιθανότητας γίνεται αφού πραγματοποιηθεί άθροιση των βάρων από τις παρουσίες του T που ήδη είναι γνωστές και ικανοποιούν τη δοκιμή με το αποτέλεσμα O_i. Στη συνέχεια, διαιρείται με το άθροισμα των βαρών των περιπτώσεων στο T το οποίο περιέχει γνωστές τιμές στο χαρακτηριστικό X. Ο αλγόριθμος CN2 κάνει χρήση μιας πιο απλής μεθόδου για τις άγνωστες τιμές όπου την κάθε τιμή που λείπει την αντικαθιστά με την πιο κοινή τιμή του χαρακτηριστικού, πριν όμως γίνει ο υπολογισμός του μέτρου εντροπίας (Batista et al., 2002).

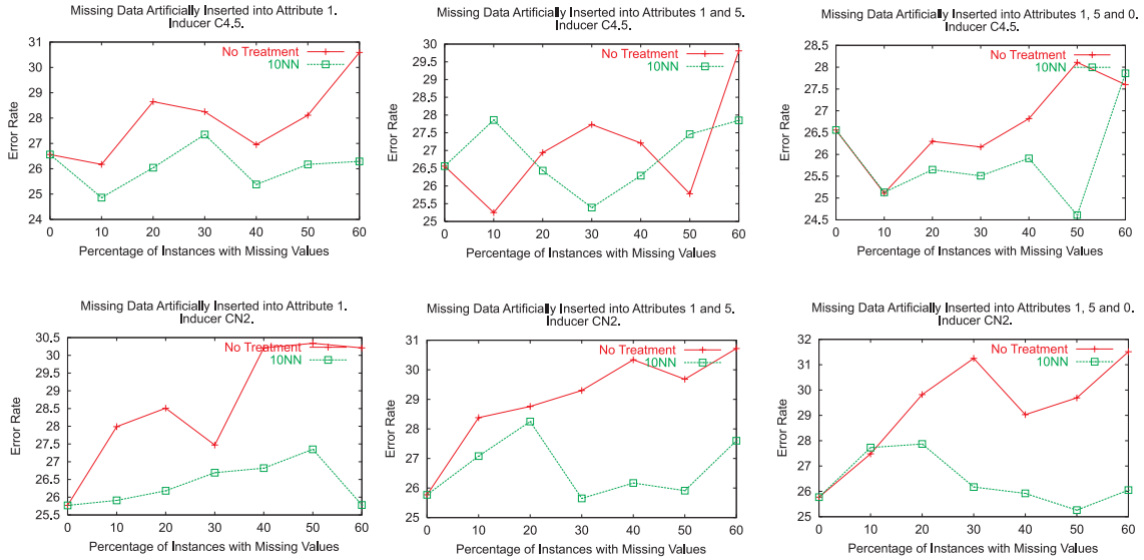
3.1.6 Χρήση του αλγορίθμου KNN σε συνδυασμό με τον CN2 και C4.5

Η σύγκριση μεταξύ των αλγορίθμων κατά κύριο λόγο είναι η ανάλυση των συμπεριφορών τους όταν αυτοί καλούνται να αντιμετωπίσουν μεγάλο αριθμό άγνωστων πεδίων. Σε συγκριτικό πείραμα που πραγματοποιήθηκε χρησιμοποιώντας ως σύνολα δεδομένων τα Bupa (*UCI Machine Learning Repository: Liver Disorders Data*

Set, 2021), CMC (*UCI Machine Learning Repository: Contraceptive Method Choice Data Set*, 2021) και Pima (Learning, 2021) έγινε ο διαχωρισμός τους σε δέκα ζευγάρια εκπαιδευτικών και δοκιμαστικών συνόλων κάνοντας χρήση της μεθόδου 10 πτυχών διασταυρούμενης επικύρωσης (10 fold cross validation). Έπειτα υλοποιείται η εκχώρηση των τιμών που είναι άγνωστες στο εκπαιδευτικό σύνολο. Με αυτόν τον τρόπο δημιουργούνται συνολικά τέσσερα αντίγραφα του εκπαιδευτικού συνόλου όπου τα δύο από αυτά δίνονται στους αλγόριθμους CN2 και C4.5 χωρίς να γίνει η επεξεργασία σε δεδομένα που λείπουν. Τα άλλα δύο αντίγραφα του εκπαιδευτικού συνόλου δίνονται στον αλγόριθμο KNN προκειμένου να γίνει η αντικατάσταση και εκτίμηση των τιμών που είναι άγνωστες. Αφού πλέον έχει γίνει η αντικατάσταση των δεδομένων που είναι άγνωστα τα άλλα δύο σύνολα εκπαίδευσης παρέχονται στους αλγόριθμους C4.5 και CN2. Μετά το πέρας των δέκα επαναλήψεων δίνεται η δυνατότητα να πραγματοποιηθεί η εκτίμηση του πραγματικού ποσοστού που αφορά το σφάλμα αφού πρώτα γίνει ο υπολογισμός του μέσου όρου όλων των ποσοστών σφάλματος της κάθε επανάληψης. Τέλος, οι επιδόσεις των CN2 και C4.5 που έχουν σχέση με τη μέθοδο που επεξεργάζεται τα δεδομένα και είναι άγνωστα μπορούν να αναλυθούν και στη συνέχεια να συγκριθούν με την απόδοση της μεθόδου που χρησιμοποιείται από τους αλγόριθμους C4.5 και CN2 για να εκπαιδευτούν όταν υπάρχουν τιμές που είναι ελλιπείς. Για να γίνει η εκχώρηση των άγνωστων δεδομένων στα εκπαιδευτικά σύνολα θα πρέπει να επιλεχθούν κάποια χαρακτηριστικά από τα οποία ορισμένες τιμές θα πρέπει να τροποποιηθούν σε άγνωστες. Επίσης, θα πρέπει να ληφθεί υπόψη πως η επιλογή των πιο αντιπροσωπευτικών χαρακτηριστικών ενός συνόλου δεδομένων δεν είναι μια απλή διαδικασία και θα πρέπει να επιλεχθούν τα τρία χαρακτηριστικά που είναι πιο σχετικά σύμφωνα με τις μεθόδους επιλογής υποσυνόλου χαρακτηριστικών. Ύστερα θα πρέπει τα δεδομένα που είναι ελλιπή να εκχωρηθούν με τυχαίο τρόπο σε ποσοστά που είναι ίσα με 10, 20, 30, 40, 50 και 60 τις εκατό του συνόλου των παρουσιών. Οι τιμές που είναι άγνωστες αντικαταστάθηκαν από τιμές χρησιμοποιώντας 1, 3, 5, 10, 20, 30, 50 έως και 100 πλησιέστερους γείτονες. Αφού ληφθούν υπόψη τα αποτελέσματα που φαίνονται στο Σχήμα 3.1 η απόδοση του 10-NNI είναι αρκετά πιο μεγάλη από την απόδοση των δύο αλγορίθμων C4.5 και CN2 για το σύνολο δεδομένων με το όνομα Bupa. Στο σύνολο δεδομένων CMC η απόδοση του 10-NNI είναι μεγαλύτερη σε σχέση με την

απόδοση που έχει επιτευχθεί, χωρίς να υπάρχει η απουσία της επεξεργασίας δεδομένων. Επιπλέον, στο Σχήμα 3.1 φαίνονται τα αποτελέσματα που συγκρίνονται για το σύνολο δεδομένων Pima όπου είναι εμφανές πως η μέθοδος 10-NNI βρίσκεται ελάχιστα πιο μπροστά σε απόδοση σε σύγκριση με τον αλγόριθμο C4.5 και πολύ ανώτερη της μεθόδου CN2 (Batista et al., 2002).

Σχήμα 3.1: Χρήση διάφορων μεθόδων στη διαχείριση μεγάλων συνόλων δεδομένων (Batista et al., 2002)



3.1.7 Πολλαπλός καταλογισμός (imputation)

Ο πολλαπλός καταλογισμός αποτελεί γενική προσέγγιση στο πρόβλημα των ελλιπών δεδομένων. Στο πρώτο στάδιο υλοποιείται η κατασκευή πολλών διαφορετικών αντιγράφων του συνόλου δεδομένων αντικαθιστώντας τις τιμές που είναι άγνωστες με γνωστές. Η διαδικασία του πολλαπλού καταλογισμού θα πρέπει να λαμβάνει υπόψη όλες τις πιθανές αβεβαιότητες που ενδέχεται να εμφανιστούν κατά την πρόβλεψη των άγνωστων τιμών μέσω της κατάλληλης μεταβλητότητας σε αρκετές πιθανές τιμές. Στο δεύτερο στάδιο θα γίνει η χρήση των τυπικών στατιστικών μεθόδων προκειμένου να γίνει η αντιστοιχία του μοντέλου ενδιαφέροντος σε κάθε ένα από τα πιθανά σύνολα δεδομένων. Οι πιθανές συνδέσεις σε κάθε ένα από τα πιθανά σύνολα θα είναι διαφορετικές επειδή υπάρχει διακύμανση που εμφανίζεται καθώς γίνεται ο υπολογισμός των άγνωστων τιμών και είναι χρήσιμες μόνο στην περίπτωση που υπολογιστεί ο μέσος όρος προκειμένου να δοθούν οι συνολικές συσχετίσεις που εκτι-

μήθηκαν. Η εκτίμηση και ο υπολογισμός των ποσοστών που αφορούν τα σφάλματα γίνεται με βάση τους κανόνες Rubin ("Chapter 9 Rubin's Rules", 2020), οι οποίοι έχουν ως κριτήριο τη μεταβλητότητα των αποτελεσμάτων ανάμεσα στα τεκμαρτά σύνολα δεδομένων περιγράφοντας την αβεβαιότητα που σχετίζεται με τις τιμές που είναι άγνωστες. Ο πολλαπλός καταλογισμός απαιτεί από τον χρήστη να υλοποιήσει τη μοντελοποίηση της κατανομής που αφορά την κάθε μεταβλητή με τιμές που είναι άγνωστες. Θα πρέπει η διαδικασία της μοντελοποίησης να γίνεται με προσεκτικό τρόπο για να είναι τα αποτελέσματα έγκυρα (Sterne et al., 2009).

3.1.8 Καταλογισμός με τη χρήση της μεθόδου Deep Learning - DataWig

Το DataWig (*DataWig documentation*, 2020) αποτελεί ένα ολοκληρωμένο πακέτο λογισμικού που έχει ως σκοπό να μειώσει την προσπάθεια που απαιτείται προκειμένου να ελαχιστοποιηθεί η απώλεια καταλογισμού σε σύνολα δεδομένων. Οι περισσότερες διεργασίες αντιμετωπίζουν τη διαδικασία του καταλογισμού για πιο ετερογενείς τύπους δεδομένων και αναφέρονται σε δυαδικές, κατηγορηματικές ή κανονικές μεταβλητές που έχουν την ευχέρεια να τροποποιηθούν σε αριθμητικές αναπαραστάσεις. Με τη βοήθεια του DataWig μπορούν να συμπληρωθούν οι υπάρχουσες βιβλιοθήκες μέσω μιας λύσης καταλογισμού για πίνακες που εμπεριέχουν τιμές που είναι κατηγορηματικού ή αριθμητικού τύπου αλλά και πιο γενικούς τύπους όπως για παράδειγμα είναι το μη δομημένο κείμενο. Το DataWig επιλέγει με αυτόματο τρόπο έναν αριθμό από χαρακτηριστικά. Το μοντέλο DataWig βασίζεται σε καθιερωμένες προσεγγίσεις ακολουθώντας την προσέγγιση του μοντέλου MICE στο οποίο κάθε στήλη θα πρέπει να προσδιοριστεί από τον χρήστη και ενδέχεται να περιέχει αρκετά χρήσιμες πληροφορίες που αφορούν τον καταλογισμό. Ο κώδικας του μοντέλου DataWig δίνει τη δυνατότητα στους χρήστες να κάνουν επέκταση των τύπων με αρκετά εύκολο τρόπο σε ακολουθίες και εικόνες. Το API δίνει τη δυνατότητα να γίνει ο καταλογισμός των τιμών που είναι άγνωστες χρησιμοποιώντας πίνακα σε ένα μέρος από τα δεδομένα τύπου pandas και καθορίζοντας ποιες θα είναι οι στήλες εξόδου και οι στήλες εισόδου. Σε διαφορετική περίπτωση όλες οι τιμές που δεν είναι γνωστές σε ένα μέρος από τα δεδομένα ενδέχεται να βρεθούν κάνοντας χρήση της συνάρτησης SimpleImputer.complete(). Επιπρόσθετα το DataWig περιέχει ένα σύνολο από χαρακτηριστικά τα οποία βοηθούν προκειμένου

να επιτυγχάνεται η αυτοματοποίηση του καταλογισμού. Οι τύποι δεδομένων που υπάρχουν κάνουν χρήση χαρακτηριστικών που μαθαίνονται αυτόματα καθώς γίνεται η εκπαίδευση του μοντέλου. Όλες οι αρχιτεκτονικές νευρωνικών δικτύων και οι υπερπαραμέτροι εξελίσσονται κάνοντας χρήση της τυχαίας αναζήτησης, η οποία ενδέχεται σε ορισμένες περιπτώσεις να περιοριστεί σε καθορισμένο χρονικό περιθώριο. Οι πιθανότητες των αποτελεσμάτων ενός μοντέλου βαθμονομούνται με τρόπο αυτόματο στο σύνολο επικύρωσης και στην περίπτωση που υπάρξουν αμφιβολίες για τον τρόπο καταλογισμού πραγματοποιείται ο υπολογισμός των στηλών εισόδου προκειμένου να γίνει ακόμη πιο κατανοητός ο τρόπος του καταλογισμού. Τέλος, το μοντέλο αποτελείται από λειτουργικά εργαλεία τα οποία αντισταθμίζουν την τροποποίηση που υλοποιείται στην ετικέτα ανάμεσα στην παραγωγή δεδομένων που δεν έχουν ετικέτα και στην εκπαίδευση (Biessmann et al., 2019).

3.2 Κωδικοποίηση χαρακτηριστικών

Αρκετά συχνό φαινόμενο αποτελεί τα σύνολα από δεδομένα να εμπεριέχουν πεδία που αποτελούν συμβολοσειρές. Με τη βοήθεια όμως στατιστικών μοντέλων δίνεται η δυνατότητα να μετατραπούν οι συμβολοσειρές σε αριθμητικές αναπαραστάσεις, αφού πρώτα κατασκευαστεί αναπαράσταση που είναι διανυσματική. Αρχικά, θα πρέπει να ληφθεί υπόψη πως ένα σύνολο από κατηγορίες μπορεί να είναι τεράστιο και να μην είναι προσδιορισμένο όπως θα ήταν για παράδειγμα ένας αριθμός από διαφορετικές συμβολοσειρές. Δεύτερον, κάποιες κατηγορίες ενδέχεται να έχουν σύνδεση μεταξύ τους, δηλαδή να έχουν σημασιολογικούς δεσμούς ή να έχουν κοινή μορφολογία. Μια από τις τεχνικές κωδικοποίησης κατηγορηματικών μεταβλητών που προορίζεται για στατιστική ανάλυση αποτελεί η τεχνική one-hot. Η μέθοδος αυτή κατασκευάζει διανύσματα που βρίσκονται σε ίση απόσταση και ορθογώνια. Ωστόσο υπάρχει ένα μεγάλο ποσοστό καρδινιλότητας το οποίο αναγκάζει τα χαρακτηριστικά να βρίσκονται σε υψηλή διάσταση. Η υψηλή καρδινιλότητα δημιουργεί αρκετά σημαντικό πρόβλημα στη ρύθμιση των δεδομένων και κατά επέκταση σε υψηλό αριθμό κατηγοριών με αποτέλεσμα να υπάρχουν στατιστικά και υπολογιστικά προβλήματα (Cerda & Varoquaux, 2020).

3.2.1 Μέθοδος κωδικοποίησης One-Hot

Αρκετοί αλγόριθμοι στατιστικής μάθησης έχουν την απαίτηση να λαμβάνουν ως είσοδο έναν αριθμητικό πίνακα χαρακτηριστικών. Στην περίπτωση που υπάρχουν κατηγορηματικές μεταβλητές στα δεδομένα θα πρέπει μέσω της μηχανικής χαρακτηριστικών να γίνει η κωδικοποίηση των διάφορων κατηγοριών με τη χρήση ενός κατάλληλου διανύσματος χαρακτηριστικών. Η κωδικοποίηση One-Hot αποτελεί μια αρκετά γνωστή και απλουστευμένη μέθοδο κωδικοποίησης. Για παράδειγμα, μια μεταβλητή που είναι κατηγορηματική και έχει ως κατηγορίες θηλυκό, αρσενικό, άλλο μπορεί να κωδικοποιηθεί σε διανύσματα τρισδιάστατων χαρακτηριστικών: [1, 0, 0], [0, 1, 0], [0, 0, 1]. Στη συνέχεια, θα προκύψει διανυσματικός χώρος όπου η κάθε κατηγορία θα είναι ορθογώνια και ισάξια με τις υπόλοιπες και θα είναι σύμφωνη με τις κλασσικές διαισθήσεις που αφορούν τις ονομαστικές κατηγορηματικές μεταβλητές. Τα μη επιμελημένα κατηγορηματικά δεδομένα αρκετές φορές ενδέχεται να οδηγήσουν σε υψηλότερη καρδινιλότητα της κατηγορικής μεταβλητής και προκύπτουν πολλαπλά προβλήματα κατά την κωδικοποίηση One-Hot. Μια πρόκληση που θα πρέπει να αντιμετωπιστεί είναι όταν το σύνολο των δεδομένων περιλαμβάνει μορφολογικές αναπαραστάσεις που είναι διαφορετικές μεταξύ τους αλλά ανήκουν στην ίδια κατηγορία. Παραδείγματος χάρη για μια κατηγορηματική μεταβλητή που έχει το όνομα εταιρεία, δεν είναι σαφές εάν οι «Pfizer International LLC», «Pfizer Limited» και «Pfizer Korea» είναι διαφορετικά ονόματα για την παρόμοια οντότητα, όμως μπορεί να σχετίζονται. Διάφορα σφάλματα όπως τυπογραφικά λάθη ενδέχεται να προκαλέσουν μορφολογικές παραλλαγές των κατηγοριών. Αν δε διορθωθούν οι διαφορετικές αναπαραστάσεις συμβολοσειρών της ίδιας κατηγορίας, θα εξάγουν ως αποτέλεσμα διαφορετικούς κωδικοποιημένους φορείς. Μια δεύτερη πρόκληση μπορεί να αποτελέσει η κωδικοποίηση κατηγοριών που δεν εμφανίζονται στο εκπαιδευτικό σύνολο. Τέλος, οι κατηγορηματικές μεταβλητές με αρκετά μεγάλη καρδινιλότητα δυσκολεύουν και καθιστούν ανέφικτη την κωδικοποίηση One-Hot (Cerda & Varoquaux, 2020).

3.2.2 Μέθοδος κωδικοποίησης ετικετών

Αυτή η μέθοδος μετατρέπει κάθε τιμή που υπάρχει σε μια στήλη με έναν αντιπροσωπευτικό αριθμό. Για παράδειγμα, ένα σύνολο από δεδομένα που αφορούν

τύπους ονομάτων από γέφυρες. Αρχικά, θα πρέπει να γίνει η επιλογή για κωδικοποίηση πεδίων που περιέχουν κείμενο αφού γίνει τοποθέτηση κάποιας ακολουθίας για κάθε πεδίο του κειμένου δηλάδη τα πεδία [Arch, Beam, Truss] να μετατραπούν αντίστοιχα σε αναπαράσταση [0, 1, 2]. Με αυτόν τον τρόπο θα ολοκληρωθεί η κωδικοποίηση της ετικέτας μεταβλητού τύπου γέφυρας. Το πρόβλημα που ενδέχεται να παρουσιαστεί κάποιες φορές όμως ανάλογα με τον τύπο των δεδομένων και τις τιμές των πεδίων είναι η ακολουθία των αριθμών που χρησιμοποιείται. Με λίγα λόγια η χρήση ενός αντιπροσωπευτικού αριθμού προκαλεί σύγχυση κατά τη σύγκριση μεταξύ των αριθμών παρόλο που δεν υπάρχει σχέση ανάμεσα στους διάφορους τύπους γέφυρας. Στην περίπτωση που κάποιος όμως κοιτάξει μόνο τους αριθμούς τότε μπορεί να καταλάβει πως το πεδίο Truss έχει μεγαλύτερη προτεραιότητα από το πεδίο Arch. Αυτό θα έχει ως αποτέλεσμα ο αλγόριθμος να θεωρήσει πως τα δεδομένα έχουν κάποιο είδος ιεραρχίας τύπου $0 < 1 < 2 \ldots < 6$ και να δώσει μεγαλύτερο βάρος σε κάποια κατηγορία. Τέλος, στο Σχήμα 3.2 αποτυπώνεται ενδεικτικό παράδειγμα κωδικοποίησης ετικετών.

Σχήμα 3.2: Παράδειγμα κωδικοποίησης ετικετών (Panda & Misra, 2021)

BRIDGE-TYPE (TEXT)	BRIDGE-TYPE (NUMERICAL)
Arch	0
Beam	1
Truss	2
Cantilever	3
Tied Arch	4
Suspension	5
Cable	6

3.2.3 Μέθοδος ενσωμάτωσης κωδικοποίησης

Η ενσωματωμένη κωδικοποίηση αποτελεί κατανεμημένη αναπαράσταση δεδομένων που είναι κατηγορηματικά. Κάθε μια από τις κατηγορίες χαρτογραφείται σε ένα διακριτό φορέα και το διάνυσμα που περιλαμβάνει ιδιότητες, οι οποίες μαθαίνουν καθώς το νευρικό δίκτυο εκπαιδεύεται. Ο διανυσματικός χώρος παρέχει

μια προβολή που περιλαμβάνει κατηγορίες που είναι σε κοντινή απόσταση ή έχουν κάποια στενή σύνδεση με το σύμπλεγμα. Μέσα από αυτό παρέχονται πλεονεκτήματα όπως η εκμάθηση τόσο των σχέσεων που έχουν τα δεδομένα μεταξύ τους, όσο και η μέθοδος κωδικοποίησης One-Hot κατά την παροχή κάποιας αναπαράστασης που είναι διανυσματική και αφορά κάποια κατηγορία. Σε αντίθεση με τη μέθοδο κωδικοποίησης One-Hot, τα διανύσματα που παρέχονται στην είσοδο δεν περιλαμβάνουν αρκετές μηδενικές τιμές. Όμως το μειονέκτημα είναι οτί απαιτείται να γίνει η εκμάθηση του μοντέλου και η κατασκευή αρκετών ακόμη μεταβλητών που θα δοθούν στην είσοδο. Η τεχνική αυτή αναπτύχθηκε για να παρέχει κατανεμημένες αναπαραστάσεις που αφορούν λέξεις. Στη συνέχεια, η τεχνική χρησιμοποιήθηκε στην ενσωμάτωση λέξεων από κείμενα όπου αυτά μέσω αλγορίθμων μαθαίνουν ανεξάρτητες αναπαραστάσεις από κάποιο νευρωνικό δίκτυο. Ένα επιπλέον πλεονέκτημα της μεθόδου αυτής είναι πως τα διανύσματα που εκπαιδεύονται, χαρτογραφούν κάθε μια από τις κατηγορίες και μπορούν να χωρέσουν σε ένα μοντέλο που έχει μέτρια δεξιότητα με την προϋπόθεση πως μπορεί να γίνει η εξαγωγή των φορέων και να χρησιμοποιηθούν ως είσοδος σε μια σειρά από διαφορετικές εφαρμογές και μοντέλα. Επιπλέον απαιτείται να υπάρχει ένα επίπεδο ενσωμάτωσης για κάθε μεταβλητή που είναι κατηγορηματικού τύπου και η ενσωμάτωση δηλώνει πως η κωδικοποίηση κάθε κατηγορίας θα πραγματοποιείται κανονικά. Τέλος, κάθε ενσωμάτωση έχει ως προαπαιτούμενο να οριστεί ο αριθμός των διαστάσεων που θα χρησιμοποιηθούν για την αναπαράσταση του διανυσματικού χώρου. Οι πιο συνηθισμένες τιμές διαστάσεων είναι 50, 100 και σε ορισμένες περιπτώσεις 300 (Brownlee, 2020α΄).

3.3 Ομαλοποίηση ή τυποποίηση χαρακτηριστικών

Σε αρκετές περιπτώσεις οι μεταβλητές εισόδου που είναι κλιμακωτές ενδέχεται να οδηγήσουν σε μια ασταθή ή και αργή διαδικασία μάθησης, ενώ οι μη κλιμακωτές μεταβλητές στόχου σε προβλήματα παλινδρόμησης μπορούν να οδηγήσουν σε αποτέλεσματα αποτυχίας της μαθησιακής διαδικασίας. Η προεπεξεργασία δεδομένων περιλαμβάνει τεχνικές τυποποίησης ή ομαλοποίησης που μπορούν να εφαρμοστούν τόσο στις μεταβλητές εισόδου όσο και στις μεταβλητές εξόδου πριν προχωρήσει η διαδικασία στην εκπαίδευση ενός μοντέλου νευρωνικού δικτύου. Θα πρέπει πάντα

η κατανομή των δεδομένων και η κλίμακα να αντλούνται από τον τομέα που ενδέχεται να μην είναι ίδιος για κάθε μια από τις μεταβλητές. Μεταβλητές εισόδου όπως είναι τα χιλιόμετρα ή οι ώρες αποτελούν μεταβλητές που έχουν κλίμακες διαφορετικές μεταξύ τους. Όταν υπάρχουν διαφορετικές κλίμακες ανάμεσα στις μεταβλητές εισόδου θα υπάρξει μεγάλη δυσκολία στη μοντελοποίηση ενός προβλήματος. Ένα μοντέλο το οποίο περιλαμβάνει μεγάλες τιμές βάρους αρκετές φορές θα παρουσιάσει αστάθειες, το οποίο σημαίνει πως ενδέχεται να εμφανίσει κακή απόδοση καθώς μαθαίνει και την ευαισθησία σε τιμές που δίνονται ως είσοδος με αποτέλεσμα ένα μεγάλο σφάλμα γενίκευσης. Ύπάρχουν και περιπτώσεις όπου μια μεταβλητή που είναι στόχος και έχει υψηλή κατανομή τιμών να οδηγήσει σε υψηλές τιμές διαβάθμισης των σφαλμάτων με αποτέλεσμα να υπάρξει δραματική τροποποίηση των τιμών βάρους καθιστώντας τη διαδικασία εκμάθησης ασταθή. Ένας κανόνας που θα πρέπει να ισχύει είναι πως οι μεταβλητές που δίνονται στην είσοδο θα πρέπει να είναι μικρές τιμές είτε τυποποιημένες με μηδενικό μέσο όρο είτε ανάμεσα στο εύρος από το μήδεν μέχρι το ένα. Αν η ποσότητα της κατανομής είναι φυσιολογική τότε θα πρέπει να τυποποιηθεί, σε διαφορετική περίπτωση τα δεδομένα θα πρέπει να υποστούν κανονικοποίηση. Αυτό θα ισχύει μόνο αν οι ποσοτικές τιμές έχουν μεγάλο εύρος όπως για παράδειγμα 10-100 ή μικρότερο από 0,01 σε 0,0001. Μόνο στην περίπτωση που οι ποσοτικές τιμές είναι μικρές ανάμεσα στο 0 εώς το 1 και η κατανομή είναι περιορισμένη τότε ενδέχεται να μη χρειάζονται κλιμάκωση τα δεδομένα. Όταν υπάρχουν αμφιβολίες για το αν θα πρέπει να ομαλοποιηθούν τα δεδομένα θα πρέπει να εξεταστούν οι πόροι, να ελεγχθεί η μοντελοποίηση με τα δεδομένα που δεν έχουν υποστεί κάποιου είδους επεξεργασία ή τα δεδομένα που είναι τυποποιημένα και να εξεταστεί αν θα υπάρξει διαφορά στην απόδοση του μοντέλου. Η κανονικοποίηση αποτελεί την επανασύνδεση των δεδομένων από ένα αρχικό εύρος έτσι ώστε όλες οι τιμές να βρίσκονται στο εύρος 0 μέχρι 1 και να είναι σε θέση να γίνει η εκτίμηση των μικρότερων και μεγαλύτερων τιμών. Στην τυποποίηση των χαρακτηριστικών ενός συνόλου δεδομένων περιλαμβάνεται η διαδικασία διαγραφής της κατανομής των πεδίων που αποτελούνται από τιμές έχοντας ως σκοπό ο μέσος όρος των τιμών να είναι 0 και η τιμή της τυπικής απόκλισης ίση με 1 δηλαδή να κεντράρονται τα δεδομένα ή να καταργείται η μέση τιμή. Η τυποποίηση των δεδομένων προϋποθέτει την εκτίμηση της ακρίβειας στην τυπική και

μέση απόκλιση των τιμών (Brownlee, 2020β΄).

3.4 Επιλογή των κατάλληλων χαρακτηριστικών

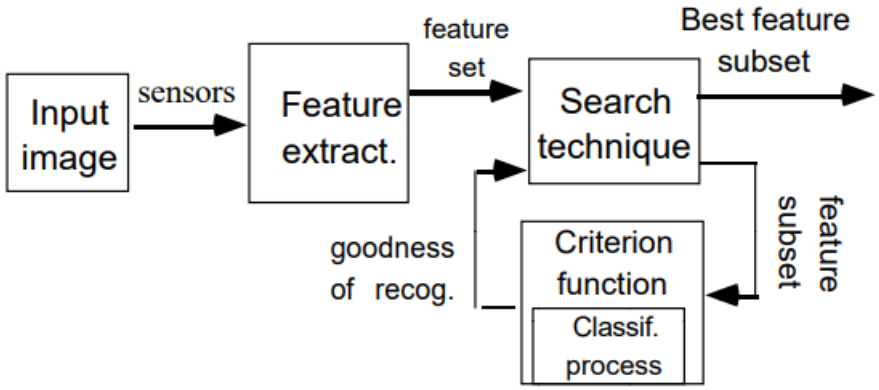
Αν κάθε χαρακτηριστικό χρησιμοποιείται ως ένα μέρος μιας κατηγοριοποίησης τότε ενδέχεται να αυξήσει το χρόνο και το κόστος που απαιτείται για τη λειτουργία του συστήματος αναγνώρισης. Την ίδια χρονική στιγμή υπάρχει και η ανάγκη να περιλαμβάνεται ένα ικανοποιητικό σύνολο από χαρακτηριστικά προκειμένου να επιτευχθεί ένα υψηλό ποσοστό αναγνώρισης. Με αυτόν τον τρόπο άρχισε η ανάπτυξη ενός συνόλου από τεχνικές, οι οποίες θα βρίσκουν το υψηλότερο υποσύνολο από χαρακτηριστικά. Η πρώτη στρατηγική επιλογής χαρακτηριστικών επιλέγει τα χαρακτηριστικά ανεξάρτητα από την απόδοση της κατηγοριοποίησης. Η δυσκολία που υπάρχει σε αυτήν τη στρατηγική αποτελεί ο εντοπισμός ενός κατάλληλου συνόλου από μετασχηματισμούς έτσι ώστε το μικρότερο μέρος από τα χαρακτηριστικά να περιέχει τις περισσότερες πληροφορίες τα οποία λαμβάνονται από τα αρχικά δεδομένα διότι είναι πιο αξιόπιστα αφού πρώτα έγινε η αφαίρεση χαρακτηριστικών που δεν είναι αναγκαία. Στη δεύτερη στρατηγική γίνεται η επιλογή χαρακτηριστικών από ένα υποσύνολο X που περιέχει δυνατότητες Y με τρόπο, ο οποίος δε θα δημιουργεί πρόβλημα στην απόδοση του συστήματος κατά τη διαδικασία της κατηγοριοποίησης. Το κύριο πρόβλημα της στρατηγικής αποτελεί η σύνδεση μεταξύ των χαρακτηριστικών όταν θα πρέπει να ταξινομηθούν και όταν γίνει η επιλογή ενός αποτελεσματικού υποσυνόλου. Η αρχιτεκτονική επιλογής χαρακτηριστικών δέχεται ως είσοδο ένα εκπαιδευτικό σύνολο το οποίο περιέχει αρνητικά και θετικά παραδείγματα από διάφορες τάξεις, οι οποίες στην πορεία θα κατηγοριοποιηθούν. Η απόδοση από κάθε ένα από τα υποσύνολα χαρακτηριστικών μπορεί να εκτιμηθεί από τη συνάρτηση αξιολόγησης. Στη συνέχεια, το υψηλότερο υποσύνολο χαρακτηριστικών θα χρησιμοποιηθεί στη σχεδίαση του συστήματος αναγνώρισης (Vafaie & De Jong, 1992).

3.4.1 Μείωση διαστάσεων

Στο Σχήμα 3.3 απεικονίζεται η αρχιτεκτονική επιλογής χαρακτηριστικών ενός συνόλου δεδομένων. Η μείωση διαστάσεων αποτελεί μια αρκετά γνωστή μέθοδο προεπεξεργασίας στην ανάλυση διαστάσεων τόσο στη μοντελοποίηση όσο και στην

οπτικοποίηση. Πλεονεκτήματα της μεθόδου είναι η μείωση διαστάσεων στον χώρο των χαρακτηριστικών προκειμένου να μειωθεί η απαίτηση για αποθηκευτικούς πόρους, η αφαίρεση άσχετων και περιττών δεδομένων, η επιτάχυνση του χρόνου εκτέλεσης και μάθησης του αλγορίθμου, η βελτίωση της ποιότητας των δεδομένων, η ακρίβεια που αποκτά το μοντέλο γίνεται υψηλότερη, η μείωση του συνόλου που περιλαμβάνει διάφορες λειτουργίες και τέλος, η δυνατότητα κατανόησης των δεδομένων για την απόκτηση γνώσεων που αφορούν διαδικασίες που οπτικοποιούν ή δημιουργούν δεδομένα. Υπάρχουν δύο τεχνικές μείωσης διαστάσεων, οι οποίες είναι η επιλογή χαρακτηριστικών και η μείωση διαστατικότητας. Το πλεονέκτημα της μεθόδου επιλογής χαρακτηριστικών είναι πως δε θα περιλαμβάνονται πληροφορίες που αφορούν ένα χαρακτηριστικό που έχει χαθεί, αλλά στην περίπτωση που είναι απαραίτητο να υπάρχει ένα μικρό σύνολο από χαρακτηριστικά τα οποία έχουν διαφορές από τα αυθεντικά χαρακτηριστικά τότε οι πληροφορίες ενδέχεται να χαθούν και να παραληφθούν κάποια από τα χαρακτηριστικά όταν ξεκινήσει η διαδικασία της επιλογής. Αντίθετα στη μέθοδο εξαγωγής το πεδίο των χαρακτηριστικών ενδέχεται κάποιες φορές να μειώνεται χωρίς να υπάρχει μεγάλη απώλεια πληροφοριών για τον αρχικό χώρο που καταλαμβάνουν τα χαρακτηριστικά (Khalid et al., 2014).

Σχήμα 3.3: Αρχιτεκτονική επιλογής χαρακτηριστικών (Khalid et al., 2014)



3.4.2 Επιλογή χαρακτηριστικών μέσω γενετικών αλγορίθμων

Οι γενετικοί αλγόριθμοι είναι γνωστοί για την ικανότητα τους να αναζητούν με αποτελεσματικό τρόπο μεγάλα σύνολα και χώρους δεδομένων. Μειονέκτημα τους αποτελεί η ευαισθησία που παρουσιάζουν σε δεδομένα με θόρυβο αλλά αποτελούν αρκετά καλή στρατηγική για επιλογή χαρακτηριστικών προκειμένου να βελτιωθεί η απόδοση συστήματος κατηγοριοποίησης. Επίσης οι γενετικοί αλγόριθμοι αποτελούν μορφή επαγωγικής μάθησης, η οποία παρουσιάζει σημαντική βελτίωση στη διαδικασία αναζήτησης. Αυτό επιτεύχθηκε από την ικανότητα τους να διαχειρίζονται πληροφορίες που είναι συσσωρευμένες σε έναν χώρο ανάζητησης όπου στην αρχή είναι άγνωστος. Ζητήματα που υπάρχουν στην εφαρμογή ενός γενετικού αλγορίθμου είναι η επιλογή της καταλληλότερης λειτουργίας για την αξιολόγηση και για την αναπαράσταση. Η πιο απλή μορφή αναπαράστασης είναι η δυαδική αναπαράσταση στην οποία κάθε δυνατότητα ανάμεσα στο σύνολο των χαρακτηριστικών μπορεί να θεωρηθεί ως δυαδικό γονίδιο όπου κάθε άτομο του αποτελεί μέρος του δυαδικού γονιδίου που έχει αμετάβλητο μήκος (Vafaie & De Jong, 1992).

Κεφάλαιο 4

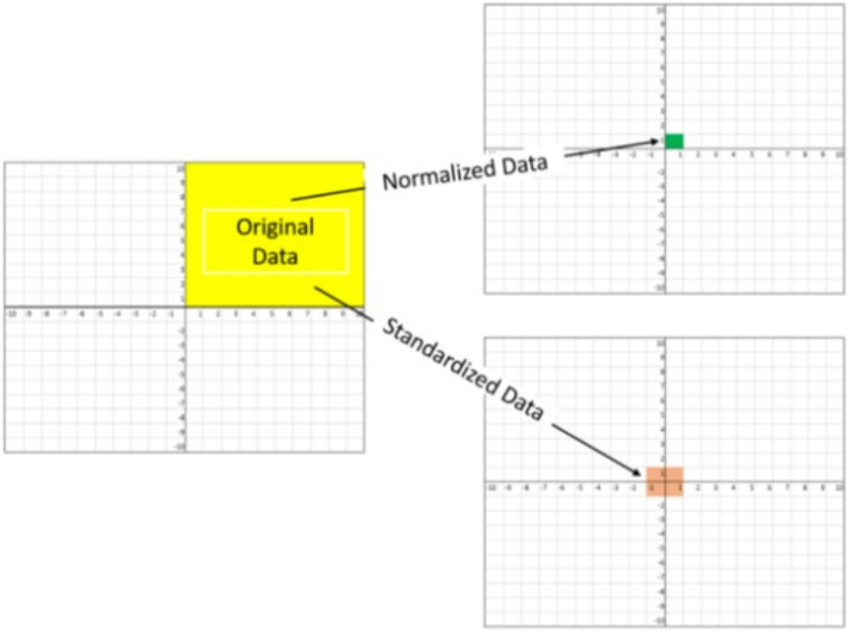
Αλγόριθμοι Κλιμάκωσης και Αντιμετώπισης Ελλιπών Τιμών

Σκοπός του κεφαλαίου αυτού είναι να αναλύσει τους αλγόριθμους κλιμάκωσης και ελλιπών τιμών που χρησιμοποιήθηκαν κατά την πειραματική διαδικασία.

4.1 Αλγόριθμοι κλιμάκωσης

Η κλιμάκωση των χαρακτηριστικών ενός συνόλου δεδομένων είναι σημαντικό βήμα στην προεπεξεργασία των δεδομένων. Μέσω αυτής μπορούν να ευνοηθούν αλγόριθμοι μηχανικής μάθησης αυξάνοντας τις επιδόσεις τους σε σημαντικό βαθμό. Δύο τεχνικές κλιμάκωσης στοιχείων αποτελούν η ομαλοποίηση και η κανονικοποίηση. Η μέθοδος της κανονικοποίησης μορφοποιεί τα χαρακτηριστικά στην κλίμακα 0 ως 1 ή -1 ως 1 ενώ η ομαλοποίηση μετατρέπει τα δεδομένα έτσι ώστε να έχουν τιμή διακύμανσης ίση με τον αριθμό 1 και μηδενικό μέσο όρο (Zheng & Casari, 2018). Στο Σχήμα 4.1 αναπαριστάται η μορφή των δεδομένων μετά την κλιμάκωση τους.

Σχήμα 4.1: Απεικόνιση των δεδομένων μετά την κλιμάκωση (Zheng & Casari, 2018)



Μερικοί αλγόριθμοι κλιμάκωσης δεδομένων είναι οι εξής:

1. Αλγόριθμος StandardScaler: Κανονικοποιεί τα χαρακτηριστικά με βάση τον μέσο όρο του συνόλου δεδομένων. Το σκορ που προκύπτει από ένα δείγμα x μπορεί να υπολογιστεί από τη συνάρτηση $z = (x - u)/s$. Το s συμβολίζει την τυπική απόκλιση ανάμεσα στα στοιχεία που προορίζονται για εκπαίδευση και το u συμβολίζει τον μέσο όρο του πλήθους των στοιχείων με τα οποία θα γίνει η εκπαίδευση. Η κλιμάκωση πραγματοποιείται ανεξάρτητα για το κάθε ένα από τα χαρακτηριστικά. Ύστερα θα αποθηκευτούν αφού πρώτα τροποποιηθούν η τυπική και η μέση απόκλιση.

2. Αλγόριθμος RobustScaler: Η κλιμάκωση των χαρακτηριστικών πραγματοποιείται σε κάθε στοιχείο ατομικά μέσω δειγμάτων που παρουσιάζουν μεγάλη ”δύναμη” σε σημεία τα οποία είναι ακραία. Ο RobustScaler αποσπά την ενδιάμεση τιμή (median), κλιμακώνοντας το σύνολο δεδομένων σύμφωνα με το εύρος ποσοτήτων το οποίο δείχνει σε ποιο τεταρτημόριο θα εφαρμοστεί η διαδικασία. Παραδείγματος χάρη για να εφαρμοστεί στο 2ο τεταρτημόριο απαιτείται η τιμή 50 ενώ για το 3ο τεταρτημόριο θα πρέπει να δοθεί η τιμή 75.

3. Αλγόριθμος QuantileTransformer: Μετασχηματίζει τα στοιχεία του συνόλου δεδομένων ατομικά αφού πρώτα εκτιμηθεί η συνάρτηση της αθροιστικής κα-

τανομής προκειμένου να μπορέσει να εφαρμοστεί μια κατανομή που θα είναι ομοιόμορφη. Οι τιμές που δίνονται στην έξοδο τοποθετούνται στην κατανομή εξόδου μέσω της συνάρτησης που είναι ποσοτική. Τέλος, με τον αλγόριθμο αυτό πραγματοποιείται η ελαχιστοποίηση του αντίκτυπου των οριακών τιμών.

4. Αλγόριθμος PowerTransformer: Πραγματοποιείται η τροποποίηση της ισχύος με σκοπό τα δεδομένα να έχουν πλέον γκαουσιανή μορφή. Ο αλγόριθμος αυτός είναι καλό να χρησιμοποιείται όταν απαιτείται να γίνει η μοντελοποίηση προβλημάτων που έχουν άμεση σχέση με διακυμάνσεις που δεν είναι σταθερές. Οι μετασχηματισμοί που μπορούν να επιλεχθούν είναι ο Yeo-Johnson (Weisberg, 2001), ο οποίος εφαρμόζεται και σε αρνητικούς αλλά και σε θετικούς αριθμούς και ο Box-Cox που απαιτεί τα στοιχεία να έχουν μόνο θετικό πρόσημο.

5. Αλγόριθμος MinMaxScaler: Τροποποιεί τα χαρακτηριστικά κλιμακώνοντας κάθε στοιχείο ατομικά σε ένα εύρος τιμών που είναι συγκεριμένο για κάποιο εκπαιδευτικό σύνολο. Εφαρμόζεται αρκετές φορές ως διαφορετική επίλυση με μέσο όρο που αφορά την κλιμάκωση διακύμανσης της μονάδας.

6. Αλγόριθμος Normalizer: Η ομαλοποίηση των στοιχείων πραγματοποιείται ατομικά. Κάθε ένα από τα στοιχεία που περιέχει τουλάχιστον ένα δείγμα το οποίο δεν είναι μηδενικό τοποθετείται ξανά στο σύνολο δεδομένων χωρίς να έχει πλέον κάποια σχέση με τα υπόλοιπα στοιχεία έτσι ώστε ο κανόνας l1 με l2 να είναι ίσος με τον αριθμό ένα. Επιπλέον ο αλγόριθμος Normalizer μπορεί να χρησιμοποιηθεί τόσο με τύπους πινάκων scipy.sparse matrix όσο με πίνακες της βιβλιοθήκης NumPy (*API Reference — scikit-learn 0.24.1 documentation*, 2021).

4.2 Αλγόριθμοι αντιμετώπισης ελλιπών τιμών

Αρκετά συχνά τα σύνολα δεδομένων περιέχουν στοιχεία με άγνωστες ή ελλιπείς τιμές, οι οποίες στις περισσότερες περιπτώσεις δεν μπορούν να αφαιρεθούν διότι διαδραματίζουν σημαντικό ρόλο στην εκπαίδευση ενός μοντέλου (Hauck, 2014). Στο Σχήμα 4.2 αποτυπώνεται ένα σύνολο δεδομένων με ελλιπείς τιμές, οι οποίες αναπαριστώνται από το nan.

Σχήμα 4.2: Σύνολο δεδομένων με ελλιπείς τιμές (Vikram, 2020)

Country	Age	Salary	Purchased
France	44	72000	No
Spain	27	48000	Yes
Germany	30	54000	No
Spain	38	61000	No
Germany	40	nan	Yes
France	35	58000	Yes
Spain	nan	52000	No
France	48	79000	Yes
Germany	50	83000	No
France	37	67000	Yes

Ο προσδιορισμός των χαρακτηριστικών μπορεί να πραγματοποιηθεί με έναν από τους παρακάτω αλγόριθμους αντιμετώπισης ελλιπών τιμών:

1. Αλγόριθμος SimpleImputer: Τροποποιεί τα πεδία που περιέχουν άγνωστες τιμές, χρησιμοποιώντας μια από τις τέσσερις στρατηγικές. Οι στρατηγικές αυτές μπορεί να είναι:

    - mean: Αντικαθιστά τα πεδία που είναι άγνωστα με τον μέσο όρο της κάθε μιας από τις στήλες
    - most_frequent: Αντικαθιστά τα πεδία που είναι άγνωστα με την τιμή που εμφανίζεται συχνότερα στην κάθε μια από τις στήλες
    - constant: Αντικαθιστά τις ελλιπείς τιμές με την τιμή που επιθυμεί ο χρήστης

- median: Αντικαθιστά τα πεδία που λείπουν με τη μέση τιμή της κάθε στήλης (*sklearn.impute.SimpleImputer — scikit-learn 0.24.1 documentation*, 2021)

2. Αλγόριθμος KNNImputer: Η τροποποίηση των άγνωστων τιμών στα σύνολα δεδομένων πραγματοποιείται με βάση την παράμετρο n_neighbours που αναπαριστά το πλήθος των πιο κοντινών γειτόνων στο εκπαιδευτικό σύνολο. Θα πρέπει να αναφερθεί πως ο KNNImputer δεν μπορεί να τροποποιήσει άγνωστες τιμές που είναι κατηγορηματικές, με αποτέλεσμα να υπάρξουν σφάλματα αν δε γίνει η κωδικοποίηση τους σε αριθμητικές. Τέλος, εφόσον ο αλγόριθμος υπολογίζει την απόσταση μεταξύ δύο γειτόνων θα πρέπει να γίνει η ομαλοποίηση των δεδομένων διότι αν υπάρχουν διαφοροποιήσεις στις κλίμακες της κάθε στήλης τότε η αντικατάσταση των άγνωστων τιμών δε θα γίνει με σωστό τρόπο (Htoon, 2020).

Κεφάλαιο 5

Υπολογιστικά αποτελέσματα

5.1 Σύντομη περιγραφή του πειράματος

Ο κύριος σκοπός της πειραματικής διαδικασίας είναι η εύρεση του υψηλότερου ποσοστού ακρίβειας ενός ή περισσοτέρων συνόλων δεδομένων που δίνονται ως είσοδοι στον κώδικα από ένα πλήθος αλγορίθμων, οι οποίοι θα συνδυάζονται με έναν αλγόριθμο αντιμετώπισης ελλιπών τιμών και με έναν αλγόριθμο κλιμάκωσης αφού πρώτα έχει προηγηθεί η κατάλληλη προεπεξεργασία δεδομένων. Η πειραματική διαδικασία ξεκινάει τροποποιώντας τα πεδία που περιλαμβάνουν άγνωστες τιμές με τη βοήθεια διαφορετικών αλγορίθμων αντιμετώπισης ελλιπών τιμών και ύστερα θα γίνεται η κωδικοποίηση κατηγορηματικών χαρακτηριστικών σε αριθμητικά. Οι αλγόριθμοι που χρησιμοποιήθηκαν προέρχονται από τη βιβλιοθήκη scikit-learn. Πιο συγκεκριμένα για την κατηγοριοποίηση χρησιμοποιήθηκαν οι αλγόριθμοι υποστήριξης διανυσμάτων μηχανών (SVM), κατηγοριοποιητής δέντρων απόφασης (DecisionTreeClassifier), κατηγοριοποιητής MLP (MLPClassifier), κατηγοριοποιητής πλησιέστερων γειτώνων (KNeighborsClassifier) και κατηγοριοποιητής τυχαίων δασών (RandomForestClassifier). Οι αλγόριθμοι που χρησιμοποιήθηκαν για παλινδρόμηση είναι ο αλγόριθμος υποστήριξης διανυσμάτων μηχανών (SVR), γραμμικής παλινδρόμησης (LinearRegression), παλινδρόμησης κορυφογραμμών (Ridge), παλινδρόμησης Lasso (Lasso), παλινδρόμησης δέντρων απόφασης (DecisionTreeRegressor), παλινδρόμησης τυχαίων δασών (RandomForestRegressor), παλινδρόμησης MLP (MLPRegressor) καθώς επίσης και παλινδρόμησης πλησιέστερων γειτόνων (KNeighborsRegressor). Με τη χρήση της μεθόδου Κ-πτυχών διασταυρούμενης επικύρωσης (K-Fold Cross Validation) μπορεί να πραγματοποιηθεί ο διαχωρισμός των συνόλων δεδομένων σε

εκπαιδευτικά και δοκιμαστικά σύνολα.

5.2 Η συλλογή των συνόλων δεδομένων

Όλα τα σύνολα δεδομένων που χρησιμοποιήθηκαν τόσο για τα πειράματα κατηγοριοποίησης όσο για τα πειράματα παλινδρόμησης προέρχονται από την ιστοσελίδα UC Irvine Machine Learning Repository (*UCI Machine Learning Repository*, 2021) και από την ιστοσελίδα Kaggle (*Kaggle: Your Home for Data Science*, 2021).

1. Σύνολα δεδομένων που χρησιμοποιήθηκαν για πειράματα κατηγοριοποίησης

    - abalone (*UCI Machine Learning Repository: Abalone Data Set*, 2021): Σε αυτό το σύνολο δεδομένων γίνεται η πρόβλεψη της ηλικίας του όστρακου. Η ηλικία του όστρακου μπορεί να καθοριστεί από το κόψιμο του κελύφους και του πλήθους των δακτυλίων στο μικροσκόπιο. Τα χαρακτηριστικά του είναι τα: Sex (φύλο του όστρακου), Length (μήκος του όστρακου), Diameter (δίαμετρος του όστρακου), Height (ύψος του όστρακου), Whole_weight (συνολικό βάρος του όστρακου), Shucked_weight (βάρος όστρακου αφού αφαιρέθηκε το κέλυφος του), Viscera_weight (βάρος σπλάχνων του όστρακου) και Shell_weight (βάρος κέλυφους του όστρακου).
    - balanceScale (*UCI Machine Learning Repository: Balance Scale Data Set*, 2021): Το σύνολο δεδομένων αυτό κατασκευάστηκε προκειμένου να μοντελοποιήσει ψυχολογικά πειραματικά αποτελέσματα. Κάθε ένα από τα παραδείγματα ταξινομείται με βάση το άκρο της κλίμακας ισορροπίας προς τα δεξιά, το άκρο προς τα αριστερά ή τα δύο άκρα να βρίσκονται σε ισορροπία. Τα χαρακτηριστικά είναι το αριστερό βάρος, η αριστερή απόσταση, το δεξί βάρος και η δεξιά απόσταση.
    - bloodTransfusion (*UCI Machine Learning Repository: Blood Transfusion Service Center Data Set*, 2021): Κάθε ένα από τα 748 δεδομένα περιλαμβάνει ως χαρακτηριστικά το R (Μήνες από την τελευταία δωρεά), F (Συνολικός αριθμός από δωρεές), M (Συνολικό αίμα που δωρίστηκε σε cc), T (Μήνες που πέρασαν από την τελευταία δωρεά) και τέλος, μια δυαδική μετα-

βλητή που δείχνει εάν δωρίζεται αίμα από το Μάρτιο του 2007 (το 0 συμβολίζει τη μη δωρεά αίματος και το 1 τη δωρεά αίματος).

- cars (*UCI Machine Learning Repository: Car Evaluation Data Set*, 2021): Στο σύνολο δεδομένων αυτό πραγματοποιείται η αξιολόγηση των αυτοκινήτων σύμφωνα με τα εξής χαρακτηριστικά: buying (τιμή αγοράς), maint (το μέγεθος διατήρησης της τιμής), doors (ο αριθμός θυρών), people (αριθμός ατόμων που μπορούν να επιβιβαστούν σε αυτό), ugboot (μέγεθος του χώρου αποσκευών), safety (εκτιμώμενη ασφάλεια του αυτοκινήτου) και acceptability (αποδοχή του αυτοκινήτου από τους ενδιαφερόμενους)
- bestSellersBooks (Saalu, 2021): Τα χαρακτηριστικά του συνόλου δεδομένων αυτού είναι το Name (όνομα του βιβλίου), Author (όνομα συγγραφέα του βιβλίου), User Rating (η βαθμολογία του βιβλίου σύμφωνα με τις αξιολογήσεις των χρηστών), Reviews (το συνολικό πλήθος των κριτικών που έγιναν για το συγκεκριμένο βιβλίο), Price (η τιμή του βιβλίου), Year (το έτος έκδοσης του βιβλίου) και τέλος, το χαρακτηριστικό Genre (συμβολίζει το είδος του βιβλίου δηλαδή αν είναι φαντασίας ή όχι).
- haberman (*UCI Machine Learning Repository: Haberman's Survival Data Set*, 2021): Το σύνολο δεδομένων περιέχει υποθέσεις από μια μελέτη που πραγματοποιήθηκε από το 1958 μέχρι το 1970 και αφορά την επιβίωση των ασθενών που είχαν υποβληθεί σε χειρουργική επέμβαση για τον καρκίνο του μαστού. Τα χαρακτηριστικά του συνόλου δεδομένων ήταν η στήλη age (συμβολίζει την ηλικία του ασθενός), year (έτος που ο ασθενής υποβλήθηκε σε χειρουργική επέμβαση), nodes (αριθμός των κόμβων που ανιχνεύτηκαν) και τέλος, το Survival status (όταν η τιμή κάποιου πεδίου της στήλης είναι ίση με 1 τότε σημαίνει πως ο ασθενής επέζησε 5 ή περισσότερα χρόνια και 2 όταν ο ασθενής πέθανε το πολύ σε 5 χρόνια).
- hayesRoth (*UCI Machine Learning Repository: Hayes-Roth Data Set*, 2021): Τα χαρακτηριστικά του συνόλου δεδομένων αυτού είναι το name (όνομα διακριτό για κάθε μια από παρουσίες του συνόλου δεδομένων και αντιπροσωπεύεται αριθμητικά), hobby (χόμπυ, τιμές που είναι ανάμεσα στο 1 και στο 3), age (ηλικία, τιμές που κυμαίνονται ανάμεσα στο 1 και στο 4), marital status (οικογενειακή κατάσταση, ονομαστικές τιμές που είναι

ανάμεσα στον αριθμό 1 εώς 4), educational level (εκπαιδευτικό επίπεδο, τιμές μεταξύ του 1 και του 4), class (κλάση, ονομαστική τιμή από 1 εώς 3).

- lenses (*UCI Machine Learning Repository: Lenses Data Set*, 2021): Σε αυτό το σύνολο δεδομένων περιλαμβάνονται τα εξής χαρακτηριστικά: age of the patient (ηλικία του ασθενή όπου με 1 συμβολίζεται νεαρός, 2 προ-πρεσβυωπικός, 3 πρεσβυωπικός), spectacle prescription (ιατρική συνταγή όπου με 1 συμβολίζεται μυωπία, 2 η υπερμετρία), astigmatic (αστιγματικός όπου με 1 συμβολίζεται πως δεν είναι και 2 πως είναι), tear production rate (ρυθμός παραγωγής δακρύων όπου με 1 συμβολίζεται μειωμένος και 2 ο κανονικός) και τέλος, class (κλάση στην οποία ανήκει το κάθε στοχείο με τιμές μεταξύ του 1 και του 3).
- drugTypes (Hirethanad, 2021): Τα χαρακτηριστικά του συνόλου δεδομένων αυτού είναι το Age (ηλικία του ατόμου), Sex (συμβολίζει το φύλο του ατόμου), BP (δείχνει το επίπεδο που βρίσκεται η αρτηριακή πίεση του ατόμου), Cholesterol (συμβολίζει τα επίπεδα χοληστερίνης που έχει ένα άτομο), Na_to_K (αναλογία του νάτριου ως προς το κάλιο) και τέλος, το χαρακτηριστικό Drug (τύπος του φάρμακου που μπορεί να λαμβάνει ο ασθενής).
- ticTacToe (*UCI Machine Learning Repository: Tic-Tac-Toe Endgame Data Set*, 2021): Το σύνολο δεδομένων ticTacToe κωδικοποιεί το πλήρες σύνολο πιθανών διαμορφώσεων πινάκων στο τέλος κάθε παρτίδας του παιχνιδιού τρίλιζας. Τα χαρακτηριστικά της βάσης δεδομένων αυτής είναι top-left-square, top-middle-square, top-right-square, middle-left-square, middle-middle-square, middle-right-square, bottom-left-square, bottom-middle-square, bottom-right-square τα οποία συμβολίζουν τις τιμές που περιέχει κάθε ένα από τα 9 κουτάκια της τρίλιζας και παίρνουν τις τιμές x,o,b. Επιπλέον υπάρχει και ένα ακόμη χαρακτηριστικό με το όνομα class (κλάση που λαμβάνει τις τιμές positive και negative).

2. Σύνολα δεδομένων που χρησιμοποιήθηκαν για πειράματα παλινδρόμησης

- aquaticToxicity (*UCI Machine Learning Repository: QSAR aquatic toxicity Data*

Set, 2021): Η χρήση του συνόλου δεδομένων αυτού είναι η ανάπτυξη μοντέλων ποσοτικής παλινδρόμησης QSAR προκειμένου να γίνει η πρόβλεψη οξείας υδάτινης τοξικότητας προς τα ψάρια τύπου Pimephales promelas σε ένα σύνολο 908 χημικών ουσιών. Τα χαρακτηριστικά της βάσης δεδομένων αυτής είναι τα TPSA, SAacc, MLOGP τα οποία αποτελούν μοριακές ιδιότητες, το H-050 και το C-040 που συμβολίζουν τα θραύσματα, RDCHI το οποίο αναπαριστά δείκτες συνδεσιμότητας, GATS1p που δείχνει την 2D αυτοσυσχέτιση και το nN που συμβολίζει τους συνταγματικούς δείκτες.

- carPriceAssignment (Goyal, 2021): Τα χαρακτηριστικά του συνόλου δεδομένων αυτού είναι το carID (αριθμός ID του αυτοκινήτου), symboling (συμβολισμός αυτοκινήτου), CarName (όνομα αυτοκινήτου), fuelType (τύπος καυσίμου με το οποίο λειτουργεί το αυτοκίνητο), aspiration (φιλοδοξία του αυτοκινήτου), doornumber (αριθμός θυρών του αυτοκινήτου), carbody (τύπος αυτοκινήτου), drivewheel (τροχός κίνησης), enginelocation (θέση τοποθέτησης του κινητήρα), wheelbase (μεταξόνιο), carlength (μήκος τους αυτοκινήτου), carwidth (πλάτος του αυτοκινήτου), carheight (ύψος του αυτοκινήτου), curbweight (βάρος αυτοκινήτου), enginetype (τύπος μηχανής), cylindernumber (αριθμός κυλίνδρων μηχανής), enginesize (μέγεθος μηχανής), fuelsystem (τύπος συστήματος καυσίμων), boreration (αναλογία οπών), stroke (εγκέφαλος αυτοκινήτου), compressionratio (αναλογία συμπίεσης), horsepower (ιπποδύναμη αυτοκινήτου), peakrpm (μέγιστος αριθμός στροφών μηχανής), citympg (κατανάλωση εντός πόλης), highwaympg (κατανάλωση εκτός πόλης) και τέλος price (τιμή του αυτοκινήτου).
- diabetes (*UCI Machine Learning Repository: Diabetes Data Set*, 2021): Το σύνολο δεδομένων αυτό περιέχει τα εξής χαρακτηριστικά: Pregnancies (αριθμός εγκυμοσυνών), Glucose (επίπεδα γλυκόζης), BloodPressure (πίεση αίματος), SkinThickness (πάχος του δέρματος), Insulin (επίπεδα ινσουλίνης), BMI (δείκτης BMI), DiabetesPedigreFunction (λειτουργία γενεαλογικού διαβήτη), age (ηλικία) και το Outcome (αποτέλεσμα).
- drivePoints (*UCI Machine Learning Repository: DrivFace Data Set*, 2021): Τα

χαρακτηριστικά του συνόλου είναι το fileName (όνομα αρχείου), subject (θέμα), imgNum (αριθμός της εικόνας), label (επιγραφή) καθώς επίσης και διάφορες μετρικές ang, xF, yF, wF, hF, xRE, yRE, xLE, yLE, xN, yN, xRM, yRM, xLM, yLM.

- advertising (Ashish, 2021): Το σύνολο δεδομένων αυτό περιλαμβάνει τα εξής χαρακτηριστικά: TV (πλήθος των προβολών της διαφήμισης στην τελεόραση), Radio (αριθμός συνολικών προβολών της διαφήμισης στο ραδιόφωνο), Newspaper (αριθμός εφημερίδων στις οποίες τοποθετήθηκε η συγκεκριμένη διαφήμιση) και Sales (αριθμός συνολικών πωλήσεων που πραγματοποιήθηκαν λόγω της διαφήμισης).
- insurance (Choi, 2021): Τα χαρακτηριστικά του συνόλου δεδομένων αυτού είναι το age (ηλικία του ατόμου), sex (το φύλο του ατόμου), bmi (δείκτης μέτρησης μάζας σώματος), children (αριθμός παιδιών), smoker (χαρακτηριστικό που δείχνει αν το άτομο είναι καπνιστής και λαμβάνει τις τιμές yes ή no), region (περιοχή που κατοικεί το άτομο) και τέλος το charges (συμβολίζει το μισθό το ατόμου).
- realEstate (Bruce, 2021): Το σύνολο δεδομένων αυτό περιλαμβάνει τα εξής χαρακτηριστικά: No (αριθμός της κατοικίας / διαμερίσματος), X1 transaction date (ημερομηνία συναλλαγής με συμβολισμό X1), X2 house age (ηλικία του σπιτιού με συμβολισμό X2), X3 distance to the nearest MRT station (απόσταση από το πλησιέστερο σταθμό MRT με συμβολισμό X3), X4 number of convenience stores (αριθμός καταστημάτων με συμβολισμό X4), X5 latitude (γεωγραφικό πλάτος με συμβολισμό X5), X6 longitude (γεωγραφικό μήκος με συμβολισμό X6), Y house price of unit area (τιμή σπιτιού με συμβολισμό Y).
- slumpTest (*UCI Machine Learning Repository: Concrete Slump Test Data Set*, 2021): Η βάση δεδομένων αυτή περιέχει 103 γραμμές δειγμάτων. Τα χαρακτηριστικά του συνόλου δεδομένων είναι τα εξής: No (αριθμός της γραμμής του συνόλου δεδομένων), Cement (ποσότητα τσιμέντου), Slag (ποσότητα σκουριάς), Fly ash (ποσότητα ιπτάμενης τέφρας), Water (ποσότητα νερού), SP, Coarse Aggr (ποσότητα τραχιού πρόσμιγματος), Fine Aggr (ποσότητα ομαλού προσμίγματος), SLUMP (cm) (εκατοστά καθίζη-

σης τσιμέντου), FLOW (cm) (εκατοστά ροής τσιμέντου) και τέλος 28-day Compressive Strength (Mpa) (ποσοστό αντοχής τσιμέντου μέσα στις 28 ημέρες).

- southGermanCredit (*UCI Machine Learning Repository: South German Credit Data Set*, 2021): Σε αυτήν τη βάση δεδομένων ταξινομούνται άτομα τα οποία μπορούν να περιγραφούν από ένα σύνολο χαρακτηριστικών ως κακοί ή καλοί πιστωτικοί κίνδυνοι. Το σύνολο των χαρακτηριστικών αυτών είναι: laufkont (κατάσταση), laufzeit (διάρκεια), moral (αριθμός πιστωτικών συναλλαγών), verw (σκόπος των συναλλαγών), hoehe (ποσό των συναλλαγών), sparkont (οικονομίες), beszeit (συνολικός χρόνος απασχόλησης), rate (ποσοστό δόσεων στις συναλλαγές), famges (φύλο και οικογενειακή κατάσταση του ατόμου), buerge (άλλοι οφειλέτες στις συναλλαγές), wohnzeit (χρόνος διαμονής στην παρούσα κατοικία), verm (ιδιοκτησία), alter (ηλικία του ατόμου), weitkred (άλλα προγράμματα δόσεων), wohn (τρόπος διαμονής σε κατοικία), bishkred (συνολικός αριθμός πιστώσεων), beruf (επάγγελμα του ατόμου), pers (αριθμός ατόμων που ευθύνονται για τις πιστώσεις), telef (στοιχείο που αναφέρει εαν συμπεριλαμβάνεται αριθμός τηλεφώνου ή όχι), gastarb (στοιχείο που δείχνει αν το άτομο που εργάζεται είναι αλλοδαπός ή οχι) και τέλος, το χαρακτηριστικό kredit (είναι το χαρακτηριστικό με το οποίο θα γίνει η κατηγοριοποίηση και δείχνει αν υπάρχει πιστωτικός κίνδυνος ή όχι).
- winequalityRed (*UCI Machine Learning Repository: Wine Quality Data Set*, 2021): Το σύνολο δεδομένων αυτό αφορά τις κόκκινες παραλλαγές του κρασιού από την Πορτογαλία με την ονομασία Vinho Verde. Το σύνολο δεδομένων περιλαμβάνει τα εξής χαρακτηριστικά: fixed acidity (συμβολίζει τη σταθερή οξύτητα ενός κρασιού), volatile acidity (συμβολίζει την πτητική οξύτητα ενός κρασιού), citric acid (συμβολίζει το κιτρικό οξύ ενός κρασιού), residual sugar (υπολειμματική ζάχαρη σε ένα κρασί), chlorides (συμβολίζει τα χλωρίδια που περιέχει το κάθε ένα κρασί), free sulfur dioxide (συμβολίζεται η ποσότητα ελεύθερου διοξειδίου του θείου σε ένα κρασί), total sulfur dioxide (συμβολίζεται η ποσότητα ολικό διοξείδιο του θείου σε κάθε ένα από τα κρασιά), density (συμβολίζει την πυκνότητα

ενός κρασιού), pH (συμβολίζει τον δείκτη pH ενός κρασιού) και τέλος, το χαρακτηριστικό sulphates (συμβολίζει τον συνολικό αριθμό των θειικών που περιέχονται σε ένα κρασί).

5.3 Τα εργαλεία που χρησιμοποιήθηκαν για την ανάπτυξη του πειράματος

1. Η γλώσσα προγραμματισμού που χρησιμοποιήθηκε για την υλοποίηση της πειραματικής διαδικασίας είναι η Python, η οποία αποτελεί γλώσσα υψηλού επιπέδου. Μέσω των γλωσσικών υποδομών της και της αντικειμενοστρέφειας βοηθάει τους προγραμματιστές να γράφουν αρκετά πιο λογικό και αναγνώσιμο κώδικα σε μεγάλης ή μικρής δυσκολίας έργα. Με τη χρήση της μπορεί κάποιος να γράψει λειτουργικό (functional), δομημένο (structured) και αντικειμενοστραφή (object oriented) προγραμματισμό. Ο ιδρυτής της ήταν ο Guido van Rossum (Van Rossum, Drake, et al., 2000) και η Python ξεκίνησε να χρησιμοποιείται για πρώτη φορα το 1991 (Van Rossum et al., 2021). Τέλος, η Python αποτελεί την καλύτερη γλώσσα προγραμματισμού στο πεδίο της μηχανικής μάθησης και της τεχνητής νοημοσύνης διότι παρέχει ένα τεράστιο σύνολο από χρήσιμες και εύκολες σε χρήση βιβλιοθήκες για ανάπτυξη έργων που απαιτούν μηχανική μάθηση και γι' αυτό είναι τόσο δημοφιλής (Pedregosa et al., 2011).

2. Το ολοκληρωμένο περιβάλλον ανάπτυξης λογισμικού που χρησιμοποιήθηκε για την ανάπτυξη του πειράματος ήταν το Spyder. Το περιβάλλον αυτό αποτελεί περιβάλλον ανοιχτού και ελεύθερου κώδικα γραμμένο στη γλώσσα προγραμματισμού Python. Σχεδιάστηκε για να εξυπηρετήσει μηχανικούς, αναλυτές δεδομένων και επιστήμονες. Με τις λειτουργίες του η προεπεξεργασία δεδομένων πραγματοποιείται με ευκολότερο τρόπο. Περιλαμβάνει τον συντάκτη (editor) που χρησιμοποιείται για τη συγγραφή του κώδικα, την κονσόλα IPython, το παράθυρο εξερεύνησης μεταβλητών (variable explorer), τον εντοπισμό σφαλμάτων (debugger) και την απεικόνιση των γραφικών παραστάσεων (Team, 2021).

3. Βιβλιοθήκη NumPy: Είναι χρήσιμη όταν κάποιος επιθυμεί να δουλέψει με πίνακες, δηλαδή να πραγματοποιήσει μαθηματικές πράξεις σε αυτούς. Μερικά

πλεονεκτήματα της είναι (Idris, 2015):

- Οι πίνακες τύπου NumPy δεν είναι τόσο αργοί όσο οι λίστες της Python και απαιτούν λιγότερους πόρους μνήμης.
- Χρήση σε πίνακες πολλαπλών διαστάσεων
- Προσφέρει λειτουργίες όπως το reverse, reshape, sort.
- Μπορεί να εφαρμοστεί για να τροποποιήσει πίνακες

4. Βιβλιοθήκη Pandas: Άρχισε να αναπτύσσεται για πρώτη φορά το 2008. Μέσω των λειτουργιών της μπορεί κάποιος να αναλύσει, τροποποιήσει και καθαρίσει ένα σύνολο δεδομένων. Αρχικά, η βιβλιοθήκη pandas θα διαβάσει ένα CSV και θα το μετατρέψει σε πλαίσια δεδομένων (dataframes). Ύστερα θα μπορούν να απαντηθούν ερωτήματα για την εύρεση του μεγαλύτερου ή μικρότερου στοιχείου σε κάθε μια από τις στήλες του συνόλου δεδομένων και αν η στήλη Χ έχει κάποια σύνδεση με τη στήλη Υ. Ο καθαρισμός των πεδίων γίνεται στην περίπτωση που υπάρχουν κενά πεδία με άγνωστες τιμές. Μια ακόμη λειτουργία της είναι η αποθήκευση ενός συνόλου δεδομένων σε αρχείο τύπου CSV αφού πρώτα αυτό έχει υποστεί αλλαγές σε σημεία που απαιτούνταν. Θα πρέπει να σημειωθεί πως η βιβλιοθήκη Pandas είναι κατασκευασμένη με βάση τη βιβλιοθήκη NumPy. Τα πλαίσια δεδομένων και οι αλλαγές τους μπορούν να δοθούν ως είσοδοι στην απεικόνιση γραφικών παραστάσεων καθώς επίσης και σε βιβλιοθήκες που προσφέρουν τη δυνατότητα χρήσης αλγορίθμων μηχανικής μάθησης όπως είναι το Scikit-learn (VanderPlas, 2016).

5. Scikit-learn: Αποτελεί δωρεάν βιβλιοθήκη μηχανικής μάθησης φτιαγμένη με βάση τη γλώσσα προγραμματισμού Python. Περιέχει μια πληθώρα από αλγόριθμους παλινδρόμησης, κατηγοριοποίησης καθώς επίσης και ομαδοποίησης. Μπορεί να λειτουργήσει αποδοτικά σε συνδυασμό με μια πληθώρα από βιβλιοθήκες όπως είναι η matplotlib, η NumPy και η Pandas (Contributors to Wikimedia projects, 2021).

5.4 Αλγόριθμοι που χρησιμοποιήθηκαν στην κατηγοριοποίηση και στην παλινδρόμηση

5.4.1 Μηχανές διανυσμάτων υποστήριξης

Σκοπός του αλγορίθμου μέσω της αναζήτησης είναι η εύρεση υπερεπιπέδου (hyperplane) σε χώρο με διάσταση που συμβολίζει το πλήθος των χαρακτηριστικών του συνόλου δεδομένων. Ο διαχωρισμός των δειγμάτων σε δύο τμήματα μπορεί να πραγματοποιηθεί κάνοντας χρήση αρκετών διαφορετικών υπερεπιπέδων. Θα πρέπει όμως να βρεθεί το υπερεπίπεδο που περιλαμβάνει τη μεγαλύτερη δυνατή απόσταση ανάμεσα στις δύο κατηγορίες και στα δείγματα δεδομένων. Αν μεγαλώσει η απόσταση περιθωρίου (margin) τότε τα δείγματα δεδομένων θα έχουν τη δυνατότητα να ταξινομηθούν με περισσότερη ακρίβεια. Τα υπερεπιπέδα αποτελούν φάσματα αποφάσεων και μετατρέπουν την ταξινόμηση των δειγμάτων σε πιο εύκολη διαδικασία. Επιπλέον ο αριθμός των χαρακτηριστικών καθορίζει και τη διάσταση που θα έχει ένα υπερεπίπεδο. Τα δείγματα που βρίσκονται σε πιο κοντινή απόσταση με το υπερεπίπεδο ελέγχουν τον προσανατολισμό και την τοποθεσία του. Στην περίπτωση που δεν υπάρχουν ή έχουν αφαιρεθεί αλλάζει η θέση του υπερεπιπέδου. Όταν το αποτέλεσμα της γραμμικής συνάρτησης είναι -1 ανήκει σε διαφορετική κλάση ενώ όταν είναι μεγαλύτερο από την τιμή τότε μπορεί να ταυτιστεί με μια από τις κλάσεις (Wang, 2005).

5.4.2 Δέντρα απόφασης

Αναπαριστά δομή δέντρου, η οποία έχει αρκετές ομοιότητες με ένα διάγραμμα ροής. Το χαρακτηριστικό αντιπροσωπεύεται από έναν εσωτερικό κόμβο, ο κανόνας απόφασης από έναν κλάδο και το αποτέλεσμα από τον κόμβο που προκύπτει. Ο κόμβος που βρίσκεται στην κορυφή του δέντρου ονομάζεται ριζικός κόμβος. Εκπαιδεύεται με γρηγορότερο ρυθμό από τον αντίστοιχο αλγόριθμο στα νευρωνικά δίκτυα έχοντας πολύ καλό ποσοστό ακρίβειας ακομή και όταν τα δεδομένα είναι μεγάλων διαστάσεων. Η λειτουργία του αλγορίθμου μπορεί να περιγραφεί από τα ακόλουθα βήματα :

1. Επιλογή του καλύτερου χαρακτηριστικού.

2. Το χαρακτηριστικό που επιλέχθηκε ως βέλτιστο θα αποτελέσει έναν καινούργιο κόμβο απόφασης και στη συνέχεια θα γίνει ο διαχωρισμός του συνόλου δεδομένων σε μικρότερα υποσύνολα.

3. Αφού κατασκευαστεί το δέντρο, η εφαρμογή των παραπάνω βημάτων θα επαναλαμβάνεται μέχρι να ικανοποιηθεί μια από τις εξής απαιτήσεις:

   (α΄) Έχουν εξεταστεί όλες οι δυνατές περιπτώσεις

   (β΄) Έχουν εξεταστεί όλα τα χαρακτηριστικά του συνόλου δεδομένων

   (γ΄) Όλο το πλήθος των πλειάδων αποτελεί ιδιοκτησία μοναδικού χαρακτηριστικού

Τα χαρακτηριστικά επιλέγονται από μέτρο επιλογής ευρετικού τύπου. Η μέθοδος ASM δημιουργεί μια κατάταξη για όλο το πλήθος των χαρακτηριστικών. Στη συνέχεια, θα επιλεχθεί το χαρακτηριστικό που έχει το υψηλότερο σκορ προκειμένου να πραγματοποιηθεί ο διαχωρισμός με βάση αυτό. Στην περίπτωση που το χαρακτηριστικό είναι συνεχές θα πρέπει να οριστούν εξαρχής σημεία που θα γίνει ο διαχωρισμός (Navlani, 2021).

5.4.3 Πολυστρωματικό αντίληπτρο - νευρωνικά δίκτυα

Ο αλγόριθμος αυτός περιλαμβάνει υψηλό αριθμό στρωμάτων που είναι γραμμικά. Σε ένα δίκτυο που έχει τρία επίπεδα, το πρώτο επίπεδο θα είναι αυτό που θα λαμβάνει την είσοδο, το δεύτερο επίπεδο δε θα είναι γνωστό και στο τρίτο που είναι και το τελευταίο επίπεδο θα εξάγεται η έξοδος ως αποτέλεσμα. Ο αριθμός των κρυφών επιπέδων μπορεί να γίνει μεγαλύτερος αν το επιθυμεί ο χρήστης. Ένα δίκτυο που είναι δομημένο σε τρία επίπεδα μπορεί να εκφραστεί από τη συνάρτηση $f(x) = f(3)(f(2)(f(1)(x)))$ και κάθε στρώμα από τη σχέση $y = f(WxT + b)$. Το W αναπαριστά το συνολικό πλήθος των παραμέτρων,το b το διάνυσμα πόλωσης, το x αναπαριστά το διάνυσμα που δίνεται στην είσοδο και το f τη συνάρτηση που θα περιγράφει τις σχέσεις μεταξύ εισόδων και εξόδων. Ο αλγόριθμος αποτελείται από αρκετά επίπεδα που συνδέονται μεταξύ τους. Σε εποπτευόμενα συστήματα κατηγοριοποίησης κάθε στοιχείο που δίνεται στην είσοδο αντιστοιχίζεται σε μια ετικέτα που αναπαριστά και μια κλάση. Στην έξοδο εξάγεται από το σύστημα ένα σκορ για την κάθε κλάση. Η μέτρηση απόδοσης ενός κατηγοριοποιητή μπορεί να εκτιμηθεί

μέσω της συνάρτησης απωλειών. Τα τρία στάδια εφαρμογής του αλγορίθμου είναι τα εξής :

1. Κίνηση προς τα εμπρός: Τροφοδοτεί στο μοντέλο την είσοδο πολλαπλασιάζοντας τα βάρη και προσθέτοντας τη μεροληψία. Επιπλέον μέσω αυτού του σταδίου εξάγεται και η εξόδος που υπολογίστηκε.

2. Υπολογισμός απωλειών: Το στάδιο αυτό είναι υπεύθυνο να κάνει σύγκριση ανάμεσα στην πραγματική και στην προβλεπόμενη έξοδο. Στην περίπτωση που υπάρχουν αστοχίες μέσω του αλγόριθμου Backpropagation τα δεδομένα στέλνονται για επανεξέταση και διόρθωση απωλειών.

3. Κίνηση προς τα πίσω: Ενημερώνει τα βάρη του μοντέλου με βάση τη ροή της κλίσης αφού όμως πρώτα γίνει η προσέγγιση της απώλειας (Noriega, 2005).

5.4.4 Τυχαία δάση

Ο αλγόριθμος τυχαίων δασών κατασκευάζει μεγάλο αριθμό δέντρων απόφασης και στη συνέχεια προκειμένου να εξάγει ως αποτέλεσμα μια πρόβλεψη που είναι όσο το δυνατόν ακριβέστερη τα συγχωνεύει. Λειτουργεί με υπερπαραμέτρους που είναι παρόμοιες των δέντρων απόφασης. Καθώς τα δέντρα επεκτείνονται ο αλγόριθμος προσθέτει συνεχώς μεγαλύτερη τυχαιότητα στο μοντέλο. Σκοπός του είναι να ανακαλύπτει το καλύτερο χαρακτηριστικό σε υποσύνολα που εμπεριέχουν χαρακτηριστικά και να λαμβάνεται υπόψη έτσι ώστε να πραγματοποιείται η διάσπαση κάθε κόμβου. Επιπλέον δυνατότητα του είναι η ευκολία στη μέτρηση της σημασίας ενός χαρακτηριστικού έτσι ώστε να αφαιρεθούν τα χαρακτηριστικά που δεν είναι σημαντικά κατά την πρόβλεψη. Μπορεί να εξάγει ικανοποιητικό αποτέλεσμα ακόμη και με τις προκαθορισμένες υπερπαραμέτρους. Όταν το πλήθος των δέντρων είναι σε μεγάλο βαθμό υψηλό τότε ενδέχεται ο αλγόριθμος να μην εξάγει επιθυμητά αποτελέσματα και να μην είναι γρήγορος διότι είναι αργός σε κατασκευή προβλέψεων (Pavlov, 2019).

5.4.5 k-πλησιέστερος γείτονας

Λειτουργεί σύμφωνα με την ιδέα υπολογισμού ευκλείδειας απόστασης ανάμεσα σε δύο στοιχεία που υπάρχουν σε ένα γράφημα. Τα βήματα εφαρμογής του αλγο-

ρίθμου είναι:

1. Τροφοδότηση των δεδομένων
2. Ορισμός της τιμής K που αναπαριστά το πλήθος των γειτόνων
3. Για κάθε στοιχείο στην αναπαράσταση του γραφήματος πραγματοποιούνται τα εξής:
   (α′) Εκτίμηση της απόστασης μεταξύ του στοιχείου που εξετάζεται τη δεδομένη στιγμή με ένα γειτονικό του
   (β′) Αποθήκευση του δείκτη και της απόστασης σε μια λίστα
4. Ταξινόμηση της λίστας που περιέχει τους δείκτες και τις αποστάσεις σε φθίνουσα σειρά
5. Εξαγωγή των πρώτων K στοιχείων από τη λίστα
6. Επιλογή των ετικετών που έχουν σχέση με τις εγγραφές K

Πλεονεκτήματα του αλγοριθμού αποτελούν η απλότητα και η ευκολία προσαρμογής του σε ένα πρόβλημα, η ευελιξία του καθώς μπορεί να χρησιμοποιηθεί τόσο σε προβλήματα παλινδρόμησης όσο και προβλήματα ταξινόμησης, δεν έχει ως απαραίτητη προϋπόθεση την κατασκευή ενός μοντέλου και την εφαρμογή αρκετών παραμέτρων σε αυτό. Μειονέκτημα του αποτελεί πως δεν είναι γρηγόρος καθώς το πλήθος των δειγμάτων γίνεται υψηλότερο (Mucherino, Papajorgji, & Pardalos, 2009).

5.5 Αλγόριθμοι που χρησιμοποιήθηκαν μόνο στην παλινδρόμηση

5.5.1 Γραμμική παλινδρόμηση

Αποτελεί μοντέλο γραμμικό ανάμεσα σε αριθμητικές μεταβλητές που δίνονται στην είσοδο x και αυτές που εξάγονται ως αποτέλεσμα στην έξοδο y αφού πρώτα γίνει η εκτίμηση τους. Στην περίπτωση που τροφοδοτούνται αρκετά στοιχεία στην είσοδο τότε η μέθοδος ονομάζεται πολλαπλή γραμμική παλινδρόμηση, ενώ όταν δε δοθούν παραπάνω από ένα στοιχεία ονομάζεται απλή γραμμική παλινδρόμηση (Brownlee, 2020γ΄). Ένα πρόβλημα παλινδρόμησης που δεν είναι περίπλοκο μπορεί να αναπαρασταθεί μέσω της εξίσωσης $y = a_0 + a_1 * x$. Σκοπός του αλγορίθμου

αποτελεί η εύρεση των βέλτιστων δυνατών τιμών του συντελεστή a_0 και a_1 μέσω της συνάρτησης κόστους, η οποία είναι γνωστή και ως MSE. Τέλος, η συνεχής ενημέρωση στις τιμές των συντελεστών a_0 και a_1 πραγματοποιείται εξετάζοντας τις κλίσεις, οι οποίες προκύπτουν από παράγωγα των δύο συντελεστών πάνω στη συνάρτηση κόστους (Seber & Lee, 2012).

5.5.2 Παλινδρόμηση κορυφογραμμής

Αποτελεί τον τρόπο που μπορεί να κατασκευαστεί ένα μοντέλο στο οποίο το πλήθος των παρατηρήσιμων στοιχείων είναι μικρότερο από το πλήθος των μεταβλητών πρόβλεψης. Διαχωρίζει το ασήμαντο στοιχείο από το σημαντικό έτσι ώστε να μην προκαλείται το πρόβλημα της υπερμοντελοποίησης και να μπορούν να βρεθούν με ευκολία οι λύσεις που θα είναι και μοναδικές. Ανήκει σε μια ομάδα εργαλειών παλινδρόμησης που βασίζονται στον συντελεστή $L2$. Ο συντελεστής $L1$ παράγει και προσθέτει μια ποινή που είναι ίση με τον αριθμό των συντελεστών με αποτέλεσμα μερικοί από τους συντελεστές να αφαιρεθούν. Απο την άλλη ο συντελεστής L2 παράγει και ενσωματώνει μια ποινή, η οποία ισούται με το τετράγωνο του πλήθους των συντελεστών χωρίς να εξάγει ως αποτέλεσμα μοντέλα που θα έχουν μεγάλη αραίωση μεταξύ τους (*Ridge Regression: Simple Definition - Statistics How To*, 2021).

5.5.3 Παλινδρόμηση Lasso

Είναι μέθοδος κανονικοποίησης όπως και η παλινδρόμηση κορυφογραμμών. Η ακριβέστερη πρόβλεψη επιτυγχάνεται μέσω διαφόρων μεθόδων παλινδρόμησης. Οι τιμές των στοιχείων που περιλαμβάνονται στο σύνολο δεδομένων συγχωνεύονται σε ένα σημείο που είναι κεντρικό και μπορεί να θεωρηθεί ως μέσος όρος και παρουσιάζουν αραιώσεις μεταξύ τους. Ο αλγόριθμος αυτός συνιστάται όταν επιθυμεί ο χρήστης να προβεί στην αυτοματοποίηση καταστροφής διαφόρων παραμέτρων όσο και στην επιλογή τμημάτων από ένα μοντέλο. Επιπλέον η συνεργασία του είναι αρκετά καλή με μοντέλα που βρίσκονται σε μεγάλα επίπεδα πολυγραμμικότητας. Χρησιμοποιεί L1 τεχνική κανονικοποίησης δηλαδή λαμβάνονται υπόψη μόνο τα βάρη χωρίς να την ενδιαφέρουν οι συντελεστές που είναι υψηλού μεγέθους. Η μαθηματική

εξίσωση της παλινδρόμησης Lasso είναι η εξής:

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

Το γράμμα λ αναπαριστά το μέγεθος στο οποίο θα γίνει η σμίκρυνση. Όταν είναι ίσο με τον αριθμό μηδέν τότε θα ληφθούν υπόψη όλα τα χαρακτηριστικά, ενώ όταν είναι ίσο με το άπειρο δε θα λαμβάνεται κανένα χαρακτηριστικό υπόψη. Όσο μικρότερη είναι η τιμή του λ τόσο θα μεγαλώνει η διακύμανση, ενώ όσο μεγαλώνει τόσο θα παρατηρείται αύξηση της προκατάληψης (*What is LASSO Regression Definition, Examples and Techniques*, 2020).

5.6 Περιγραφή κώδικα

Στην ενότητα αυτή θα πραγματοποιηθεί η περιγραφή του κώδικα της πειραματικής διαδικασίας. Πιο συγκεκριμένα θα περιγραφούν τα αρχεία classification.py, regression.py, algorithms.py και functions.py καθώς επίσης και διάφορες συναρτήσεις όπως είναι η init και find_best_scale_method.

1. Αρχεία classification.py και regression.py: Η δομή των δύο αρχείων είναι παρόμοια. Τόσο στο αρχείο της κατηγοροποίησης όσο και στο αρχείο της παλινδρόμησης συλλέγονται όλα τα σύνολα δεδομένων τα οποία θα τροφοδοτήσουν την είσοδο των πειραματικών διαδικασιών.

Απεικόνιση 5.1: Δομή αρχείου classification.py

```
files = [
    'classification_datasets/cars.csv',
    'classification_datasets/abalone.csv',
    'classification_datasets/bloodTransfusion.csv',
    'classification_datasets/TTticacoe.csv',
    'classification_datasets/balanceScale.csv',
    'classification_datasets/bestSellersBooks.csv',
    'classification_datasets/haberman.csv',
    'classification_datasets/hayesRoth.csv',
    'classification_datasets/lenses.csv',
```

```
    'classification_datasets/drugTypes.csv'
]


init(files, 'Classification')
```

2. Αρχείο functions.py

   (α′) Συνάρτηση init: Δέχεται τα σύνολα δεδομένων που προέρχονται από το αρχείο classification.py ή το regression.py. Στη συνέχεια, εφαρμόζονται ενέργειες κωδικοποίησης κατηγορηματικών τιμών σε αριθμητικές αν αυτές υπάρχουν και η αντικατάσταση των άγνωστων τιμών χρησιμοποιώντας έναν από τους αλγόριθμους αντιμετώπισης ελλιπών τιμών. Στο τέλος, πραγματοποιείται η εκκίνηση των πειραματικών διαδικασιών.

Απεικόνιση 5.2: Δομή συνάρτησης init

```
def init(files, method):
    print(f"Method: {method}")
    for file in files:
        file_name = file.split('/')[1].split('.')[0]
        print(f"Dataset: {file_name}")

        # Import data
        df = pd.read_csv(file, error_bad_lines=False)

        # Target variable
        target = df.columns[-1]

        # List with dummy variables to encode
        dummy_variables = [column for column in df.
           dtypes.index[:-1] if df.dtypes[column] ==
           object]

        # Fill data with missing values
```

```
        answer = input('Fill data with missing values ?
            (Y/N or y/n) : ')
        if answer.lower() == 'y':
            df = fill_with_missing_values(df)

        # Encoding dummy variables
        if dummy_variables: df = pd.get_dummies(df,
           columns=dummy_variables, drop_first=True)

        # Creating features and target arrays
        X = df.drop(target, axis=1).values
        y = df[target].values

        # Get results after imputing missing data with
           different imputers
        if df.isnull().values.any():
            for imputer in imputers:
                name_imputer = imputer().__class__.
                   __name__
                print(f"Imputer: {name_imputer}")
                imp = get_imputer_function(name_imputer
                   , imputer)
                X_fitted = imp.fit_transform(X)
                # Get results before data preprocessing
                get_results(X_fitted, y, method,
                   file_name)
                print('')
            return

        # Get results without missing data
        get_results(X, y, method, file_name)
        print('')
```

3. Συνάρτηση get_ results: Αρμοδιότητα της αποτελεί η εξαγωγή των αποτελεσμάτων πριν και μετά την προεπεξεργασία των δεδομένων. Πιο συγκεκριμένα, μέσω της μεθόδου KFold θα πραγματοποιηθεί ο διαχωρισμός των συνόλων δεδομένων σε δέκα διαφορετικά εκπαιδευτικά και δοκιμαστικά σύνολα. Στη συνέχεια, ανάλογα με τη μέθοδο (κατηγοριοποίηση ή παλινδρόμηση), λαμβάνονται και οι απαραίτητοι αλγόριθμοι που θα συμμετάσχουν στην εκπαίδευση των μοντέλων και θα συνδυαστούν με τον καλύτερο αλγόριθμο κλιμάκωσης για το συγκεκριμένο σύνολο δεδομένων.

Απεικόνιση 5.3: Δομή συνάρτησης get_results

```
def get_results(X, y, method, file_name):
    scores_before = []
    scores_after = []

    # K-Fold Cross Validation to Split Training and Test
       Sets
    folds = KFold(n_splits=3, random_state=21, shuffle=True
       )

    # Get all Classification or Regression Algorithms
       Setups Before and After Data Preprocessing
    algorithms = get_class_algorithms(folds) if method == '
       Classification' else get_reg_algorithms(folds,
       file_name)


    for name, setup in algorithms.items():
        for train_index, test_index in folds.split(X, y):
            X_train, X_test, y_train, y_test = X[
               train_index], X[test_index], y[train_index],
                y[test_index]
            for status, algorithm in setup.items():
                if status == 'Before':
```

```
                    algorithm.fit(X_train, y_train)
                    scores_before.append(algorithm.score(
                        X_test, y_test))
                else:
                    # Find best scaling method
                    best_scaler_name, X_train, X_test =
                        find_best_scale_method(X_train,
                        X_test)
                    algorithm.fit(X_train, y_train)
                    scores_after.append(algorithm.score(
                        X_test, y_test))
        print('====================')
        print(f"{name} {method}")
        print(f"Scaler: {best_scaler_name}")
        print(f"Before Data Preprocessing Score: {math.ceil
            (np.average(scores_before) * 100)}%")
        print(f"After Data Preprocessing Score: {math.ceil(
            np.average(scores_after) * 100)}%")
        print('====================')
        scores_before.clear()
        scores_after.clear()
```

4. Συνάρτηση find_best_scale_method: Στόχος της αποτελεί η εύρεση του καλύτερου αλγορίθμου κλιμάκωσης. Η εύρεση του πραγματοποιείται αφού πρώτα εξεταστούν όλοι οι μέσοι όροι και βρεθεί ο μεγαλύτερος, ο οποίος θα καθορίσει και τον αλγόριθμο που θα χρησιμοποιηθεί στην πειραματική διαδικασία για το συγκεκριμένο σύνολο δεδομένων.

Απεικόνιση 5.4: Δομή συνάρτησης find_best_scale_method

```
def find_best_scale_method(X_train, X_test):
    max_score_X_train = -100
    max_score_X_test = -100
    best_scaler_name = ''
```

```
    for scaler in scalers:
        if np.average(scaler().fit_transform(X_train)) >
           max_score_X_train:
            best_scaler_name = scaler().__class__.__name__
            max_score_X_train = np.average(scaler().
               fit_transform(X_train))
            X_train = scaler().fit_transform(X_train)
        if np.average(scaler().fit_transform(X_test)) >
           max_score_X_test:
            max_score_X_test = np.average(scaler().
               fit_transform(X_test))
            X_test = scaler().fit_transform(X_test)
    return best_scaler_name, X_train, X_test
```

5. Αρχείο algorithms.py: Στο αρχείο αυτό υπάρχουν συνολικά δύο συναρτήσεις, η get_class_algorithms καθώς και η συνάρτηση get_reg_algorithms, οι οποίες έχουν παρόμοια δομή. Η συνάρτηση get_reg_algorithms είναι υπεύθυνη τόσο για την κατασκευή όσο και για την τροφοδότηση των αλγορίθμων παλινδρόμησης με τις προκαθορισμένες αλλά και με τις υπερπαραμέτρους που ενσωματώθηκαν μετά τη λειτουργία του RandomizedSearchCV στη συνάρτηση get_results του αρχείου functions.py που αναλύθηκε παραπάνω. Το ίδιο ισχύει και για τη συνάρτηση get_class_algorithms, η οποία όμως δημιουργεί και τροφοδοτεί τους αλγόριθμους κατηγοριοποίησης.

Απεικόνιση 5.5: Δομή συνάρτησης get_reg_algorithms

```
def get_reg_algorithms(folds, file_name):
    svr_model = SVR()
    linear_model = LinearRegression(fit_intercept=False)
    ridge_model = Ridge(normalize=True)
    lasso_model = Lasso(normalize=True)
    dtm_model = DecisionTreeRegressor()
    rfr_model = RandomForestRegressor()
    mlp_model = MLPRegressor(max_iter=100)
```

```
knn_model = KNeighborsRegressor()

param_grid_svr = {
    'C': [0.1 , 1, 10, 100, 1000],
    'epsilon': [0.0001, 0.001, 0.01, 0.1, 1]
}

param_grid_linear = {
        'kernel': ['linear'],
        'C': [1e-03, 1e-02, 0.1, 1],
        'gamma': [1e-03, 1e-02, 0.1, 1],
        'epsilon': [1e-02, 0.1, 1],
}

if file_name == 'computer_hardware':
    param_grid_linear = {
        'kernel': ['rbf'],
        'C': [10000, 100000, 500000]
    }


param_grid_ridge_lasso = {
    'alpha': [1e-04, 1e-03, 1e-02, 0.1, 1]
}

param_grid_dtm = {
    'criterion': ['mse', 'mae'],
    'max_features': ['auto'],
    'max_depth': range(2, 32, 2),
    'random_state': range(0, 10)
}
```

```
    param_grid_rfr = {
        'bootstrap': [True],
        'max_features': ['auto'],
        'max_depth': range(70, 120)
    }

    param_grid_mlp = {
        'solver': ['lbfgs'],
        'alpha': 10.0 ** -np.arange(1, 10),
        'hidden_layer_sizes': np.arange(10, 15)
    }

    param_grid_knn = {
        'algorithm':['auto', 'ball_tree','kd_tree','brute']
           ,
        'weights':['uniform', 'distance'],
        'n_neighbors': list(range(5, 11)),
        'leaf_size': list(range(1, 6))
    }

    return {
        'SVR': {
            'Before': svr_model,
            'After': RandomizedSearchCV(svr_model,
               param_grid_svr, cv=folds, n_jobs=-1)
        },
        'Linear': {
            'Before': linear_model,
            'After': RandomizedSearchCV(svr_model,
               param_grid_linear, cv=folds, n_jobs=-1)
        },
        'Ridge': {
```

```
            'Before': ridge_model,
            'After': RandomizedSearchCV(ridge_model,
               param_grid_ridge_lasso, cv=folds, n_jobs=-1)
        },
        'Lasso': {
            'Before': lasso_model,
            'After': RandomizedSearchCV(lasso_model,
               param_grid_ridge_lasso, cv=folds, n_jobs=-1)
               ,
        },
        'Decision Tree': {
            'Before': dtm_model,
            'After': RandomizedSearchCV(dtm_model,
               param_grid_dtm, cv=folds, n_jobs=-1)
        },
        'Random Forest': {
            'Before': rfr_model,
            'After': RandomizedSearchCV(rfr_model,
               param_grid_rfr, cv=folds, n_jobs=-1)
        },
        'MLP': {
            'Before': mlp_model,
            'After': RandomizedSearchCV(mlp_model,
               param_grid_mlp, cv=folds, n_jobs=-1)
        },
        'KNN': {
            'Before': knn_model,
            'After': RandomizedSearchCV(knn_model,
               param_grid_knn, cv=folds, n_jobs=-1)
        }
    }
```

5.7 Σύγκριση αποτελεσμάτων για κάθε σύνολο δεδομένων

Στους παρακάτω πίνακες με τα αποτελέσματα στη στήλη Αλγόριθμος αναπαριστώνται οι αλγόριθμοι που χρησιμοποιήθηκαν στην πειραματική διαδικασία της κατηγοριοποίησης ή της παλινδρόμησης. Στη στήλη Σκορ πριν την ΠΕ αναπαριστάται ο μέσος όρος του ποσοστού ακρίβειας όλων των συνόλων δεδομένων πριν την προεπεξεργασία ενώ στη στήλη Σκορ μετά την ΠΕ αναπαριστάται ο μέσος όρος του ποσοστού ακρίβειας όλων των συνόλων δεδομένων αφού έγινε η προεπεξεργασία.

5.7.1 Σύγκριση αποτελεσμάτων για σύνολα δεδομένων κατηγοριοποίησης

Πίνακας 5.1: Αποτελέσματα συνόλου δεδομένων cars

Αλγόριθμος	**Σκορ πριν την ΠΕ**	**Σκορ μετά την ΠΕ**
SVM	96%	98%
Δέντρα απόφασης	92%	92%
MLP	96%	98%
Τυχαία Δάση	82%	86%
KNN	91%	91%

Στον Πίνακα 5.1 απεικονίζονται τα ποσοστά ακρίβειας του συνόλου δεδομένων cars πριν και μετά την προεπεξεργασία των στοιχείων του. Ο καλύτερος αλγόριθμος κλιμάκωσης που χρησιμοποιήθηκε σε συνδυασμό με όλους τους αλγόριθμους για το συγκεκριμένο σύνολο δεδομένων ήταν ο RobustScaler. Το καλύτερο σκορ μετά την προεπεξεργασία δεδομένων συγκέντρωσαν τόσο ο αλγόριθμος SVM όσο και ο αλγόριθμος MLP με ποσοστό ακρίβειας 98%. Στη συνέχεια, δεύτερος καλύτερος είναι ο αλγόριθμος των δέντρων απόφασης με σκορ 92%, τρίτος ο KNN με σκορ 91% και τελευταίος ο αλγόριθμος τυχαίων δασών με ποσοστό ακρίβειας 86%. Ο αλγόριθμος που ευνοήθηκε περισσότερο από την κλιμάκωση ήταν ο αλγόριθμος τυχαίων δασών.

Πίνακας 5.2: Αποτελέσματα συνόλου δεδομένων abalone

Αλγόριθμος	**Σκορ πριν την ΠΕ**	**Σκορ μετά την ΠΕ**
SVM	26%	26%
Δέντρα απόφασης	20%	25%
MLP	28%	28%
Τυχαία Δάση	24%	24%
KNN	26%	26%

Στον Πίνακα 5.2 απεικονίζονται τα ποσοστά ακρίβειας του συνόλου δεδομένων abalone πριν και μετά την προεπεξεργασία των χαρακτηριστικών του. Σε αυτό το σύνολο ο αλγόριθμος κλιμάκωσης που συνδυάστηκε καλύτερα με όλους τους αλγόριθμους μηχανικής μάθησης ήταν ο QuantileTransformer. Το υψηλότερο σκορ συγκέντρωσε ο αλγόριθμος MLP με ποσοστό ακρίβειας 28%. Στη δεύτερη θέση βρίσκονται οι αλγόριθμοι SVM και KNN με σκορ 26%. Στην τρίτη θέση ο αλγόριθμος δέντρων απόφασης με ακρίβεια 25% και στην τελευταία ο αλγόριθμος τυχαίων δασών με σκορ 24%. Πρέπει να σημειωθεί πως οι διαφορές ανάμεσα στα ποσοστά ακρίβειας ήταν σχεδόν αμελητέες δηλαδή οριακά κάποιος αλγόριθμος εξήγαγε υψηλότερο σκορ από τον κατώτερο του. Ο αλγόριθμος που ευνοήθηκε περισσότερο από την κλιμάκωση ήταν ο αλγόριθμος δέντρων απόφασης.

Πίνακας 5.3: Αποτελέσματα συνόλου δεδομένων bloodTransfusion

Αλγόριθμος	**Σκορ πριν την ΠΕ**	**Σκορ μετά την ΠΕ**
SVM	77%	79%
Δέντρα απόφασης	72%	76%
MLP	74%	79%
Τυχαία Δάση	75%	78%
KNN	75%	77%

Στον Πίνακα 5.3 αποτυπώνονται τα σκορ του συνόλου δεδομένων bloodTransfusion πριν και μετά την προεπεξεργασία των στοιχείων του. Ο αλγόριθμος κλιμάκωσης που χρησιμοποιήθηκε σε αυτό το σύνολο δεδομένων και είχε τα καλύτερα αποτελέσματα σε συνδυασμό με τους υπόλοιπους αλγόριθμους μηχανικής μάθησης ήταν ο QuantileTransformer. Το μεγαλύτερο ποσοστό ακρίβειας το πέτυχε ο αλγόριθμος MLP καθώς επίσης και ο αλγόριθμος SVM με σκορ 79%. Το δεύτερο μεγαλύτερο σκορ εξήγαγε ο αλγόριθμος τυχαίων δασών με ποσοστό ακρίβειας 78%, το τρίτο μεγαλύτερο ο αλγόριθμος των K πλησιέστερων γειτόνων με σκορ 77% και στην τελευταία θέση βρίσκεται ο αλγόριθμος δέντρων απόφασης με ποσοστό 76%. Ο αλγόριθμος που ευνοήθηκε περισσότερο από την κλιμάκωση ήταν ο αλγόριθμος MLP και δέντρων απόφασης.

Στον Πίνακα 5.4 απεικονίζονται τα σκορ του συνόλου δεδομένων ticTacToe πριν και μετά την προεπεξεργασία των χαρακτηριστικών του. Ο αλγόριθμος κλιμάκωσης MinMaxScaler ήταν ο καλύτερος και χρησιμοποιήθηκε συνδυαστικά με όλους τους αλγορίθμους μηχανικής μάθησης στο συνόλο δεδομένων ticTacToe. Στην πρώτη θέση

Πίνακας 5.4: Αποτελέσματα συνόλου δεδομένων ticTacToe

Αλγόριθμος	**Σκορ πριν την ΠΕ**	**Σκορ μετά την ΠΕ**
SVM	99%	100%
Δέντρα απόφασης	97%	97%
MLP	99%	99%
Τυχαία Δάση	100%	100%
KNN	100%	100%

υπάρχουν τρεις αλγόριθμοι. Οι αλγόριθμοι αυτοί είναι τα τυχαία δάση, ο αλγόριθμος Κ πλησιέστερων γειτόνων και ο αλγόριθμος μηχανών διανυσμάτων υποστήριξης με ποσοστό ακρίβειας 100%. Την δεύτερη θέση λαμβάνει ο αλγόριθμος MLP, ο οποίος βρίσκεται ελάχιστα πιο πίσω σε σχέση με τους υπόλοιπους αλγόριθμους που βρίσκονται στην πρώτη θέση με ποσοστό 99%. Τέλος, την τρίτη και τελευταία θέση παίρνει ο αλγόριθμος δέντρων απόφασης με σκορ 97%.

Πίνακας 5.5: Αποτελέσματα συνόλου δεδομένων balanceScale

Αλγόριθμος	**Σκορ πριν την ΠΕ**	**Σκορ μετά την ΠΕ**
SVM	24%	24%
Δέντρα απόφασης	6%	28%
MLP	27%	27%
Τυχαία Δάση	9%	13%
KNN	6%	31%

Στον Πίνακα 5.5 αποτυπώνονται τα ποσοστά ακρίβειας του συνόλου δεδομένων balanceScale πριν και μετά την προεπεξεργασία των στοιχείων του. Μετά την προεπεξεργασία δεδομένων παρατηρούνται σημαντικές βελτιώσεις επιδόσεων στους αλγόριθμους δέντρων απόφασης, τυχαίων δασών και Κ πλησιέστερων γειτόνων. Στην πρώτη θέση βρίσκεται ο αλγόριθμος Κ πλησιέστερων γειτόνων με ποσοστό ακρίβειας 31%. Δεύτερος στην κατάταξη βρίσκεται ο αλγόριθμος δέντρων απόφασης με σκορ 28%. Στη συνέχεια, στην τρίτη θέση και ελάχιστα πιο πίσω βρίσκεται ο αλγόριθμος MLP με ποσοστό ακρίβειας 27%, ακολουθώντας ο αλγόριθμος SVM που έχει σκορ 24%. Τελευταίος είναι ο αλγόριθμων τυχαίων δασών με αρκετά χαμηλότερο ποσοστό ακρίβειας ίσο με 13%. Οι αλγόριθμοι που ευνοήθηκαν περισσότερο από την κλιμάκωση των στοιχείων του συνόλου δεδομένων ήταν ο αλγόριθμος δέντρων απόφασης, τυχαίων δασών και ο KNN. Τέλος, ο αλγόριθμος κλιμάκωσης που χρησιμοποιήθηκε με τον καλύτερο τρόπο σε όλες τις περιπτώσεις ήταν ο StandardScaler.

Στον Πίνακα 5.6 αναπαριστώνται τα ποσοστά ακρίβειας του συνόλου δεδο-

Πίνακας 5.6: Αποτελέσματα συνόλου δεδομένων bestSellersBooks

Αλγόριθμος	**Σκορ πριν την ΠΕ**	**Σκορ μετά την ΠΕ**
SVM	64%	90%
Δέντρα απόφασης	83%	74%
MLP	56%	91%
Τυχαία Δάση	71%	88%
KNN	88%	88%

μένων bestSellersBooks πριν και μετά την προεπεξεργασία των στοιχείων του. Οι αλγόριθμοι που ευνοήθηκαν σε υψηλότερο βαθμό από την κλιμάκωση των δεδομένων ήταν ο SVM, ο αλγόριθμος τυχαίων δασών και ο MLP. Η χρήση του αλγορίθμου δέντρων απόφασης δε συνίσταται για αυτό το σύνολο δεδομένων διότι μετά την προεπεξεργασία δεδομένων συγκεντρώνει το χαμηλότερο σκορ αγγίζοντας το 74%. Οι υπόλοιποι αλγόριθμοι σε αντίθεση με τον αλγόριθμο δέντρων απόφασης παρουσιάζουν σημαντική βελτίωση έπειτα από την προεπεξεργασία των δεδομένων. Πιο συγκεκριμένα στην πρώτη θέση της κατάταξης βρίσκεται ο αλγόριθμος MLP με ποσοστό ακρίβειας 91%, στη δεύτερη θέση με 90% σκορ ο SVM, στην τρίτη θέση ο αλγόριθμος KNN και ο αλγόριθμος τυχαίων δασών με ποσοστό ακρίβειας 88%. Ο αλγόριθμος κλιμάκωσης που λειτούργησε καλύτερα σε όλες τις περιπτώσεις είναι ο QuantileTransformer.

Πίνακας 5.7: Αποτελέσματα συνόλου δεδομένων haberman

Αλγόριθμος	**Σκορ πριν την ΠΕ**	**Σκορ μετά την ΠΕ**
SVM	73%	69%
Δέντρα απόφασης	67%	73%
MLP	76%	69%
Τυχαία Δάση	73%	74%
KNN	72%	74%

Στον Πίνακα 5.7 αποτυπώνονται τα σκορ του συνόλου δεδομένων haberman πριν και μετά την προεπεξεργασία των χαρακτηριστικών του. Αρχικά, θα πρέπει να αναφερθεί πως οι αλγόριθμοι MLP και SVM μετά την προεπεξεργασία των δεδομένων συγκεντρώνουν μικρότερο ποσοστό ακρίβειας που είναι ίσο με 69%. Στην πρώτη θέση και με σκορ 74% βρίσκονται οι αλγόριθμοι τυχαίων δασών και K πλησιέστερων γειτόνων. Στη συνέχεια, με αρκετά μικρή διαφορά και ποσοστό 73% ακολουθεί ο αλγόριθμος δέντρων απόφασης. Ο αλγόριθμος κλιμάκωσης που προτιμήθηκε για όλες τις περιπτώσεις επειδή ήταν ο καλύτερος σε αυτό το σύνολο δεδομένων είναι

ο QuantileTransformer. Τέλος, ο αλγόριθμος που ευνοήθηκε σε μεγαλύτερο βαθμό από την κλιμάκωση ήταν ο δέντρων απόφασης.

Πίνακας 5.8: Αποτελέσματα συνόλου δεδομένων hayesRoth

Αλγόριθμος	**Σκορ πριν την ΠΕ**	**Σκορ μετά την ΠΕ**
SVM	37%	62%
Δέντρα απόφασης	84%	73%
MLP	37%	70%
Τυχαία Δάση	42%	62%
KNN	82%	73%

Στον Πίνακα 5.8 απεικονίζονται τα ποσοστά ακρίβειας του συνόλου δεδομένων hayesRoth πριν και μετά την προεπεξεργασία των στοιχείων του. Ο αλγόριθμος κλιμάκωσης που εφαρμόστηκε σε όλες τις περιπτώσεις επειδή ήταν ο καλύτερος είναι ο QuantileTransformer. Οι αλγόριθμοι KNN και δέντρων απόφασης παρουσίασαν μείωση του σκορ σε σχέση με το σκορ που είχαν πριν πραγματοποιηθεί η διαδικασία της προεπεξεργασίας επειδή στο σύνολο δεδομένων υπήρξαν στοιχεία με θόρυβο. Παρόλα αυτά όμως κατάφεραν να συγκεντρώσουν το μεγαλύτερο ποσοστό ακρίβειας ίσο με 73%. Στη δεύτερη θέση βρίσκεται ο MLP με σκορ 70%. Την τρίτη και τελευταία θέση με ποσοστό ακρίβειας ίσο με το 62% κατέχουν οι αλγόριθμοι τυχαίων δασών και SVM. Οι αλγόριθμοι μηχανικής μάθησης που ευνοήθηκαν περισσότερο από την κλιμάκωση ήταν ο SVM, ο MLP και ο αλγόριθμος τυχαίων δασών.

Πίνακας 5.9: Αποτελέσματα συνόλου δεδομένων lenses

Αλγόριθμος	**Σκορ πριν την ΠΕ**	**Σκορ μετά την ΠΕ**
SVM	67%	80%
Δέντρα απόφασης	85%	77%
MLP	75%	72%
Τυχαία Δάση	77%	72%
KNN	82%	70%

Στον Πίνακα 5.9 απεικονίζονται τα σκορ του συνόλου δεδομένων lenses πριν και μετά την προεπεξεργασία των στοιχείων του. Ο αλγόριθμος που ευνοήθηκε σε μεγαλύτερο βαθμό από την κλιμάκωση ήταν ο SVM. Μετά από την προεπεξεργασία του συνόλου δεδομένων, την πρώτη θέση της κατάταξης με σκορ 80% λαμβάνει ο αλγόριθμος SVM. Ελάχιστα πιο πίσω του βρίσκεται ο αλγόριθμος δέντρων απόφασης με ποσοστό ακρίβειας 77%. Η τρίτη θέση της κατάταξης απονέμεται στους

αλγόριθμους MLP νευρικών δικτύων και τυχαίων δασών. Τέλος, στην τέταρτη θέση βρίσκεται ο αλγόριθμος Κ πλησιέστερων γειτόνων. Ο αλγόριθμος κλιμάκωσης που χρησιμοποιήθηκε καλύτερα σε συνδυασμό με όλους τους αλγόριθμους μηχανικής μάθησης ήταν ο MinMaxScaler.

Πίνακας 5.10: Αποτελέσματα συνόλου δεδομένων drugTypes

Αλγόριθμος	**Σκορ πριν την ΠΕ**	**Σκορ μετά την ΠΕ**
SVM	70%	93%
Δέντρα απόφασης	99%	92%
MLP	59%	93%
Τυχαία Δάση	69%	86%
KNN	99%	93%

Στον Πίνακα 5.10 αποτυπώνονται τα ποσοστά ακρίβειας του συνόλου δεδομένων drugTypes πριν και μετά την προεπεξεργασία των χαρακτηριστικών του. Σε αυτό το σύνολο δεδομένων μπορεί να επιλεχθεί οποιοσδήποτε αλγόριθμος μεταξύ του KNN, των μηχανών διανυσμάτων υποστήριξης και των νευρωνικών δικτύων για να εκπαιδευτεί το μοντέλο επειδή βρίσκονται πρώτοι στην κατάταξη με ποσοστό ακρίβειας 93%. Ελάχιστα πιο πίσω και στη δεύτερη θέση με σκορ 92% βρίσκεται ο αλγόριθμος δέντρων απόφασης και στην τελευταία θέση ο αλγόριθμος τυχαίων δασών με ποσοστό 86%. Ο PowerTransformer είναι ο αλγόριθμος κλιμάκωσης που παρουσίασε τα καλύτερα αποτελέσματα και χρησιμοποιείται σε όλες τις περιπτώσεις. Τέλος, οι αλγόριθμοι που ευνοήθηκαν περισσότερο από την κλιμάκωση ήταν ο MLP, ο αλγόριθμος τυχαίων δασών και ο SVM.

5.7.2 Σύγκριση αποτελεσμάτων για σύνολα δεδομένων παλινδρόμησης

Στον Πίνακα 5.11 αναπαριστώνται τα σκορ του συνόλου δεδομένων aquaticToxicity πριν και μετά την προεπεξεργασία των στοιχείων του. Όλοι οι αλγόριθμοι εκτός από τον αλγόριθμο τυχαίων δασών ευνοήθηκαν περισσότερο από την κλιμάκωση του συνόλου δεδομένων. Ο αλγόριθμος που εξάγει το μεγαλύτερο ποσοστό ακρίβειας ίσο με 54% είναι ο SVR. Στη συνέχεια, τον ακολουθούν στη δεύτερη θέση ο αλγόριθμος Κ πλησιέστερων γειτόνων και των τυχαίων δασών με σκορ 52%. Στην τρίτη θέση βρίσκεται ο MLP με σκορ ίσο με 49%. Οι αλγόριθμοι παλινδρόμησης κορυφογραμ-

Πίνακας 5.11: Αποτελέσματα συνόλου δεδομένων aquaticToxicity

Αλγόριθμος	**Σκορ πριν την ΠΕ**	**Σκορ μετά την ΠΕ**
SVR	11%	54%
Γραμμική παλ.	33%	41%
Παλ. κορυφογραμ.	33%	41%
Παλ. Lasso	-1%	41%
Δέντρα απόφασης	18%	27%
Τυχαία Δάση	54%	52%
MLP	38%	49%
KNN	33%	52%

μών, η παλινδρόμηση Lasso και η γραμμική παλινδρόμηση συγκεντρώνουν ποσοστό ακρίβειας που αγγίζει το 41% και βρίσκονται στην τέταρτη θέση. Στο τέλος, με αρκετά χαμηλότερο σκορ ίσο με 27% βρίσκεται ο αλγόριθμος δέντρων απόφασης. Ο QuantileTransformer είναι ο αλγόριθμος κλιμάκωσης που συνδυάζεται καλύτερα με τους αλγορίθμους σε αυτό σύνολο δεδομένων.

Πίνακας 5.12: Αποτελέσματα συνόλου δεδομένων slumpTest

Αλγόριθμος	**Σκορ πριν την ΠΕ**	**Σκορ μετά την ΠΕ**
SVR	-2%	49%
Γραμμική παλ.	88%	31%
Παλ. κορυφογραμ.	51%	29%
Παλ. Lasso	-12%	12%
Δέντρα απόφασης	63%	-33%
Τυχαία Δάση	80%	19%
MLP	-2374%	-6%
KNN	67%	29%

Στον Πίνακα 5.12 αναπαριστώνται τα ποσοστά ακρίβειας του συνόλου δεδομένων slumpTest πριν και μετά την προεπεξεργασία των στοιχείων του. Σε αυτό το σύνολο δεδομένων ο SVR ήταν ο αλγόριθμος που συγκέντρωσε το υψηλότερο ποσοστό ακρίβειας ίσο με το 49%. Στη δεύτερη θέση της κατάταξης βρίσκεται ο αλγόριθμος γραμμικής παλινδρόμησης με σκορ που αγγίζει το 31%. Η τρίτη θέση απονέμεται στους αλγόριθμους πλησιέστερων γειτόνων και στην παλινδρόμηση κορυφογραμμής με ποσοστό 29%. Στην τέταρτη θέση με διαφορά 10% βρίσκεται ο αλγόριθμος τυχαίων δασών με ποσοστό 19%. Την πέμπτη θέση κατέχει ο αλγόριθμος παλινδρόμησης Lasso με ποσοστό ακρίβειας 12%. Στην έκτη θέση υπάρχει ο MLP με ποσοστό -6% και στην τελευταία ο αλγόριθμος δέντρων απόφασης με σκορ -33%. Ο αλγόριθμος κλιμάκωσης που λειτούργησε καλύτερα σε αυτό το σύνολο δε-

δομένων ήταν ο MinMaxScaler. Τέλος, οι αλγόριθμοι που ευνοήθηκαν περισσότερο από την κλιμάκωση ήταν ο SVR και ο αλγόριθμος παλινδρόμησης Lasso.

Πίνακας 5.13: Αποτελέσματα συνόλου δεδομένων drivePoints

Αλγόριθμος	**Σκορ πριν την ΠΕ**	**Σκορ μετά την ΠΕ**
SVR	25%	34%
Γραμμική παλ.	100%	-10%
Παλ. κορυφογραμ.	92%	-12%
Παλ. Lasso	67%	-10%
Δέντρα απόφασης	100%	36%
Τυχαία Δάση	100%	36%
MLP	99%	-12%
KNN	100%	59%

Στον Πίνακα 5.13 αποτυπώνονται τα σκορ του συνόλου δεδομένων drivePoints πριν και μετά την προεπεξεργασία των χαρακτηριστικών του. Ο αλγόριθμος κλιμάκωσης που εφαρμόστηκε σε συνδυασμό με τους υπόλοιπους αλγόριθμους μηχανικής μάθησης και ήταν ο καλύτερος είναι ο QuantileTransformer. Πρώτος στην κατάταξη και με αρκετά υψηλότερο σκορ σε σχέση με τους υπόλοιπους ήταν ο αλγόριθμος Κ πλησιέστερων γειτόνων με ποσοστό ακρίβειας 59%. Στη δεύτερη θέση της κατάταξης βρίσκονται οι αλγόριθμοι δέντρων απόφασης και τυχαίων δασών με σκορ που είναι ίσο με 36%. Ελάχιστα πιο πίσω με ποσοστό 34% βρίσκεται ο αλγόριθμος SVR. Στην τέταρτη θέση και με μεγάλη διαφορά στο σκορ σε σχέση με τον SVR βρίσκονται οι αλγόριθμοι παλινδρόμησης Lasso και γραμμικής παλινδρόμησης με ποσοστό ίσο με -10%. Στην τελευταία θέση της κατάταξης με σκορ ίσο με -12% βρίσκονται οι αλγόριθμοι νευρωνικών δικτύων MLP και παλινδρόμησης κορυφογραμμής. Ο αλγόριθμος που ευνοήθηκε περισσότερο από την κλιμάκωση ήταν ο SVR.

Πίνακας 5.14: Αποτελέσματα συνόλου δεδομένων southGermanCredit

Αλγόριθμος	**Σκορ πριν την ΠΕ**	**Σκορ μετά την ΠΕ**
SVR	-14%	13%
Γραμμική παλ.	20%	4%
Παλ. κορυφογραμ.	18%	19%
Παλ. Lasso	0%	19%
Δέντρα απόφασης	-55%	4%
Τυχαία Δάση	22%	19%
MLP	-67%	7%
KNN	-14%	13%

Στον Πίνακα 5.14 αποτυπώνονται τα ποσοστά ακρίβειας του συνόλου δεδομέ-

νων southGermanCredit πριν και μετά την προεπεξεργασία των χαρακτηριστικών του. Μετά την προεπεξεργασία δεδομένων διαπιστώνεται πως οι αλγόριθμοι με την καλύτερη απόδοση είναι ο αλγόριθμος τυχαίων δασών, η παλινδρόμηση Lasso και η παλινδρόμηση κορυφογραμμών με ποσοστό ακρίβειας ίσο με 19%. Στη δεύτερη θέση βρίσκονται οι αλγόριθμοι KNN και SVR με σκορ 13%. Την τρίτη θέση κατέχει ο MLP με ποσοστό 7%. Τέλος, στην τέταρτη θέση βρίσκονται οι αλγόριθμοι γραμμικής παλινδρόμησης και των δέντρων απόφασης με σκορ 4%. Όλοι οι αλγόριθμοι συνδυάστηκαν με τον QuantileTransformer κατά την πειραματική διαδικασία επειδή ήταν αυτός που παρουσίασε τα καλύτερα αποτελέσματα. Η κλιμάκωση ευνόησε περισσότερο τον αλγόριθμο παλινδρόμησης Lasso, τον SVR, τον αλγόριθμο δέντρων απόφασης, τον KNN και τον MLP.

Πίνακας 5.15: Αποτελέσματα συνόλου δεδομένων carPriceAssignment

Αλγόριθμος	**Σκορ πριν την ΠΕ**	**Σκορ μετά την ΠΕ**
SVR	-16%	53%
Γραμμική παλ.	-75%	-12%
Παλ. κορυφογραμ.	79%	69%
Παλ. Lasso	81%	65%
Δέντρα απόφασης	85%	64%
Τυχαία Δάση	92%	69%
MLP	-44%	83%
KNN	83%	77%

Ο Πίνακας 5.15 αναπαριστά τα ποσοστά ακρίβειας του συνόλου δεδομένων carPriceAssignment πριν και μετά την προεπεξεργασία των στοιχείων του. Κατά την πειραματική διαδικασία ο QuantileTransformer ήταν ο αλγόριθμος κλιμάκωσης που ήταν ο καλύτερος και χρησιμοποιήθηκε σε συνδυασμό με τους υπόλοιπους αλγόριθμους μηχανικής μάθησης. Η κλιμάκωση ευνόησε πιο πολύ τους αλγόριθμους MLP και SVR. Στην πρώτη θέση της κατάταξης βρίσκεται ο MLP με αρκετά υψηλότερο σκορ σε σχέση με τους υπόλοιπους συγκεντρώνοντας ποσοστό ακρίβειας ίσο με 83%. Στη δεύτερη θέση βρίσκεται ο αλγόριθμος πλησιέστερων γειτόνων με ποσοστό ίσο του 77%. Στη συνέχεια, ακολουθούν οι αλγόριθμοι τυχαίων δασών και παλινδρόμησης κορυφογραμμών με σκορ ίσο του 69%. Ελάχιστα πιο πίσω στην τέταρτη θέση βρίσκεται ο αλγόριθμος παλινδρόμησης Lasso με ποσοστό ακρίβειας 65%. Με διαφορά 1% σε σχέση με τον αλγόριθμο παλινδρόμησης Lasso στην πέμπτη θέση βρίσκεται ο αλγόριθμος δέντρων απόφασης με σκορ 64%. Στην έκτη θέση βρί-

σκεται ο αλγόριμος μηχανών διανυσμάτων υποστήριξης με ποσοστό 53% και στην τελευταία θέση η γραμμική παλινδρόμηση με ποσοστό -12%.

Πίνακας 5.16: Αποτελέσματα συνόλου δεδομένων insurance

Αλγόριθμος	**Σκορ πριν την ΠΕ**	**Σκορ μετά την ΠΕ**
SVR	-10%	74%
Γραμμική παλ.	72%	-9%
Παλ. κορυφογραμ.	56%	75%
Παλ. Lasso	75%	75%
Δέντρα απόφασης	69%	82%
Τυχαία Δάση	84%	82%
MLP	-58%	85%
KNN	15%	80%

Στον Πίνακα 5.16 αναπαριστώνται τα σκορ του συνόλου δεδομένων insurance πριν και μετά την προεπεξεργασία των χαρακτηριστικών του. Μετά το πέρας της πειραματικής διαδικασίας διαπιστώθηκε πως ο MLP συγκέντρωσε το μεγαλύτερο σκορ σε σχέση με τους υπόλοιπους και βρίσκεται στην πρώτη θέση με σκορ 85%. Τη δεύτερη θέση με ποσοστό ακρίβειας 82% κατακτούν οι αλγόριθμοι τυχαίων δασών και δέντρων απόφασης. Με ποσοστό 80% βρίσκεται στην τρίτη θέση ο αλγόριθμος πλησιέστερων γειτόνων. Πίσω του και στην τέταρτη θέση βρίσκονται οι αλγόριθμοι παλινδρόμησης κορυφογραμμών και Lasso με σκορ 75%. Ελάχιστα πιο πίσω τους με ποσοστό 74% βρίσκεται ο αλγόριθμος SVR και με μεγάλη διαφορά στην τελευταία θέση ακολουθεί ο αλγόριθμος γραμμικής παλινδρόμησης με ποσοστό ακρίβειας -9%. Η κλιμάκωση του συνόλου δεδομένων πραγματοποιήθηκε με την εφαρμογή του αλγορίθμου Normalizer, ο οποίος ήταν και ο καλύτερος και ευνόησε περισσότερο τον αλγόριθμο παλινδρόμησης κορυφογραμμών, δέντρων απόφασης, MLP, K πλησιέστερων γειτόνων και τον SVR.

Πίνακας 5.17: Αποτελέσματα συνόλου δεδομένων realEstate

Αλγόριθμος	**Σκορ πριν την ΠΕ**	**Σκορ μετά την ΠΕ**
SVR	51%	61%
Γραμμική παλ.	58%	58%
Παλ. κορυφογραμ.	53%	61%
Παλ. Lasso	-2%	59%
Δέντρα απόφασης	45%	39%
Τυχαία Δάση	70%	53%
MLP	-255%	58%
KNN	55%	59%

Ο Πίνακας 5.17 απεικονίζει τα ποσοστά ακρίβειας του συνόλου δεδομένων realEstate πριν και μετά την προεπεξεργασία των χαρακτηριστικών του. Μετά την προεπεξεργασία των δεδομένων οι αλγόριθμοι με το μεγαλύτερο σκορ είναι ο αλγόριθμος παλινδρόμησης κορυφογραμμών και ο SVR σημειώνοντας ποσοστό ακρίβειας ίσο με 61%. Τη δεύτερη θέση καταλαμβάνουν ο KNN και η παλινδρόμηση Lasso με σκορ 59%. Ελάχιστα πιο πίσω με ποσοστό 58% βρίσκεται ο MLP και η γραμμική παλινδρόμηση. Στην τέταρτη θέση κατατάσσεται ο αλγόριθμος τυχαίων δασών με ποσοστό 53%. Την πέμπτη και τελευταία θέση λαμβάνει ο αλγόριθμος δέντρων απόφασης έχοντας μεγαλή διαφορά σε σχέση με τους υπόλοιπους αλγόριθμους με ποσοστό ακρίβειας 39%. Για την κλιμάκωση των στοιχείων αυτού συνόλου δεδομένων χρησιμοποιήθηκε ο Normalizer, ο οποίος ήταν και ο καλύτερος. Οι αλγόριθμοι που ευνοήθηκαν περισσότερο από την κλιμάκωση ήταν ο SVR, ο αλγόριθμος παλινδρόμησης κορυφογραμμών, παλινδρόμησης Lasso, και νευρωνικών δικτύων MLP.

Πίνακας 5.18: Αποτελέσματα συνόλου δεδομένων winequalityRed

Αλγόριθμος	**Σκορ πριν την ΠΕ**	**Σκορ μετά την ΠΕ**
SVR	18%	38%
Γραμμική παλ.	35%	36%
Παλ. κορυφογραμ.	30%	36%
Παλ. Lasso	0%	36%
Δέντρα απόφασης	9%	13%
Τυχαία Δάση	50%	44%
MLP	30%	38%
KNN	13%	40%

Ο Πίνακας 5.18 αποτυπώνει τα σκορ του συνόλου δεδομένων winequalityRed πριν και μετά την προεπεξεργασία των στοιχείων του. Ο αλγόριθμος που συγκέντρωσε το μεγαλύτερο ποσοστό ακρίβειας και βρίσκεται στην πρώτη θέση της κατάταξης με ποσοστό 44% είναι ο αλγόριθμος τυχαίων δασών. Αμέσως μετά από αυτόν στη δεύτερη θέση βρίσκεται ο αλγόριθμος των πλησιέστερων γειτόνων με σκορ ίσο του 40%. Την τρίτη θέση με σχεδόν αμελητέα διαφορά σε σχέση με τον αλγόριθμο KNN καταλαμβάνουν οι αλγόριθμοι νευρωνικών δικτύων και μηχανών διανυσμάτων υποστήριξης με ποσοστό ίσο του 38%. Στη συνέχεια, με ποσοστό 36% βρίσκονται στην τέταρτη θέση της κατάταξης οι αλγόριθμοι γραμμικής παλινδρόμησης, παλινδρόμησης κορυφογραμμών και παλινδρόμησης Lasso. Ο αλγόριθμος δέντρων απόφασης καταφέρνει να πετυχεί το μικρότερο ποσοστό ακρίβειας ίσο με

13% και βρίσκεται στην τελευταία θέση. Ο QuantileTransformer ήταν ο καλύτερος αλγόριθμος κλιμάκωσης σε αυτό το σύνολο δεδομένων και ευνόησε πιο πολύ τους αλγόριθμους SVR, παλινδρόμησης Lasso και τον KNN.

Πίνακας 5.19: Αποτελέσματα συνόλου δεδομένων diabetes

Αλγόριθμος	**Σκορ πριν την ΠΕ**	**Σκορ μετά την ΠΕ**
SVR	24%	28%
Γραμμική παλ.	20%	27%
Παλ. κορυφογραμ.	24%	31%
Παλ. Lasso	0%	31%
Δέντρα απόφασης	-29%	1%
Τυχαία Δάση	27%	27%
MLP	-108%	27%
KNN	11%	23%

Στον Πίνακα 5.19 αναπαριστώνται τα ποσοστά ακρίβειας του συνόλου δεδομένων diabetes πριν και μετά την προεπεξεργασία των χαρακτηριστικών του. Στο σύνολο δεδομένων diabetes οι αλγόριθμοι που συγκέντρωσαν το υψηλότερο σκορ κατά την πειραματική διαδικασία ήταν η παλινδρόμηση κορυφογραμμών και η παλινδρόμηση Lasso συγκεντρώνοντας ποσοστό 31%. Η δεύτερη θέση της κατάταξης ανήκει στον αλγόριθμο μηχανών διανυσμάτων υποστήριξης SVR με σκορ 28%. Στην τρίτη θέση με ποσοστό ακρίβειας 27% βρίσκονται οι αλγόριθμοι γραμμικής παλινδρόμησης, νευρωνικών δικτύων και τυχαίων δασών. Την τέταρτη θέση με σκορ 23% κατέχει ο αλγόριθμος πλησιέστερων γειτόνων. Την τελευταία θέση με αρκετά μικρότερο ποσοστό ίσο με 1% σε σχέση με τους υπόλοιπους καταλαμβάνει ο αλγόριθμος δέντρων απόφασης. Ο αλγόριθμος κλιμάκωσης που χρησιμοποιήθηκε με τον καλύτερο δυνατό τρόπο και είχε τα καλύτερα αποτελέσματα ήταν ο QuantileTransformer ευνοώντας τον αλγόριθμο παλινδρόμησης κορυφογραμμών και παλινδρόμησης Lasso και MLP.

Ο Πίνακας 5.20 απεικονίζει τα σκορ του συνόλου δεδομένων advertising πριν και μετά την προεπεξεργασία των στοιχείων του. Στην πρώτη θέση κατάφερε να κυριαρχήσει ο αλγόριθμος γραμμικής παλινδρόμησης με σκορ 82%. Ελάχιστα πιο πίσω του και στη δεύτερη θέση βρίσκονται οι αλγόριθμοι παλινδρόμησης κορυφογραμμών, παλινδρόμησης Lasso και Κ πλησιέστερων γειτόνων με ποσοστό ακρίβειας ίσο με 80%. Με διαφορά 2% πίσω από τους αλγόριθμους της δεύτερης θέσης βρίσκεται ο MLP με σκορ 78%. Στην τέταρτη θέση της κατάταξης με ποσοστό 76% βρίσκεται ο

Πίνακας 5.20: Αποτελέσματα συνόλου δεδομένων advertising

Αλγόριθμος	**Σκορ πριν την ΠΕ**	**Σκορ μετά την ΠΕ**
SVR	87%	75%
Γραμμική παλ.	77%	82%
Παλ. κορυφογραμ.	67%	80%
Παλ. Lasso	-4%	80%
Δέντρα απόφασης	90%	69%
Τυχαία Δάση	95%	76%
MLP	-77%	78%
KNN	90%	80%

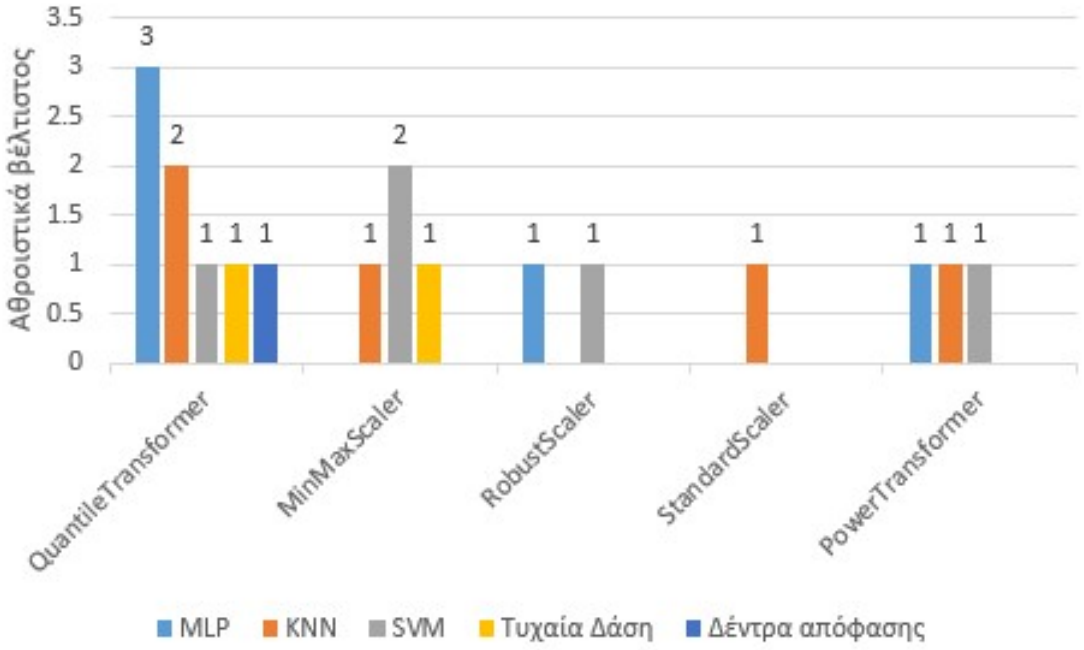
αλγόριθμος τυχαίων δασών. Με σχεδόν αμελητέα διαφορά στην πέμπτη θέση κατατάσσεται ο αλγόριθμος μηχανών διανυσμάτων υποστήριξης με ποσοστό 75%. Τέλος, πίσω του και στην έκτη θέση με σκορ 69% βρίσκεται ο αλγόριθμος δέντρων απόφασης. Ο αλγόριθμος QuantileTransformer ήταν ο καλύτερος αλγόριθμος κλιμάκωσης των δειγμάτων στο συγκεκριμένο σύνολο δεδομένων. Οι αλγόριθμοι που ευνοήθηκαν περισσότερο από την κλιμάκωση ήταν ο αλγόριθμος παλινδρόμησης Lasso και νευρωνικών δικτύων MLP.

5.8 Συγκεντρωτική σύγκριση αποτελεσμάτων

Στα Σχήματα 5.1 και 5.2 ο άξονας X αναπαριστά τους αλγόριθμους μηχανικής μάθησης που συνεργάστηκαν με κάποιον αλγόριθμο κλιμάκωσης ενώ ο άξονας Y με τον τίτλο αθροιστικά καλύτερος αναπαριστά πόσες φορές ο συνδυασμός συγκεκριμένου αλγορίθμου μηχανικής μάθησης και κλιμάκωσης ήταν ο καλύτερος.

Μετά την εκτέλεση της πειραματικής διαδικασίας σε όλα τα σύνολα δεδομένων κατηγοριοποίησης διαπιστώθηκε πως ο αλγόριθμος με την καλύτερη απόδοση σε συνδυασμό με τον αλγόριθμο κλιμάκωσης QuantileTransformer ήταν ο MLP πετυχαίνοντας το υψηλότερο σκορ σε 3 από τα 10 σύνολα δεδομένων. Στη δεύτερη θέση της κατάταξης βρίσκεται ο αλγόριθμος Κ πλησιέστερων γειτόνων σε συνδυασμό με τον QuantileTransformer και ο SVM σε συνδυασμό με τον αλγόριθμο κλιμάκωσης MinMaxScaler επειδή είχαν το καλύτερο ποσοστό ακρίβειας σε 2 από τα 10 σύνολα δεδομένων. Οι συνδυασμοί SVM με RobustScaler, MLP με RobustScaler, SVM με QuantileTransformer, τυχαία δάση με MinMaxScaler, KNN με MinMaxScaler, KNN με StandardScaler, τυχαία δάση με QuantileTransformer, KNN με QuantileTransformer,

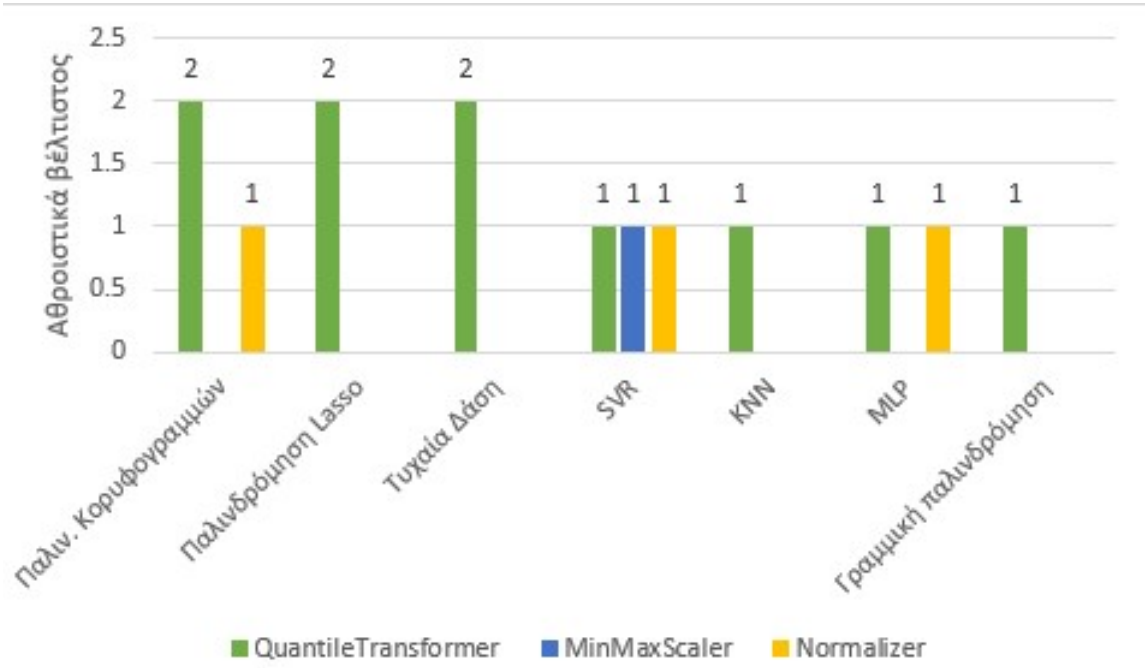
Σχήμα 5.1: Συγκεντρωτικά αποτελέσματα αλγορίθμων κατηγοριοποίησης



δέντρα απόφασης με QuantileTransformer, SVM με PowerTransformer, MLP με PowerTransformer και KNN με PowerTransformer, βρίσκονται στην τρίτη και τελευταία θέση της κατάταξης με τις καλύτερες επιδόσεις σε 1 από τα 10 σύνολα δεδομένων. Επιπλέον ο αλγόριθμος που παρουσίασε τις περισσότερες φορές βελτίωση στην απόδοση του εξαιτίας της κλιμάκωσης των δειγμάτων στα σύνολα δεδομένων ήταν ο αλγόριθμος τυχαίων δασών. Τέλος, ο αλγόριθμος αντιμετώπισης ελλιπών τιμών που εξήγαγε τις περισσότερες φορές στα σύνολα δεδομένων κατηγοριοποίησης τα μεγαλύτερα ποσοστά ακρίβειας ήταν ο SimpleImputer.

Αφού ολοκληρώθηκε η πειραματική διαδικασία σε όλα τα σύνολα δεδομένων παλινδρόμησης παρατηρείται πως ο αλγόριθμος παλινδρόμησης κορυφογραμμών, ο αλγόριθμος παλινδρόμησης Lasso και ο αλγόριθμος τυχαίων δασών σε συνδυασμό με τον αλγόριθμο κλιμάκωσης QuantileTransformer είχαν την καλύτερη απόδοση σε 2 από τα 10 σύνολα δεδομένων. Στη δεύτερη και τελευταία θέση της κατάταξης βρίσκονται οι συνδυασμοί SVR με MinMaxScaler, SVR με QuantileTransformer, K πλησιέστεροι γείτονες με QuantileTransformer, ο MLP με αλγόριθμο κλιμάκωσης QuantileTransformer στη μια περίπτωση και Normalizer στην άλλη, ο SVR με Normalizer, ο αλγόριθμος παλινδρόμησης κορυφογραμμών με Normalizer και η γραμμική παλινδρόμηση με τον QuantileTransformer σημειώνοντας την καλύτερη

Σχήμα 5.2: Συγκεντρωτικά αποτελέσματα αλγορίθμων παλινδρόμησης



απόδοση σε μόνο 1 από τα 10 σύνολα δεδομένων. Ο αλγόριθμος αντιμετώπισης ελλιπών τιμών που εξήγαγε τις περισσότερες φορές στα σύνολα δεδομένων παλινδρόμησης τα υψηλότερα σκορ ήταν ο KNNImputer επειδή συνεργάζεται καλύτερα με δεδομένα που είναι συνεχή. Επίσης, οι αλγόριθμοι μηχανικής μάθησης που παρουσίασαν τις πιο πολλές φορές βελτίωση της απόδοσης τους εξαιτίας της κλιμάκωσης ήταν ο MLP και ο SVR.

Κεφάλαιο 6

Συμπεράσματα

Στην παρούσα διπλωματική εργασία αναλύθηκαν οι έννοιες της μηχανικής μάθησης, οι μέθοδοι εφαρμογής (μη εποπτευόμενη μάθηση, εποπτευόμενη μάθηση και η μαζική μάθηση κ.ο.κ), προκλήσεις της προεπεξεργασίας δεδομένων (ελλιπή δεδομένα, κωδικοποίηση χαρακτηριστικών, επιλογή κατάλληλων χαρακτηριστικών) και πραγματοποιήθηκαν δύο πειραματικές διαδικασίες. Στις πειραματικές διαδικασίες χρησιμοποιήθηκαν πέντε συνολικά αλγόριθμοι κατηγοριοποίησης (αλγόριθμος υποστήριξης διανυσμάτων μηχανών, κατηγοριοποιητής πλησιέστερων γειτώνων, κατηγοριοποιητής δέντρων απόφασης, κατηγοριοποιητής MLP, κατηγοριοποιητής τυχαίων δασών) και οκτώ αλγόριθμοι παλινδρόμησης (αλγόριθμος γραμμικής παλινδρόμησης, παλινδρόμησης Lasso, παλινδρόμησης τυχαίων δασών, παλινδρόμησης MLP, αλγόριθμος παλινδρόμησης πλησιέστερων γειτόνων, παλινδρόμησης κορυφογραμμών, αλγόριθμος υποστήριξης διανυσμάτων μηχανών, παλινδρόμησης δέντρων απόφασης). Επιπλέον στα πειράματα συμμετείχαν έξι αλγόριθμοι κλιμάκωσης (PowerTransformer, MinMaxScaler, RobustScaler, StandardScaler, QuantileTransformer, Normalizer) και δύο αλγόριθμοι αντιμετώπισης ελλιπών τιμών (KNNImputer, SimpleImputer). Στόχοι αποτελούσαν η εύρεση ενός ή πιο πολλών αλγορίθμων μηχανικής μάθησης ως τους καλύτερους σύμφωνα με το ποσοστό ακρίβειας που συγκέντρωναν καθώς επίσης και η εύρεση των αλγορίθμων που ευνοήθηκαν πιο πολύ σε σχέση με τους υπόλοιπους αφού πρώτα πραγματοποιήθηκε η προπεξεργασία δεδομένων και η συνεισφορά κάποιου αλγόριθμου κλιμάκωσης. Ακόμη ένας στόχος ήταν η εύρεση των αλγορίθμων όπου σε συνδυασμό με έναν αλγόριθμο κλιμάκωσης ήταν οι καλύτεροι συγκεντρωτικά πετυχαίνοντας τα υψηλότερα σκορ σε πολλαπλά σύνολα δεδομένων τόσο σε ζητούμενα κατηγοριο-

ποίησης όσο και παλινδρόμησης. Η υλοποίηση των πειραμάτων επιτεύχθηκε αφού πρώτα δόθηκαν ως είσοδοι δέκα σύνολα δεδομένων κατηγοριοποίησης και δέκα σύνολα δεδομένων παλινδρόμησης τα οποία συγκεντρώθηκαν από τις ιστοσελίδες UCI Machine Learning Repository και Kaggle. Προκειμένου όμως να μην υπάρξουν ανακρίβειες και λανθασμένες εκτιμήσεις από τα όλα τα σύνολα δεδομένων αφαιρέθηκαν οι ελλιπείς τιμές και κωδικοποιήθηκαν τα χαρακτηριστικά από κατηγορηματικά σε αριθμητικά. Τα αποτελέσματα έδειξαν πως ο αλγόριθμος MLP σε συνδυασμό με το αλγόριθμο κλίμάκωσης QuantileTransformer ήταν ο καλύτερος συγκεντρώνοντας το υψηλότερο ποσοστό ακρίβειας σε 3 από τα 10 σύνολα δεδομένων κατηγοριοποίησης. Οι καλύτεροι αλγόριθμοι μηχανικής μάθησης για τα σύνολα δεδομένων παλινδρόμησης ήταν ο αλγόριθμος παλινδρόμησης Lasso, ο αλγόριθμος παλινδρόμησης κορυφογραμμών και ο αλγόριθμος τυχαίων δασών σε συνδυασμό με τον αλγόριθμο κλιμάκωσης QuantileTransformer επειδή είχαν το μεγαλύτερο σκορ σε 2 από τα 10 σύνολα δεδομένων. Ο συνδυασμός μηχανών διανυσμάτων υποστήριξης με QuantileTransformer συγκεντρώνουν το χαμηλότερο ποσοστό ακρίβειας ταυτόχρονα και στις δύο πειραματικές διαδικασίες. Συνδυασμοί όπως ο αλγόριθμος MLP με PowerTransformer, K πλησιέστεροι γείτονες με MinMaxScaler πετυχαίνουν το χαμηλότερο σκορ στο πείραμα της κατηγοριοποίησης, ενώ συνδυασμοί όπως ο αλγόριθμος MLP με QuantileTransformer στο πείραμα της παλινδρόμησης δεν καταφέρνει να έχει αντίστοιχες επιδόσεις όπως στα σύνολα κατηγοριοποίησης και βρίσκεται ανάμεσα στις χειρότερες περιπτώσεις. Το ίδιο φαινόμενο παρατηρείται και για τον συνδυασμό τυχαίων δασών με QuantileTransformer που υπερτερεί σε σύνολα δεδομένων παλινδρόμησης και όχι σε κατηγοριοποίησης. Επιπρόσθετα ο αλγόριθμος αντιμετώπισης ελλιπών στοιχείων που υπερίσχυσε στο πείραμα της παλινδρόμησης ήταν ο KNNImputer ενώ στο πείραμα της κατηγοριοποίησης ο SimpleImputer. Τέλος, οι αλγόριθμοι που ευνοήθηκαν σε μεγαλύτερο βαθμό σε σχέση με τους υπόλοιπους εξαιτίας της κλιμάκωσης των δεδομένων ήταν ο αλγόριθμος τυχαίων δασών για τα σύνολα της κατηγοριοποίησης και ο αλγόριθμος μηχανών διανυσμάτων υποστήριξης για τα σύνολα της παλινδρόμησης.

References

API Reference — scikit-learn 0.24.1 documentation. (2021, Feb). Retrieved from https://scikit-learn.org/stable/modules/classes.html#module-sklearn.preprocessing

Ashish. (2021, Feb). *Advertising Dataset*. Retrieved from https://www.kaggle.com/ashydv/advertising-dataset

Batista, G. E., & Monard, M. C. (2003). An analysis of four missing data treatment methods for supervised learning. *Applied Artificial Intelligence*, *17*(5-6), 519–533. Retrieved from https://d1wqtxts1xzle7.cloudfront.net/44690676/GAsRULE_for_Knowledge_Discovery20160413-15866-74bhns.pdf?1460541356=&response-content-disposition=inline%3B+filename%3DGa_s_rule_for_knowledge_discovery.pdf&Expires=1613337207&Signature=Lktv4SyhFADzo6jYOg-UlsU3At0UCJFJBDvbMFtyBFU~H84Kr8IJtmVV3uPOHFiX2SMfOgqSFIaEIfUOYk0V5NE2CZh1vOgW-fgzjDJD1LWpr8jeUdVhRQ1wTIe8mp0Nz1MBeIgG5WUo6u1yJva9opAA63NF5iSrhnXwyuM0rnxmMW-CIobVt6emsWemzyBdd-7ssAeUeOghuOUxf0bG7eyMzjHTA__&Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA#page=150

Batista, G. E., Monard, M. C., et al. (2002). A study of k-nearest neighbour as an imputation method. *His*, *87*(251-260), 48. Retrieved from https://www.researchgate.net/profile/Maria-Carolina_Monard/publication/2475229_A_Study_of_K-Nearest_Neighbour_as_an_Imputation_Method/links/0deec51ae1802c9861000000.pdf

Biessmann, F., Rukat, T., Schmidt, P., Naidu, P., Schelter, S., Taptunov, A., ... Salinas, D. (2019). Datawig: Missing value imputation for tables. *Journal of Machine Learning Research*, *20*(175), 1–6. Retrieved from https://www.jmlr.org/papers/volume20/18-753/18-753.pdf

Bora, B. (2021, Feb). *On-Demand Webinar – AI Development Using Data Science VMs (DSVM), Deep Learning VMs (DLVM) & Azure Batch AI.* Retrieved from https://docs.microsoft.com/el-gr/archive/blogs/machinelearning/on-demand-webinar-ai-development-using-data-science-vms-dsvm-deep-learning-vms-dlvm-azure-batch-ai

Brownlee, J. (2020α′, Aug). *3 Ways to Encode Categorical Variables for Deep Learning.* Retrieved from https://machinelearningmastery.com/how-to-prepare-categorical-data-for-deep-learning-in-python

Brownlee, J. (2020β′, Aug). *How to use Data Scaling Improve Deep Learning Model Stability and Performance.* Retrieved from https://machinelearningmastery.com/how-to-improve-neural-network-stability-and-modeling-performance-with-data-scaling

Brownlee, J. (2020γ′, Aug). *Linear Regression for Machine Learning.* Retrieved from https://machinelearningmastery.com/linear-regression-for-machine-learning

Bruce. (2021, Feb). Real estate price prediction. *UCI Machine Learning Repository*. Retrieved from https://www.kaggle.com/quantbruce/real-estate-price-prediction

Cerda, P., & Varoquaux, G. (2020). Encoding high-cardinality string categorical variables. *IEEE Transactions on Knowledge and Data Engineering*. Retrieved from https://arxiv.org/pdf/1907.01860.pdf

Chapter 9 Rubin's Rules. (2020, Sep). *Rubin's Rules*. Retrieved from https://bookdown.org/mwheymans/bookmi/rubins-rules.html

Choi, M. (2021, Feb). Medical Cost Personal Datasets. *UCI Machine Learning Repository*. Retrieved from https://www.kaggle.com/mirichoi0218/insurance

Contributors to Wikimedia projects. (2021, Feb). *scikit-learn - Wikipedia.* Retrieved from https://en.wikipedia.org/w/index.php?title=Scikit-learn&oldid=1005671708

DataWig documentation. (2020, Jul). Retrieved from https://datawig.readthedocs.io/en/latest

Distributed model training using Dask and Scikit-learn - Datafoam. (2020, Mar). Retrieved from https://datafoam.com/2020/03/17/distributed-model-training-using

-dask-and-scikit-learn

Fast Threshold Clustering Algorithm. (2013, Nov). Retrieved from https://cssanalytics.wordpress.com/2013/11/26/fast-threshold-clustering-algorithm-ftca

Géron, A. (2019). *Hands-on machine learning with scikit-learn, keras, and tensorflow: Concepts, tools, and techniques to build intelligent systems*. O'Reilly Media. Retrieved from https://books.google.gr/books?hl=en&lr=&id=HHetDwAAQBAJ&oi=fnd&pg=PP1&dq=Hands-on+machine+learning+with+Scikit-Learn,+Keras,+and+TensorFlow:+Concepts,+tools,+and+techniques+to+build+intelligent+systems&ots=0Loh0pfgOs&sig=2KyMl1jc1iWNvusf9VZD5Un1ch8&redir_esc=y#v=onepage&q=Hands-on%20machine%20learning%20with%20Scikit-Learn%2C%20Keras%2C%20and%20TensorFlow%3A%20Concepts%2C%20tools%2C%20and%20techniques%20to%20build%20intelligent%20systems&f=false

Goyal, S. (2021, Feb). Car Data. *Kaggle*. Retrieved from https://www.kaggle.com/goyalshalini93/car-data

Halevy, A., Norvig, P., & Pereira, F. (2009). The unreasonable effectiveness of data. *IEEE Intelligent Systems*, *24*(2), 8–12. Retrieved from https://storage.googleapis.com/pub-tools-public-publication-data/pdf/35179.pdf

Hauck, T. (2014, Nov). Imputing missing values through various strategies - scikit-learn Cookbook. *Imputing missing values through various strategies*. Retrieved from https://subscription.packtpub.com/book/big_data_and_business_intelligence/9781783989485/1/ch01lvl1sec16/imputing-missing-values-through-various-strategies

Hirethanad, J. R. (2021, Feb). *drug200.csv.* Retrieved from https://www.kaggle.com/jeevanrh/drug200csv

Htoon, K. S. (2020, Jul). A Guide To KNN Imputation - Kyaw Saw Htoon - Medium. *Medium*. Retrieved from https://medium.com/@kyawsawhtoon/a-guide-to-knn-imputation-95e2dc496e

Idris, I. (2015). *NumPy: Beginner's Guide*. Packt Publishing. Retrieved from https://books.google.gr/books?id=m2T9CQAAQBAJ&printsec=frontcover&dq=NUMPY&hl=en&sa=X&ved=

2ahUKEwilrbDQleruAhURTcAKHWRVDf0QuwUwAHoECAUQBw#v=onepage&q=NUMPY&f=false

Kaggle: Your Home for Data Science. (2021, Feb). Retrieved from https://www.kaggle.com

Khalid, S., Khalil, T., & Nasreen, S. (2014). A survey of feature selection and feature extraction techniques in machine learning. In *2014 science and information conference* (pp. 372–378). Retrieved from https://www.researchgate.net/profile/Shamila_Nasreen/publication/265727419_A_Survey_Of_Feature_Selection_And_Feature_Extraction_Techniques_In_Machine_LearningSAI2014/links/55b12f2c08aec0e5f4310e76.pdf

Learning, U. M. (2021, Feb). *Pima Indians Diabetes Database*. Retrieved from https://www.kaggle.com/uciml/pima-indians-diabetes-database

Mucherino, A., Papajorgji, P. J., & Pardalos, P. M. (2009, Sep). k-Nearest Neighbor Classification. *SpringerLink*, 83–106. Retrieved from https://link.springer.com/chapter/10.1007/978-0-387-88615-2_4

Müller, A. C., & Guido, S. (2016). *Introduction to machine learning with python: a guide for data scientists*. Retrieved from https://books.google.gr/books?hl=en&lr=&id=1-4lDQAAQBAJ&oi=fnd&pg=PP1&dq=Introduction+to+machine+learning+with+Python:+a+guide+for+data+scientists&ots=28hPGJIFW_&sig=FdkdTt0PIQyiVJJM4xBVD9XNrQY&redir_esc=y#v=onepage&q=Introduction%20to%20machine%20learning%20with%20Python%3A%20a%20guide%20for%20data%20scientists&f=false

Navlani, A. (2021, Feb). *Decision Tree Classification in Python*. Retrieved from https://www.datacamp.com/community/tutorials/decision-tree-classification-python

Noriega, L. (2005). Multilayer perceptron tutorial. *School of Computing. Staffordshire University*. Retrieved from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.608.2530&rep=rep1&type=pdf

Panda, M., & Misra, H. (2021). *Handbook of Research on Automated Feature Engineering and Advanced Applications in Data Science*. IGI Global. Retrieved from https://books.google.gr/books?id=6dIPEAAAQBAJ&pg=PA9&dq=Categorical+encoding#v=onepage&q=Categorical%20encoding&f=false

Pavlov, Y. L. (2019). *Random Forests*. Berlin, Germany: De Gruyter. Retrieved from https://books.google.gr/books?id=HBmBDwAAQBAJ&printsec=frontcover&dq=random+forest&hl=en&sa=X&ved=2ahUKEwjE2a2HmeruAhVwQkEAHe5MDY4Q6AEwAHoECAYQAg#v=onepage&q=random%20forest&f=false

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... others (2011). Scikit-learn: Machine learning in python. *The Journal of Machine Learning Research*, *12*, 2825–2830. Retrieved from https://www.jmlr.org/papers/volume12/pedregosa11a/pedregosa11a.pdf

Ridge Regression: Simple Definition - Statistics How To. (2021, Feb). Retrieved from https://www.statisticshowto.com/ridge-regression

Saalu, S. (2021, Feb). *Amazon Top 50 Bestselling Books 2009 - 2019*. Retrieved from https://www.kaggle.com/sootersaalu/amazon-top-50-bestselling-books-2009-2019

Seber, G. A., & Lee, A. J. (2012). *Linear regression analysis* (Vol. 329). John Wiley & Sons. Retrieved from https://books.google.gr/books?hl=en&lr=&id=X2Y6OkXl8ysC&oi=fnd&pg=PR5&dq=Linear+regression+&ots=seiQB_qOmv&sig=GMNYYDNEzGnHQtNXxcduurA7Fag&redir_esc=y#v=onepage&q=Linear%20regression&f=false

sklearn.impute.SimpleImputer — scikit-learn 0.24.1 documentation. (2021, Feb). Retrieved from https://scikit-learn.org/stable/modules/generated/sklearn.impute.SimpleImputer.html

Sterne, J. A. C., White, I. R., Carlin, J. B., Spratt, M., Royston, P., Kenward, M. G., ... Carpenter, J. R. (2009, Jun). Multiple imputation for missing data in epidemiological and clinical research: potential and pitfalls. *BMJ*, *338*, b2393. Retrieved from https://www.bmj.com/content/338/bmj.b2393/

Team, S. (2021, Feb). *Home — Spyder IDE*. Retrieved from https://www.spyder-ide.org

UCI Machine Learning Repository. (2021, Feb). Retrieved from https://archive.ics.uci.edu/ml/index.php

UCI Machine Learning Repository: Abalone Data Set. (2021, Feb). Retrieved from https://archive.ics.uci.edu/ml/datasets/abalone

UCI Machine Learning Repository: Balance Scale Data Set. (2021, Feb). Retrieved from https://archive.ics.uci.edu/ml/datasets/balance+scale

UCI Machine Learning Repository: Blood Transfusion Service Center Data Set. (2021, Feb). Retrieved from https://archive.ics.uci.edu/ml/datasets/Blood+Transfusion+Service+Center

UCI Machine Learning Repository: Car Evaluation Data Set. (2021, Feb). Retrieved from https://archive.ics.uci.edu/ml/datasets/car+evaluation

UCI Machine Learning Repository: Concrete Slump Test Data Set. (2021, Feb). Retrieved from http://archive.ics.uci.edu/ml/datasets/concrete+slump+test

UCI Machine Learning Repository: Contraceptive Method Choice Data Set. (2021, Feb). Retrieved from https://archive.ics.uci.edu/ml/datasets/Contraceptive+Method+Choice

UCI Machine Learning Repository: Diabetes Data Set. (2021, Feb). Retrieved from https://archive.ics.uci.edu/ml/datasets/diabetes

UCI Machine Learning Repository: DrivFace Data Set. (2021, Feb). Retrieved from https://archive.ics.uci.edu/ml/datasets/DrivFace

UCI Machine Learning Repository: Haberman's Survival Data Set. (2021, Feb). Retrieved from https://archive.ics.uci.edu/ml/datasets/Haberman's+Survival

UCI Machine Learning Repository: Hayes-Roth Data Set. (2021, Feb). Retrieved from https://archive.ics.uci.edu/ml/datasets/Hayes-Roth

UCI Machine Learning Repository: Lenses Data Set. (2021, Feb). Retrieved from https://archive.ics.uci.edu/ml/datasets/Lenses

UCI Machine Learning Repository: Liver Disorders Data Set. (2021, Feb). Retrieved from https://archive.ics.uci.edu/ml/datasets/Liver+Disorders

UCI Machine Learning Repository: QSAR aquatic toxicity Data Set. (2021, Feb). Retrieved from https://archive.ics.uci.edu/ml/datasets/QSAR+aquatic+toxicity

UCI Machine Learning Repository: South German Credit Data Set. (2021, Feb). Retrieved from https://archive.ics.uci.edu/ml/datasets/South+German+Credit

UCI Machine Learning Repository: Tic-Tac-Toe Endgame Data Set. (2021, Feb). Retrieved from https://archive.ics.uci.edu/ml/datasets/Tic-Tac-Toe+Endgame

UCI Machine Learning Repository: Wine Quality Data Set. (2021, Feb). Retrieved from https://archive.ics.uci.edu/ml/datasets/wine+quality

Vafaie, H., & De Jong, K. A. (1992). Genetic algorithms as a tool for feature selection in machine learning. In *Ictai* (pp. 200–203). Retrieved from https://www.researchgate.net/profile/Kenneth_De_Jong/publication/2722353_Genetic_Algorithms_as_a_Tool_for_Feature_Selection_in_Machine_Learning/links/09e41510956a7d0528000000.pdf

VanderPlas, J. (2016). *Python Data Science Handbook*. Sebastopol, CA, USA: O'Reilly Media. Retrieved from https://books.google.gr/books?id=6omNDQAAQBAJ&printsec=frontcover&dq=Pandas+machine+learning&hl=en&sa=X&ved=2ahUKEwjo9qLLmeruAhWJEWMBHTyZDKAQ6AEwAXoECAQQAg#v=onepage&q=Pandas%20machine%20learning&f=false

Van Rossum, G., Drake, F. L., et al. (2000). *Python reference manual*. iUniverse Indiana. Retrieved from http://ft-sipil.unila.ac.id/dbooks/Python%20Reference%20Manual.pdf

Van Rossum, G., et al. (2021, Feb). Python (programming language).. Retrieved from https://thereaderwiki.com/en/Python_(programming_language)

Vikram, V. (2020, Jul). Why feature scaling is important? *Kaggle*. Retrieved from https://www.kaggle.com/vin1234/why-feature-scaling-is-important/notebook

Wang, L. (2005). *Support vector machines: theory and applications* (Vol. 177). Springer Science & Business Media. Retrieved from https://books.google.gr/books?hl=en&lr=&id=uTzMPJjVjsMC&oi=fnd&pg=PA1&dq=Support+Vector+Machine&ots=GFAG9r1Io6&sig=eidB02fnX_QaX8JcRcoBOlmA5fQ&redir_esc=y#v=onepage&q=Support%20Vector%20Machine&f=false

Weisberg, S. (2001). Yeo-johnson power transformations. *Department of Applied Statistics, University of Minnesota. Retrieved June*, *1*, 2003. Retrieved from https://www.stat.umn.edu/arc/yjpower.pdf

What is LASSO Regression Definition, Examples and Techniques. (2020, Dec). Retrieved from https://www.mygreatlearning.com/blog/understanding-of-lasso-regression

Zheng, A., & Casari, A. (2018). *Feature engineering for machine learning: principles and techniques for data scientists*. " O'Reilly Media, Inc.". Retrieved from https://books.google.gr/books?id=

sthSDwAAQBAJ&printsec=frontcover&dq=Feature+engineering+for+machine+learning:+principles+and+techniques+for+data+scientists&hl=en&sa=X&ved=2ahUKEwjc3e65luruAhUD6RoKHak0AecQ6AEwAHoECAUQAg#v=onepage&q=Feature%20engineering%20for%20machine%20learning%3A%20principles%20and%20techniques%20for%20data%20scientists&f=false