Εθνικο Μετσοβιο Πολυτεχνειο
Σχολη Ηλεκτρολογων Μηχανικων και Μηχανικων Υπολογιστων
Τομεας Τεχνολογιας Πληροφορικης και Υπολογιστων
ΔΠΜΣ Επιστημη Δεδομενων και Μηχανικη Μαθηση

Πρόβλεψη αποτελέσματος αγώνα NBA με τη χρήση Μηχανικής Μάθησης

Μελέτη και υλοποίηση

Μεταπτυχιακη Εργασια

του

ΚΕΤΣΕΑ Θ. ΣΤΑΥΡΟΥ

Επιβλέπων: Γεώργιος Στάμου, Καθηγητής ΕΜΠ

Συνεπιβλέπουσα: Παρασκευή Τζούβελη, ΕΔΙΠ ΕΜΠ

Αθήνα, Νοέμβριος 2022

Εθνικο Μετσοβιο Πολυτεχνειο
Σχολη Ηλεκτρολογων Μηχανικων και Μηχανικων Υπολογιστων
Τομεας Τεχνολογιας Πληροφορικης και Υπολογιστων
ΔΠΜΣ Επιστημη Δεδομενων και Μηχανικη Μαθηση

Πρόβλεψη αποτελέσματος αγώνα NBA με τη χρήση Μηχανικής Μάθησης

Μελέτη και υλοποίηση

ΜΕΤΑΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

του

ΚΕΤΣΕΑ Θ. ΣΤΑΥΡΟΥ

Επιβλέπων: Γεώργιος Στάμου
Καθηγητής ΕΜΠ

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 18η Νοεμβρίου 2022.

(Υπογραφή)	*(Υπογραφή)*	*(Υπογραφή)*
.....	.....	.....
Γεώργιος Στάμου	Στέφανος Κόλλιας	Αθανάσιος Βουλόδημος
Καθηγητής ΕΜΠ	Καθηγητής ΕΜΠ	Επίκουρος Καθηγητής ΕΜΠ

Αθήνα, Νοέμβριος 2022

Εθνικο Μετσοβιο Πολυτεχνειο
Σχολη Ηλεκτρολογων Μηχανικων και Μηχανικων Υπολογιστων
Τομεας Τεχνολογιας Πληροφορικης και Υπολογιστων
ΔΠΜΣ Επιστημη Δεδομενων και Μηχανικη Μαθηση

ΔΗΛΩΣΗ ΜΗ ΛΟΓΟΚΛΟΠΗΣ ΚΑΙ ΑΝΑΛΗΨΗΣ ΠΡΟΣΩΠΙΚΗΣ ΕΥΘΥΝΗΣ

Με πλήρη επίγνωση των συνεπειών του νόμου περί πνευματικών δικαιωμάτων, δηλώνω ενυπογράφως ότι είμαι αποκλειστικός συγγραφέας της παρούσας Πτυχιακής Εργασίας, για την ολοκλήρωση της οποίας κάθε βοήθεια είναι πλήρως αναγνωρισμένη και αναφέρεται λεπτομερώς στην εργασία αυτή. Έχω αναφέρει πλήρως και με σαφείς αναφορές, όλες τις πηγές χρήσης δεδομένων, απόψεων, θέσεων και προτάσεων, ιδεών και λεκτικών αναφορών, είτε κατά κυριολεξία είτε βάσει επιστημονικής παράφρασης. Αναλαμβάνω την προσωπική και ατομική ευθύνη ότι σε περίπτωση αποτυχίας στην υλοποίηση των ανωτέρω δηλωθέντων στοιχείων, είμαι υπόλογος έναντι λογοκλοπής, γεγονός που σημαίνει αποτυχία στην Πτυχιακή μου Εργασία και κατά συνέπεια αποτυχία απόκτησης του Τίτλου Σπουδών, πέραν των λοιπών συνεπειών του νόμου περί πνευματικών δικαιωμάτων. Δηλώνω, συνεπώς, ότι αυτή η Πτυχιακή Εργασία προετοιμάστηκε και ολοκληρώθηκε από εμένα προσωπικά και αποκλειστικά και ότι, αναλαμβάνω πλήρως όλες τις συνέπειες του νόμου στην περίπτωση κατά την οποία αποδειχθεί, διαχρονικά, ότι η εργασία αυτή ή τμήμα της δεν μου ανήκει διότι είναι προϊόν λογοκλοπής άλλης πνευματικής ιδιοκτησίας.

(Υπογραφή)

.....
Σταύρος Κετσέας

18 Νοεμβρίου 2022

Περίληψη

Η Μηχανική Μάθηση αποτελεί ένα σύγχρονο τεχνολογικό εργαλείο επεξεργασίας στοιχείων, με στόχο την πρόβλεψη αξιόπιστων αποτελεσμάτων, βασισμένη σε ιστορικά γεγονότα. Η κατάλληλη χρήση της, δύναται να εξοπλίσει τη φαρέτρα όλων των εμπλεκομένων που απαρτίζουν το άθλημα της καλαθοσφαίρισης, ξεκινώντας από τους παίκτες και τους προπονητές, έως και τους χορηγούς, διευθυντές, αλλά και τους ίδιους τους φιλάθλους, καθώς ενισχύει την προβλεπτική τους ικανότητα, η οποία κάλλιστα θα μπορούσε να εφαρμοστεί και στον στοιχηματισμό. Το γεγονός αυτό αποτελεί, συνεπώς, πόλο έλξης του ενδιαφέροντος πάσης φύσεως εταιρείας στοιχηματισμού αθλητικών γεγονότων.

Στην παρούσα διπλωματική εργασία οι αλγόριθμοι Μηχανικής Μάθησης που θα εφαρμοστούν είναι οι Logistic Regression, K Neighbors Classifier, Naive Bayes, Decision Tree Classifier, SVM - Linear Kernel, SVM - Radial Kernel, Gaussian Process Classifier, MLP Classifier, Ridge Classifier, Random Forest Classifier, Quadratic Discriminant Analysis, Ada Boost Classifier, Gradient Boosting Classifier, Linear Discriminant Analysis, Extra Trees Classifier, Extreme Gradient Boosting, Light Gradient Boosting Machine και ο CatBoost Classifier.

Το πρόβλημα που χρήζει επίλυσης, είναι η δυαδική ταξινόμηση του αποτελέσματος του αγώνα σε νίκη ή ήττα για τη γηπεδούχο ομάδα ή αντίστοιχα για τη φιλοξενούμενη. Η επίλυση αυτή είναι απότοκο της κατάλληλης διαχείρισης και επεξεργασίας των ιστορικών δεδομένων, παρελθοντικών αγώνων του NBA, με στόχο την ανακάλυψη εκείνων των χαρακτηριστικών ή ακόμη και την κατασκευή νέων, τα οποία θα βελτιστοποιήσουν στο μέγιστο δυνατό το επιθυμητό αποτέλεσμα. Η μετρική ως προς την οποία θα κριθεί η αποτελεσματικότητα της παρούσας εργασίας είναι η ορθότητα ή accuracy. Στην προσπάθεια αυτή επέκτασης της εργασίας, θα δοκιμαστούν και δύο μέθοδοι βαθιάς μηχανικής μάθησης, τα LSMT και TCN, καθώς και ένα τεχνητό νευρωνικό δίκτυο ANN. Τέλος, η εργασία αυτή πιθανότατα να αποτελέσει πηγή έμπνευσης σε μελλοντικούς ερευνητές, για πιθανή εφαρμογή της, τόσο σε νέα δεδομένα αγώνων NBA, όσο και σε διαφορετικά αθλήματα, αλλά και για περιθώρια βελτίωσής της.

Λέξεις Κλειδιά

Μηχανική Μάθηση, Πρόβλεψη, Βαθιά Μάθηση, Δυαδική Ταξινόμηση, Βελτιστοποίηση, Ορθότητα, NBA

Abstract

Machine Learning is a modern technological data processing tool, with the aim of predicting reliable results, based on historical events. Its proper use can equip the quiver of everyone involved with basketball, starting from the players and coaches, up to the sponsors, managers, but also the fans themselves, as it strengthens their predictive ability, which could be well applied to betting as well. As a result, this fact has lead to interest attraction of various sport betting companies.

In this thesis, the Machine Learning algorithms that will be applied are Logistic Regression, K Neighbors Classifier, Naive Bayes, Decision Tree Classifier, SVM - Linear Kernel, SVM - Radial Kernel, Gaussian Process Classifier, MLP Classifier, Ridge Classifier, Random Forest Classifier, Quadratic Discriminant Analysis, Ada Boost Classifier, Gradient Boosting Classifier, Linear Discriminant Analysis, Extra Trees Classifier, Extreme Gradient Boosting, Light Gradient Boosting Machine and the CatBoost Classifier.

The problem that needs to be solved is the binary classification of the result of an NBA game, into victory or defeat for the home or the away team. This solution is the result of the proper management and processing of historical data, with the aim of discovering features or even building new ones, which will optimize the desired results to the maximum.

The metric according to which the effectiveness of this work will be judged is accuracy. Finally, this work is likely to be a source of inspiration for future researchers, for its possible application, both to new data of NBA games and to different sports, but could also be used as room for improvement. Enhancing this effort to extend the current thesis' scope, three different deep machine learning methods, have been tested, such as ANN, LSMT and TCN, as well as an additional analysis, in which it will be investigated whether in specific time periods , such as months or match-days, our models are likely to achieve better performance or not.

Keywords

Machine Learning, Prediction, Deep Learning, Binary Classification, Optimization, accuracy, NBA, Logistic Regression, K Neighbors Classifier, Naive Bayes, Decision Tree Classifier, SVM - Linear Kernel, SVM - Radial Kernel, Gaussian Process Classifier, MLP Classifier, Ridge Classifier, Random Forest Classifier, Quadratic Discriminant Analysis, Ada Boost Classifier, Gradient Boosting Classifier, Linear Discriminant Analysis, Extra Trees Classifier, Extreme Gradient Boosting, Light Gradient Boosting Machine,CatBoost Classifier

στους γονείς και στον αδερφό μου

Ευχαριστίες

Η παρούσα μεταπτυχιακή εργασία εκπονήθηκε στα πλαίσια του μεταπτυχιακού κύκλου σπουδών ¨Επιστήμη των Δεδομένων και Μηχανική Μάθηση¨ της Σχολής Ηλεκτρολόγων Μηχανικών & Μηχανικών Υπολογιστών του Εθνικού Μετσοβίου Πολυτεχνείου και ολοκληρώνει το πιο πρόσφατο ακαδημαϊκό μου επίτευγμα. Η χαρά μου είναι μεγάλη και θα ήθελα να ευχαριστήσω όλους τους ανθρώπους που συνέβαλαν στο να φτάσω ως εδώ.

Αρχικά, θα ήθελα να ευχαριστήσω τον κ. Γεώργιο Στάμου, Καθηγητή ΕΜΠ, που ήταν επιβλέπων της εργασίας και μου επέτρεψε να καταπιαστώ με το συγκεκριμένο θέμα, που τόσο πολύ επιθυμούσα, και να διευρύνω τους επιστημονικούς μου ορίζοντες. Ακολούθως, θα ήθελα να ευχαριστήσω τους κ. Στέφανο Κόλλια, Καθηγητή ΕΜΠ, και κ. Αθανάσιο Βουλόδημο, Επίκουρο Καθηγητή ΕΜΠ, για την τιμή που μου έκαναν να αποτελέσουν την εξεταστική επιτροπή.

Θα ήθελα να ευχαριστήσω θερμά την κα. Παρασκευή Τζούβελη, ΕΔΙΠ ΕΜΠ, που ήταν η συνεπιβλέπουσα της εργασίας και με βοήθησε σημαντικά στην καταλληλότερη προσέγγιση του προβλήματος. Η καθοδήγησή της και η στήριξη της, τόσο σε επιστημονικό, όσο και σε ανθρώπινο επίπεδο, συνέβαλαν τα μέγιστα στην ολοκλήρωση της εργασίας.

Ένα μεγάλο ευχαριστώ οφείλω ακόμα στους φίλους και συμφοιτητές μου, που ενάμισι χρόνο τώρα με στήριξαν και με βοήθησαν να φτάσω ως εδώ και συγκεκριμένα τον Κωνσταντίνο Πλεύρη για την καθοδήγησή του, στην επιλογή αυτού του μεταπτυχιακού και στη συνολική συνεισφορά του.

Θα ήθελα να ευχαριστήσω ακόμα περισσότερο τους συνεργάτες που είχα όλα αυτά τα χρόνια, σε εργασίες και ασκήσεις, καθώς και τα άτομα που διαβάσαμε και μοχθήσαμε μαζί με τις ώρες.

Τέλος, ένα μεγάλο ευχαριστώ στους δικούς μου ανθρώπους, τους φίλους και την οικογένειά μου, που πάντα πίστευαν σε μένα και με στήριζαν.

Σταύρος Κετσέας

Περιεχόμενα

Κατάλογος Σχημάτων

Κατάλογος Πινάκων

Κεφάλαιο 1

Εισαγωγή

Από την αρχαιότητα ο άνθρωπος εξέφραζε τις ανησυχίες του για το μέλλον του και για το πώς θα μπορούσε να το προβλέψει, αγωνιώντας για το τι θα ακολουθήσει. Επιθυμούσε να μπορεί να προβλέπει μελλοντικές καταστάσεις, όπως τα καιρικά φαινόμενα που θα επικρατήσουν στο άμεσο μέλλον, για να μπορέσει να κερδίσει και τα ανάλογα πλεονεκτήματα, όπως, για παράδειγμα η έκβαση ενός πολέμου ή ενός αποτελέσματος αθλητικού γεγονότος, όπως τους νικητές των Ολυμπιακών αγώνων.

Στην αρχαία Ελλάδα, συγκεκριμένα, για λιγότερο σημαντικές, καθημερινές αποφάσεις, αλλά και για πολύ σπουδαίες, όπως η εκκίνηση μιας πολεμικής εκστρατείας, γινόταν η χρήση προφητειών από τα μαντεία, με περίτεχνους τρόπους, όπως το μάσημα φύλλων δάφνης ή και η ερμηνεία οιωνών.

Η μέθοδος της επίτευξης της επιθυμητής πρόβλεψης αλλάζει με τα χρόνια μορφές, αλλά η δίψα του ανθρώπου για την ποθητή μαντεψιά παραμένει η ίδια. Η προσπάθεια ανάκτησης της πληροφορίας εκείνης, η οποία θα μπορέσει να οδηγήσει στην τελική πρόβλεψη, είναι αδιάκοπη ανά τους αιώνες. Όπλο του ανθρώπου στον αγώνα αυτό, αποτελεί η τεχνολογία και η εξέλιξή της. Οι αποτυχημένες προσπάθειες και οι δοκιμές πολλών και διαφορετικών πιθανών εναλλακτικών σεναρίων, αρχίζουν σιγά σιγά και ξετυλίγουν τον τρόπο προσέγγισης της βέλτιστης λύσης.

Και εδώ εισάγεται η έννοια των δεδομένων, αλλά και έννοιες όπως το παρελθόν και η προϊστορία. Δεδομένα, των οποίων η κατάλληλη προεπεξεργασία και ανάλυση, δύνανται να οδηγήσουν σε επεξηγήσιμη πληροφορία, ιδανική για την ορθότερη πρόγνωση ενός μελλοντικού συμβάντος. Μέσω της εύρεσης επαναλαμβανόμενων μοτίβων που συνέβησαν στο παρελθόν, συχνοτήτων και ποσοστών εμφάνισης γεγονότων, αλλά και προτύπων στα δεδομένα, ο άνθρωπος, πλέον, κατόρθωσε συστηματικά, μεθοδικά και ορθολογικά να βελτιώσει αισθητά τις προβλέψεις του.

Σήμερα ζούμε στην εποχή των μεγάλων δεδομένων. Η ραγδαία ανάπτυξη της τεχνολογίας και πιο συγκεκριμένα της επιστήμης της πληροφορικής, μας επιτρέπει την πρόσβαση σε τεράστιους όγκους δεδομένων, άρα και σε πολύτιμες πληροφορίες. Η χειρόγραφη καταγραφή σταματάει και το διαδίκτυο επιτρέπει την άμεση ανταλλαγή πληροφορίας σε μακρινές αποστάσεις. Θεωρητικά αυτή η έκρηξη δεδομένων θα σήμαινε και τέλειες προβλέψεις. Αρχικά δεν συνέβη αυτό, καθώς προέκυψε το πρόβλημα αποθήκευσης, επεξεργασίας και μελέτης τέτοιου όγκου δεδομένων. Το πρόβλημα αυτό ξεπεράστηκε επιτυχώς με τη δημιουργία υπολογιστικών συστημάτων, ικανών να επεξεργαστούν εκατομμύρια δεδομένα ταυτόχρονα,

καθιστώντας γρηγορότερη και ευκολότερη την αποθήκευσή τους.

Τα σκήπτρα των προβλέψεων έρχεται να κατακτήσει η Μηχανική Μάθηση, μια τεχνολογία διαχείρησης παρελθοντικών δεδομένων, με τρομακτικά εύστοχες προβλέψεις, υπό την κατάλληλα, πάντα, προεπεξεργασία τους. Οι εφαρμογές της είναι αμέτρητες. Μερικές από αυτές είναι η Ιατρική, τα Οικονομικά, τα Τραπεζικά Συστήματα, η γεωργία, η κτηνοτροφία, ο αθλητισμός. Συγκεκριμένα στον αθλητισμό ξεχωρίζει η πρόβλεψη αποτελεσμάτων αγώνων, στατιστικών αγώνα και πρωταθλητών σεζόν, των μεγαλύτερων πρωταθλημάτων παγκοσμίως. Το λαμπρότερο από αυτά, αλλά και εκείνο με το μεγαλύτερο οικονομικό ενδιαφέρον είναι το NBA.

1.1 Αντικείμενο της διπλωματικής

Το πρωτάθλημα της Εθνικής Ομοσπονδίας Καλαθοσφαίρισης των Ηνωμένων Πολιτειών Αμερικής, γνωστό και ως NBA, θεωρείται από πολλούς το κορυφαίο και θεαματικότερο άθλημα παγκοσμίως. Είναι λογικό, όπως και στα υπόλοιπα κορυφαία πρωταθλήματα, να κεντρίζεται το οικονομικό ενδιαφέρον πολλών παραγόντων, γύρω από το χώρο του εκάστοτε αθλήματος. Το οικονομικό μέγεθος της βιομηχανίας του αθλητισμού, το 2022, ξεπερνάει κατά πολύ το 1 τρισεκατομμύριο δολλάρια, γεγονός που ενθαρρύνει την επιστημονική έρευνα και ανάλυση.

Ο κλάδος της ανάλυσης δεδομένων καλαθοσφαίρισης ασχολείται με την παραγωγή χρήσιμης πληροφορίας, ικανής να βελτιώσει τόσο την απόδοση των παικτών ατομικά, όσο και συνολικά των ομάδων, διευκολύνοτας έτσι σημαντικά το έργο των προπονητών και μαγνητίζοντας την προσοχή των διευθυντών, αλλά και των κυνηγών ταλέντων. Εάν παρομοιάσει κανείς την έκβαση ενός αγώνα με την επίλυση ενός γρίφου, τότε σίγουρα τα στοιχεία που θα οδηγήσουν στην αποκρυπτογράφησή του θα είναι τα ατομικά χαρακτηριστικά των παικτών, η φυσική κατάσταση της ομάδας, οι τακτικές των προπονητών, αλλά και πολλοί, ακόμη, απρόβλεπτοι παράγοντες.

Στόχος της παρούσας εργασίας είναι η ανάπτυξη μοντέλων μηχανικής μάθησης που επιτυγχάνουν υψηλή ευστοχία ή ορθότητα (accuracy) στις προβλέψεις τους, ίσης ή υψηλότερης σε σχέση με την ήδη υπάρχουσα βιβλιογραφία. Η ανάλυση θα γίνει σε διαφορετικό σύνολο δεδομένων από την αναφερθείσα βιβλιογραφία και θα χρησιμοποιηθούν 18 διαφορετικοί αλγόριθμοι μηχανικής μάθησης, αλλά θα δοκιμαστούν και τεχνικές βαθιάς μάθησης για να αποδειχθεί η χρησιμότητα ή μη χρησιμότητά τους στα συγκεκριμένα δεδομένα. Το σύνολο δεδομένων περιλαμβάνει αγώνες από τα έτη 2012 έως και 2018 του NBA, με τα αντίστοιχα στατιστικά αγώνα ομάδας (team box scores). Θα ακολουθηθούν ήδη υπάρχοντες τρόποι προσέγγισης του προβλήματος, αλλά και κατασκευή νέως χαρακτηριστικών, εμπνευσμένων από εμάς. Σε περίπτωση που δεν εφαρμοστεί μείωση της διαστατικότητας με μεθόδους όπως η PCA, θα μπορέσουμε να ερμηνεύσουμε τα αποτελέσματα μας, αποδίδοντας την υψηλότερη ορθότητα στα καλύτερα 15 χαρακτηριστικά, τα οποία διαδραμάτισαν το σημαντικότερο ρόλο για τη δημιουργία του βέλτιστου μοντέλου, με τις αντίστοιχες υπερπαραμέτρους του. Επιπρόσθετη ανάλυση θα πραγματοποιηθεί για την πιθανή εύρεση μοτίβου αύξησης της ορθότητας σε σχέση με το χρόνο (μήνας ή αγωνιστικές), ανά σεζόν.

1.2 Οργάνωση του τόμου

Η εργασία αυτή είναι οργανωμένη σε επτά κεφάλαια: Στο κεφάλαιο 1 γίνεται η εισαγωγή στο θέμα και αναφέρεται το αντικείμενο και ο στόχος της εν λόγω εργασίας. Στο Κεφάλαιο 2 δίνεται το θεωρητικό υπόβαθρο των βασικών τεχνολογιών Μηχανικής και Βαθιάς Μάθησης που σχετίζονται με τη διπλωματική αυτή. Αρχικά περιγράφεται η θεωρία πίσω από τη Μηχανική Μάθηση και τις μετρικές εκτίμησης της αποτελεσματικότητας ενός μοντέλου Μηχανικής Μάθησης και στη συνέχεια γίνεται ανάλυση της θεωρίας των αλγορίθμων που χρησιμοποιήθηκαν. Στο Κεφάλαιο 3 περιγράφονται οι σχετικές με το θέμα βιβλιογραφικές αναφορές, οι οποίες αποτέλεσαν πηγή έμνευσης και κατευθυντήριο μοχλό για την επίτευξη των επιθυμητών αποτελεσμάτων. Στο Κεφάλαιο 4 παρουσιάζεται το σύνολο δεδομένων που χρησιμοποιήθηκε, καθώς και η διαδικασία προ-επεξεργασίας και μηχανικής χαρακτηριστικών. Η εφαρμογή και υλοποίηση των μοντέλων, μαζί με τα αποτελέσματά τους, με ανάλυση των υπερπαραμέτρων τους και λεπτομέρειες σχετικά με τις πλατφόρμες και τα προγραμματιστικά εργαλεία που χρησιμοποιήθηκαν δίνονται στο Κεφάλαιο 5. Στο Κεφάλαιο 6 παρουσιάζονται συνοπτικά και συγκεντρωτικά τα συμπεράσματα και η συνεισφορά αυτής της εργασίας. Τέλος, στο Κεφάλαιο 7 δίνονται οι προκλήσεις αυτή της διπλωματικής εργασίας, καθώς και μελλοντικές επεκτάσεις της.

Μέρος Ι

Θεωρητικό Μέρος

Κεφάλαιο 2

Θεωρητικό υπόβαθρο

Στο κεφάλαιο αυτό παρουσιάζονται οι γενικότερες έννοιες της πρόβλεψης ενός αθλητικού γεγονότος και της δυαδικής ταξινόμησης, αλλά και ειδικότερα οι τεχνικές και οι αλγόριθμοι μηχανικής και βαθιάς μάθησης που χρησιμοποιήθηκαν για την ευστοχότερη πρόβλεψη του αποτελέσματος ενός αγώνα NBA. Εκτός αυτών, περιγράφονται και οι αντίστοιχες μετρικές που χρησιμοποιήθηκαν για το σκοπό αυτό.

2.1 Πρόβλεψη Αθλητικού Γεγονότος

2.1.1 Τι είναι γενικότερα η πρόβλεψη (prediction)

Η πρόβλεψη είναι η εκτίμηση του τι πρόκειται να γίνει ή αλλιώς η προγενέστερη δήλωση ενός μελλοντικού συμβάντος σε σχέση με την πραγματοποίησή του. Οι προβλέψεις συχνά, αλλά όχι πάντοτε, βασίζονται στην εμπειρία και στη γνώση [14].

Τα μελοντικά γεγονότα είναι αβέβαια και ως αποτέλεσμα δεν μπορεί να υπάρξει εγγυημένη, έγκυρη πληροφόρηση για την έκβαση ενός μελλονικού γεγονότος. Η έννοια της πρόβλεψης συνδράμει στην κατάστρωση σχεδίων για πιθανά τέτοια γεγονότα [14].

Με μια μη στατιστική οπτική, ο όρος ¨πρόβλεψη¨ χρησιμοποιείται συχνά και αναφέρεται σε μια τεκμηριωμένη εικασία ή γνώμη. Μια πρόβλεψη αυτού του είδους, μπορεί να ενισχυθεί από τη συλλογιστική πορεία μιας σκέψης και την εμπειρία ενός ατόμου που προσπαθεί να προβλέψει και μπορεί να είναι χρήσιμη, αν το άτομο αυτό κατέχει τη γνώση στο πεδίο στο οποίο προσπαθεί να πραγματοποιήσει την πρόβλεψη.

Η μέθοδος Delphi [15] είναι μια τεχνική για την εξαγωγή τέτοιων προβλέψεων, που βασίζονται σε κρίσεις ειδικών, με ελεγχόμενο τρόπο. Αυτός ο τύπος πρόβλεψης μπορεί να γίνει αντιληπτός ως συνεπής με στατιστικές τεχνικές με την έννοια ότι, τουλάχιστον, τα ¨δεδομένα¨ που χρησιμοποιούνται είναι οι γνωστικές εμπειρίες του ειδικού που προβλέπει μια διαισθητική ¨καμπύλη πιθανοτήτων¨.

2.1.2 Τι είναι η πρόβλεψη στη στατιστική

Στη στατιστική, η πρόβλεψη είναι μέρος της στατιστικής συμπερασμάτων. Μια συγκεκριμένη προσέγγιση σε μια τέτοια εξαγωγή συμπερασμάτων είναι γνωστή ως προγνωστική εξαγωγή συμπερασμάτων, αλλά η πρόβλεψη μπορεί να πραγματοποιηθεί σε οποιαδήποτε από

τις διάφορες προσεγγίσεις στα στατιστικά συμπεράσματα. Όταν οι πληροφορίες μεταφέρονται με την πάροδο του χρόνου, συχνά σε συγκεκριμένα σημεία του χρόνου, η διαδικασία είναι γνωστή ως πρόγνωση (forecasting), όπως για παράδειγμα η πρόγνωση του καιρού. Η πρόγνωση απαιτεί συνήθως μεθόδους χρονοσειρών, ενώ η απλή πρόβλεψη εκτελείται συχνά σε εγκάρσιας διατομής δεδομένα [14].

Οι στατιστικές τεχνικές που χρησιμοποιούνται για την πρόβλεψη περιλαμβάνουν την ανάλυση παλινδρόμησης και τις διάφορες υποκατηγορίες της, όπως η γραμμική παλινδρόμηση, τα γενικευμένα γραμμικά μοντέλα (λογιστική παλινδρόμηση, παλινδρόμηση (Poisson) κ.λ.π. Όταν αυτά ή/και σχετικό, γενικευμένο σύνολο μεθόδων παλινδρόμησης ή μηχανικής μάθησης αναπτύσσονται σε εμπορική χρήση, το πεδίο είναι γνωστό ως προγνωστική ανάλυση [14].

Σε πολλές εφαρμογές, όπως η ανάλυση χρονοσειρών, είναι δυνατό να εκτιμηθούν τα μοντέλα που δημιουργούν τις παρατηρήσεις. Εάν τα μοντέλα μπορούν να εκφραστούν ως συναρτήσεις μεταφοράς ή με όρους παραμέτρων χώρου κατάστασης, τότε μπορούν να υπολογιστούν εξομαλυνόμενες, φιλτραρισμένες και προβλεπόμενες εκτιμήσεις δεδομένων. Η εξομάλυνση διακύμανσης μπορεί να χρησιμοποιηθεί για την ανάκτηση δεδομένων ενδιαφέροντος από θορυβώδεις μετρήσεις. Αυτές οι τεχνικές βασίζονται σε προγνωστικά ενός βήματος μπροστά (τα οποία ελαχιστοποιούν τη διακύμανση του σφάλματος πρόβλεψης). Όταν τα μοντέλα δημιουργίας είναι μη γραμμικά, τότε μπορούν να εφαρμοστούν σταδιακές γραμμικοποιήσεις εντός του Εκτεταμένου φίλτρου Kalman [16] και ομαλότερες αναδρομές. Ωστόσο, σε μη γραμμικές περιπτώσεις, οι εγγυήσεις απόδοσης βέλτιστης ελάχιστης διακύμανσης δεν ισχύουν πλέον [14].

Για να χρησιμοποιηθεί η ανάλυση παλινδρόμησης για πρόβλεψη, συλλέγονται δεδομένα για τη μεταβλητή που πρόκειται να προβλεφθεί, που ονομάζεται εξαρτημένη μεταβλητή ή μεταβλητή απόκρισης, και σε μία ή περισσότερες μεταβλητές των οποίων οι τιμές υποτίθεται ότι την επηρεάζουν, που ονομάζονται ανεξάρτητες μεταβλητές ή επεξηγηματικές μεταβλητές. Μια συναρτησιακή μορφή, συχνά γραμμική, υποτίθεται για την υποτιθέμενη αιτιακή σχέση και οι παράμετροι της συνάρτησης εκτιμώνται από τα δεδομένα - δηλαδή επιλέγονται έτσι, ώστε να βελτιστοποιηθεί με κάποιο τρόπο η προσαρμογή της συνάρτησης, που παραμετροποιείται έτσι, στο δεδομένα. Αυτό είναι το βήμα της εκτίμησης. Για το βήμα πρόβλεψης, οι επεξηγηματικές τιμές μεταβλητής που θεωρούνται σχετικές με μελλοντικές (ή τρέχουσες αλλά δεν έχουν παρατηρηθεί ακόμη) τιμές της εξαρτημένης μεταβλητής εισάγονται στην παραμετροποιημένη συνάρτηση για τη δημιουργία προβλέψεων για την εξαρτημένη μεταβλητή [14].

2.1.3 Τι είναι η πρόβλεψη στον αθλητισμό

Η πρόβλεψη της έκβασης των αθλητικών γεγονότων έχει αυξηθεί σε δημοτικότητα τα τελευταία χρόνια. Οι χάντικαπερ - άτομα που έχουν οριστεί για να διορθώσουν ή να αξιολογήσουν το μειονέκτημα ή πλεονέκτημα ενός αγωνιζόμενου ή μιας ομάδας - προβλέπουν το αποτέλεσμα των παιχνιδιών χρησιμοποιώντας μια ποικιλία μαθηματικών τύπων, μοντέλων προσομοίωσης ή ποιοτικής ανάλυσης. Παλιοί παίκτες, όπως ο Τζίμι ο Έλληνας [17], πιστεύεται ότι είχαν πρόσβαση σε πληροφορίες που τους έδιναν κάποιου είδους πλεονέκτημα. Οι

πληροφορίες κυμαίνονταν από προσωπικά ζητήματα, όπως ο τζόγος ή το ποτό μέχρι τους άγνωστους τραυματισμούς ή και γενικότερα οτιδήποτε μπορεί να επηρεάσει την απόδοση ενός παίκτη στο γήπεδο και συνεπώς την έκβαση του αποτελέσματος [14]. Οι πρόσφατοι καιροί άλλαξαν τον τρόπο με τον οποίο προβλέπονται τα αθλήματα. Οι προβλέψεις, σήμερα, τυπικά αποτελούνται από δύο διακριτές προσεγγίσεις: Παιχνίδια καταστάσεων και μοντέλα που βασίζονται σε στατιστικά. Τα παιχνίδια καταστάσεων είναι πολύ πιο δύσκολο να μετρηθούν. επειδή συνήθως περιλαμβάνουν το κίνητρο μιας ομάδας.

Η ευρεία χρήση της τεχνολογίας έχει φέρει μαζί της πιο σύγχρονα συστήματα αθλητικών στοιχημάτων. Αυτά τα συστήματα είναι συνήθως αλγόριθμοι και μοντέλα προσομοίωσης που βασίζονται σε ανάλυση παλινδρόμησης. Ο Jeff Sagarin [18], ένας αθλητικός στατιστικολόγος, έφερε την προσοχή στον αθλητισμό δημοσιεύοντας τα αποτελέσματα των μοντέλων του στο USA Today. Αυτή τη στιγμή πληρώνεται ως σύμβουλος από τους Ντάλας Μάβερικς για τις συμβουλές του σχετικά με τις ενδεκάδες και τη χρήση του συστήματος Winval του, το οποίο αξιολογεί τους ελεύθερους παράγοντες. Ο Brian Burke [19], ένας πρώην πιλότος μαχητικών του Ναυτικού που έγινε αθλητικός στατιστικολόγος, δημοσίευσε τα αποτελέσματά του σχετικά με τη χρήση της ανάλυσης παλινδρόμησης για να προβλέψει το αποτέλεσμα των παιχνιδιών NFL [20]. Ο Ken Pomeroy [21] είναι ευρέως αποδεκτός ως ηγετική αρχή στα στατιστικά του κολεγιακού μπάσκετ. Ο ιστότοπός του περιλαμβάνει τις βαθμολογίες του College Basketball, ένα σύστημα στατιστικών που βασίζεται στο ρυθμό. Σε αντίθεση με άλλα παιχνίδια που προσφέρονται σε ένα καζίνο, η πρόβλεψη σε αθλητικά γεγονότα μπορεί να είναι λογική και συνεπής [14].

2.2 Εισαγωγή στη θεωρία της Μηχανικής Μάθησης

2.2.1 Τι είναι η δυαδική ταξινόμηση (binary classification) στη Μηχανική Μάθηση

Στην ενότητα αυτή μεταβαίνουμε από τις γενικότερες έννοιες της πρόβλεψης και της στατιστικής σε πιο ειδικές, που περιγράφουν καλύτερα το πρόβλημα που καλείται να επιλύσει η παρούσα εργασία. Το πρόβλημα το οποίο καλούμαστε να επιλύσουμε, η πρόβλεψη δηλαδή του αποτελέσματος ενός αγώνα NBA, του οποίου το αποτέλεσμα είναι ¨Νίκη ή Ήττα¨, ή αλλιώς στη γλώσσα του υπολογιστή ¨0 ή 1¨ ή γενικότερα ¨Είναι - Δεν είναι¨, ονομάζεται πρόβλημα δυαδικής ταξινόμησης και η πρόβλεψη του αποτελέσματός του προσεγγίζεται ικανοποιητικά με τη χρήση αλγορίθμων μηχανικής μάθησης, ειδικά για σκοπούς δυαδικής ταξινόμησης.

Τι είναι όμως η δυαδική ταξινόμηση. Η δυαδική ταξινόμηση είναι η διαδικασία ταξινόμησης ενός συνόλου που αποτελείται από δύο ομάδες, οι οποίες ονομάζονται κλάσεις ή κατηγορίες, σύμφωνα με κάποιον κανόνα ταξινόμησης και μπορεί να επιλυθεί με τη χρήση της Μηχανικής Μάθησης. Τυπικά προβλήματα δυαδικής ταξινόμησης, για παράδειγμα, είναι η ανίχνευση μιας ασθένειας σε έναν ασθενή (νοσεί - δεν νοσεί) ή ο ποιοτικός έλεγχος σε μια βιομηχανία (πληρούνται οι προδιαγραφές - δεν πληρούνται) ή και στην ανάκτηση πληροφοριών (να είναι στα αποτελέσματα αναζήτησης η συγκεκριμένη σελίδα - να μην είναι) [1].

Η δυαδική ταξινόμηση θα λέγαμε ότι είναι η διχοτόμηση που εφαρμόζεται σε μια πραγ-

ματική κατάσταση. Σε πραγματικά σενάρια δυαδικής ταξινόμησης, οι κλάσεις δεν είναι ισορροπημένες και σε σχέση με τη συνολική ευστοχία που μπορούμε να επιτύχουμε (overall accuracy), η σχετική αναλογία διαφορετικών τύπων σφαλμάτων παρουσιάζει ενδιαφέρον. Για παράδειγμα, στις ιατρικές εξετάσεις, η ανίχνευση μιας ασθένειας όταν δεν υπάρχει (ψευδώς θετικό) είναι διαφορετικής σημαντικότητας από τη μη ανίχνευση μιας ασθένειας όταν υπάρχει (ψευδώς αρνητικό) [1].

2.2.2 Στατιστική Δυαδική Ταξινόμηση

Η στατιστική ταξινόμηση είναι ένα πρόβλημα που μελετάται στη μηχανική μάθηση. Είναι ένας τύπος εποπτευόμενης μάθησης, μια μέθοδος μηχανικής μάθησης όπου οι κατηγορίες είναι προκαθορισμένες και χρησιμοποιείται για την κατηγοριοποίηση νέων πιθανολογικών παρατηρήσεων στις εν λόγω κατηγορίες. Όταν υπάρχουν μόνο δύο κατηγορίες, το πρόβλημα είναι γνωστό ως στατιστική δυαδική ταξινόμηση [1].

Μερικές από τις πιο διαδεδομένες μεθόδους Στατιστικής Δυαδικής Ταξινόμησης είναι:

- Δέντρα Αποφάσεων (Decision trees)
- Τυχαία Δάση (Random forests)
- Μπεϋζιανά Δίκτυα (Bayesian networks)
- Μηχανές Ψποστήριξης Φορέα (Support vector machines)
- Νευρωνικά Δίκτυα (Neural networks)
- Λογιστική Παλινδρόμηση (Logistic regression)
- Μοντέλο Probit (Probit model)
- Γενετικός Προγραμματισμός (Genetic Programming)
- Προγραμματισμός πολλαπλών εκφράσεων (Multi expression programming)
- Γραμμικός γενετικός προγραμματισμός (Linear genetic programming)

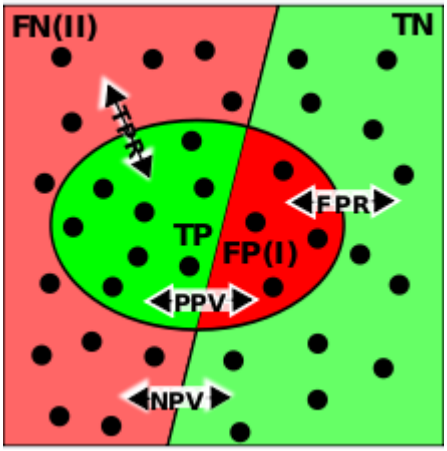
Κάθε ταξινομητής μπορεί να είναι καλύτερος μόνο σε έναν επιλεγμένο τομέα, με βάση τον αριθμό των παρατηρήσεων, τη διάσταση του διανύσματος χαρακτηριστικών, το θόρυβο στα δεδομένα και πολλούς άλλους παράγοντες. Για παράδειγμα, τα Random forests αποδίδουν καλύτερα από τα Support Vector Machines για τρισδιάστατα νέφη σημείων. Στη διπλωματική αυτή εργασία χρησιμοποιήθηκαν και άλλοι αλγόριθμοι μηχανικής μάθησης, των οποίων η λειτουργικότητα θα αναλυθεί στη συνέχεια.

2.2.3 Εκτίμηση Δυαδικών Ταξινομητών

Υπάρχουν πολλές μετρικές που μπορούν να χρησιμοποιηθούν για τη μέτρηση της απόδοσης ενός ταξινομητή. Διαφορετικά πεδία έχουν διαφορετικές προτιμήσεις για συγκεκριμένες μετρήσεις λόγω διαφορετικών στόχων. Στην ιατρική χρησιμοποιούνται συχνά η ευαισθησία (sensitivity) και η ειδικότητα (specificity), ενώ στην ανάκτηση πληροφοριών προτιμάται η

ακρίβεια (precision) και η ανάκληση (recall). Στο δικό μας πρόβλημα, καταλληλότερη μετρική θεωρείται η ορθότητα (accuracy), για λόγους που εξυπηρετούν τα δεδομένα και οι οποίοι θα αναλυθούν σε μετέπειτα ενότητα. Μια σημαντική διάκριση είναι μεταξύ μετρήσεων που είναι ανεξάρτητες από το πόσο συχνά εμφανίζεται κάθε κατηγορία στον πληθυσμό (ο επιπολασμός- prevalence) και μετρήσεων που εξαρτώνται από τον επιπολασμό – και οι δύο τύποι είναι χρήσιμοι, αλλά έχουν πολύ διαφορετικές ιδιότητες [1].

Δεδομένης της ταξινόμησης ενός συγκεκριμένου συνόλου δεδομένων, υπάρχουν τέσσερις βασικοί συνδυασμοί πραγματικής κατηγορίας δεδομένων και εκχωρημένης κατηγορίας: αληθώς θετικά TP (true positives ή σωστές θετικές εκχωρήσεις) , αληθώς αρνητικά TN (true negatives ή σωστές αρνητικές εκχωρήσεις), ψευδώς θετικά FP (false positives ή λανθασμένες θετικές εκχωρήσεις) και ψευδώς αρνητικά FN (false negatives ή λανθασμένες αρνητικές εκχωρήσεις) [1].



Σχήμα 2.1: *Οι τέσσερις βασικοί συνδυασμοί πραγματικής κατηγορίας δεδομένων και εκχωρημένης κατηγορίας [1].*

Αυτά μπορούν να τοποθετηθούν σε έναν πίνακα 2×2, με στήλες που αντιστοιχούν στην πραγματική τιμή – θετική συνθήκη ή αρνητική συνθήκη – και σειρές που αντιστοιχούν στην τιμή ταξινόμησης – θετικό αποτέλεσμα δοκιμής ή αρνητικό αποτέλεσμα δοκιμής.

2.2.4 Οι 8 βασικές αναλογίες

Υπάρχουν οκτώ βασικές αναλογίες που μπορεί κανείς να υπολογίσει από αυτόν τον πίνακα, οι οποίες διατίθενται σε τέσσερα συμπληρωματικά ζεύγη (κάθε ζεύγος αθροίζει στο 1). Αυτά προκύπτουν διαιρώντας κάθε έναν από τους τέσσερις αριθμούς με το άθροισμα της γραμμής ή της στήλης του, δίνοντας οκτώ αριθμούς, στους οποίους μπορούν να αναφέρονται γενικά με τη μορφή ¨αληθής θετικός λόγος γραμμής¨ ή ¨ψευδής αρνητικός λόγος στήλης¨.

Υπάρχουν λοιπόν δύο ζεύγη αναλογιών στηλών και δύο ζεύγη αναλογιών σειρών, και μπορεί κανείς να τους συνοψίσει με τέσσερις αριθμούς επιλέγοντας έναν λόγο από κάθε ζεύγος - οι άλλοι τέσσερις αριθμοί είναι τα συμπληρώματα [1].

Οι αναλογίες σειρών είναι:

- true positive rate (TPR) = (TP/(TP+FN)), γνωστή και ως ευαισθησία ή ανάκληση (precision or recall). Αυτή είναι το ποσοστό του πληθυσμού με την κατάσταση για την οποία

το τεστ είναι σωστό.

- Με συμπληρωματικό της τη false negative rate (FNR) = (FN/(TP+FN))
- true negative rate (TNR) = (TN/(TN+FP), γνωστή και ως ειδικότητα (specificity).
- Με συμπληρωματικό της τη false positive rate (FPR) = (FP/(TN+FP)).

Οι αναλογίες στηλών είναι:

- positive predictive value (PPV, γνωστό και ως precision) (TP/(TP+FP)). Πρόκειται για το ποσοστό του πληθυσμού με ένα δεδομένο αποτέλεσμα δοκιμής για το οποίο το τεστ είναι σωστό.
- Με συμπληρωματικό της τη false discovery rate (FDR) (FP/(TP+FP))
- negative predictive value (NPV) (TN/(TN+FN)).
- Με συμπληρωματικό της τη false omission rate (FOR) (FN/(TN+FN)).

Στις διαγνωστικές δοκιμές, οι κύριες αναλογίες που χρησιμοποιούνται είναι οι πραγματικοί λόγοι στηλών – πραγματικός θετικός ρυθμός και πραγματικός αρνητικός ρυθμός – όπου είναι γνωστοί ως ευαισθησία και ειδικότητα. Στην ανάκτηση πληροφοριών, οι κύριοι λόγοι είναι οι αληθινοί θετικοί λόγοι (γραμμή και στήλη) – θετική προγνωστική αξία και αληθινός θετικός ρυθμός – όπου είναι γνωστοί ως ακρίβεια και ανάκληση [1].

Υπάρχει μια σειρά από άλλες μετρήσεις, πιο απλά η ακρίβεια ή το Fraction Correct (FC), που μετρά το κλάσμα όλων των περιπτώσεων που έχουν κατηγοριοποιηθεί σωστά. Το συμπλήρωμα είναι το Fraction Incorrect (FiC). Το F-score συνδυάζει την ακρίβεια και την ανάκληση σε έναν αριθμό μέσω μιας επιλογής ζύγισης, πιο απλά ίσης ζύγισης, όπως η ισορροπημένη βαθμολογία (βαθμολογία F1). Ορισμένες μετρήσεις προέρχονται από συντελεστές παλινδρόμησης: η επισήμανση και η πληροφόρηση, και ο γεωμετρικός μέσος όρος τους, ο συντελεστής συσχέτισης Matthews. Άλλες μετρήσεις περιλαμβάνουν το στατιστικό J Youden, τον συντελεστή αβεβαιότητας, τον συντελεστή phi και το κάππα του Cohen [1].

2.2.5 Διαφορές μεταξύ βασικών μετρικών δυαδικής ταξινόμησης

Γιατί χρειαζόμαστε μετρήσεις αξιολόγησης.

Όντας άνθρωποι, θέλουμε να γνωρίζουμε την αποτελεσματικότητα ή την απόδοση οποιουδήποτε μηχανήματος ή λογισμικού συναντάμε, για παράδειγμα, αν σκεφτούμε ένα αυτοκίνητο που θέλουμε να μάθουμε τα χιλιόμετρα, ή εάν υπάρχει ένας συγκεκριμένος αλγόριθμος που θέλουμε να μάθουμε για την πολυπλοκότητα του χώρου και του χρόνου, ομοίως πρέπει να υπάρχει κάποιος τρόπος με τον οποίο μπορούμε να μετρήσουμε την αποτελεσματικότητα ή την απόδοση των Μοντέλων Μηχανικής Μάθησης [2].

Για αυτό το λόγο, θα μετελήσουμε μερικές από τις μετρικές για τα μοντέλα ταξινόμησής μας. Υπάρχουν ξεχωριστές μετρικές για τα μοντέλα παλινδρόμησης και ταξινόμησης, καθώς η παλινδρόμηση μας δίνει συνεχείς τιμές ως έξοδο και η ταξινόμηση μας δίνει διακριτές τιμές ως έξοδο. Θα επικεντρωθούμε στις μετρικές ταξινόμησης.

- **Accuracy - Ορθότητα**

  Η πιο συχνά και ευρέως χρησιμοποιούμενη μετρική, για οποιοδήποτε μοντέλο είναι η ορθότητα. Βασικά κάνει αυτό που λέει, υπολογίζει ποια είναι η ορθότητα πρόβλεψης του μοντέλου μας. Είναι η μετρική εκείνη, πάνω στην οποία βασίζεται η παρούσα εργασία. Η φόρμουλα δίνεται ως εξής [1]:

$$Accuracy = \frac{NumberOfCorrectPredictions}{TotalNumberOfPoints} * 100$$

  Όπως μπορούμε να δούμε, βασικά μας λέει μεταξύ όλων των σημείων, πόσα από αυτά έχουν προβλεφθεί σωστά.

  Πλεονεκτήματα :

  1. Εύκολη στη χρήση μετρική.
  2. Εύκολη στην κατανόηση και τη συσχέτιση.
  3. Παρέχει σωστή αποτελεσματικότητα του μοντέλου εάν τα σημεία δεδομένων είναι ισορροπημένα, όπως αυτά του προβλήματός μας.
  4. Μπορεί να χρησιμοποιηθεί για σύγκριση δύο μοντέλων, καθώς δίνει μια ενιαία τιμή ως μέτρηση.

  Μειονεκτήματα:

  1. Δεν συνιστάται για μη ισορροπημένα δεδομένα, καθώς τα αποτελέσματα μπορεί να είναι παραπλανητικά. Για παράδειγμα έχουμε 100 σημεία δεδομένων μεταξύ των οποίων οι 95 μονάδες είναι αρνητικές και οι 5 μονάδες είναι θετικές, και αν έχουμε ένα ¨ανόητο¨ μοντέλο, το οποίο μόνο προβλέπει αρνητικά αποτελέσματα, τότε στο τέλος της εκπαίδευσης θα έχουμε ένα μοντέλο που θα προβλέπει μόνο αρνητικά και θα εξακολουθεί να είναι 95 τοις εκατό ορθό, με βάση τον παραπάνω τύπο. Ως εκ τούτου, δεν συνιστάται για μη ισορροπημένα δεδομένα. Τα δικά μας δεδομένα δεν είναι ανισόρροπα.
  2. Δεν καταλαβαίνουμε πού κάνει λάθη το μοντέλο μας.
  3. Δεν λαμβάνει υπόψη λανθασμένες προβλέψεις.
  4. Η βαθμολογία πιθανότητας δεν λαμβάνεται υπόψη.
  5. Δεν είναι τόσο ερμηνεύσιμο, όσο ο πίνακας σύγχυσης.

- **Confusion Matrix - Πίνακας Σύγχυσης**

  Όπως υποδηλώνει το όνομα, είναι ένας πίνακας 2x2 που έχει τα ¨Πραγματικά¨ και τα ¨Προβλεπόμενα¨ ως Γραμμές και Στήλες, αντίστοιχα. Καθορίζει τον αριθμό των Σωστών και Λανθασμένων Προβλέψεων. Δεν ασχοληθήκαμε με τις λανθασμένες προβλέψεις στη μέθοδο Ακρίβειας και λαμβάνουμε υπόψη μόνο τις σωστές, επομένως το Confusion Matrix μας βοηθά να κατανοήσουμε και τις δύο πτυχές [1].

  Ας ρίξουμε μια ματιά στο διάγραμμα για να το κατανοήσουμε καλύτερα:

Confusion Matrix		Predicted	
		Positive	Negative
Actual	Positive	True Positive	False Negative
	Negative	False Positive	True Negative

Σχήμα 2.2: *Confusion Matrix [2]*

Ας φανταστεί κανείς ότι υπάρχει ένα πρόβλημα δυαδικής ταξινόμησης με τις κλάσεις ως θετικές και αρνητικές ετικέτες. Τώρα, εάν το πραγματικό μας σημείο είναι θετικό και το προβλεπόμενο από το μοντέλο μας σημείο είναι επίσης θετικό, τότε θα λάβουμε ένα Αληθώς θετικό, εδώ "True" σημαίνει σωστά ταξινομημένο και ¨Θετικό¨ είναι η προβλεπόμενη κλάση από το μοντέλο. Ομοίως, αν έχουμε την πραγματική τάξη ως αρνητική και προβλέφθηκε ως θετική, δηλαδή μια εσφαλμένη προβλεπόμενη, τότε θα λάβουμε False Positive, "False" σημαίνει Λανθασμένη πρόβλεψη και "Positive" είναι η προβλεπόμενη κλάση από το μοντέλο [1].

Θέλουμε πάντα τα διαγώνια στοιχεία, της κύριας διαγωνίου του δισδιάστατου πίνακα, να έχουν υψηλές τιμές. Καθώς είναι σωστές προβλέψεις, δηλαδή TP και TN.

Πλεονεκτήματα :

1. Καθορίζει ένα συγχυσμένο μοντέλο ανάμεσα σε ετικέτες μιας κατηγορίας.
2. Λαμβάνει τους τύπους σφαλμάτων που γίνονται από το μοντέλο.
3. Καλύτερο από την ορθότητα, καθώς δείχνει και τις λανθασμένες προβλέψεις. Καταλαβαίνει σε βάθος τα λάθη που κάνει το μοντέλο και διορθώνει τις περιοχές όπου πηγαίνει λανθασμένα.

Μειονεκτήματα:

1. Όχι πολύ κατάλληλο για Multiclass ταξινόμηση.
2. Δεν δίνει πιθανότητες κατηγορίας.
3. Ο έλεγχος για υπερπροσαρμογή ή υποπροσαρμογή του μοντέλου είναι δύσκολος, καθώς δεν έχουμε ενιαία τιμή.
4. Δεν μπορεί να χρησιμοποιηθεί για σύγκριση 2 μοντέλων, καθώς δεν δίνει μία αποκλειστική μετρική.

- **Precision and Recall - Ακρίβεια και Ανάκληση**

  Η ακρίβεια είναι η μετρική που δηλώνει, μεταξύ όλων των προβλεπόμενων θετικών κλάσεων, πόσες είναι πραγματικά θετικές, ο τύπος δίνεται παρακάτω [1]:

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive}$$

  Η ανάκληση είναι η μετρική που δηλώνει, μεταξύ όλων των θετικών τάξεων πόσες έχουν πραγματικά προβλεφθεί σωστά, ο τύπος δίνεται παρακάτω:

$$Recall = \frac{TruePositive}{TruePositive + FalseNegative}$$

Συχνά αναζητούμε υψηλή ακρίβεια και ανάκληση. Αν και τα δύο είναι υψηλά σημαίνει ότι το μοντέλο μας είναι λογικό. Εδώ, λαμβάνουμε επίσης υπόψη τα λανθασμένα σημεία. Επομένως γνωρίζουμε πού κάνει λάθη το μοντέλο μας. Λαμβάνεται επίσης υπόψη η τάξη μειοψηφίας.

Πλεονεκτήματα :

1. Μας λέει για την αποτελεσματικότητα του μοντέλου.

2. Μας δείχνει επίσης το πόσο τα δεδομένα είναι προκατειλημμένα προς μια κλάση.

3. Μας βοηθά να κατανοήσουμε εάν το μοντέλο μας προσαρμόζεται καλά σε μη ισορροπημένο σύνολο δεδομένων, για τη μειοψηφική κατηγορία.

Μειονεκτήματα :

1. Η ανάκληση ασχολείται με τα Αληθώς θετικά και τα Ψευδώς αρνητικά και η ακρίβεια ασχολείται με τα Αληθώς θετικά και τα Ψευδώς θετικά. Δεν ασχολείται με όλα τα κελιά του πίνακα σύγχυσης. Τα Αληθώς αρνητικά δεν λαμβάνονται ποτέ υπόψη.

2. Ως εκ τούτου, η ακρίβεια και η ανάκληση θα πρέπει να χρησιμοποιούνται μόνο σε καταστάσεις όπου η σωστή αναγνώριση της αρνητικής κατηγορίας δεν παίζει ρόλο.

3. Εστιάζει μόνο στη θετική τάξη.

4. Ταιριάζει καλύτερα στη δυαδική ταξινόμηση.

5. Τα μοντέλα δεν μπορούν να συγκριθούν με βάση και τις δύο βαθμολογίες, καθώς δεν είναι μια αποκλειστική μετρική.

- **F1 score**

  Η βαθμολογία F1 είναι ο αρμονικός μέσος όρος της ακρίβειας και της ανάκλησης, όπου μια βαθμολογία F1 φτάνει την καλύτερη τιμή της στο 1 (τέλεια ακρίβεια και ανάκληση). Η βαθμολογία F1 είναι επίσης γνωστή ως συντελεστής Sorensen–Dice ή συντελεστής ομοιότητας ζαριών (DSC) [1].

  Αξιοποιεί τόσο τα πλεονεκτήματα της Ακρίβειας όσο και της Ανάκλησης. Ένα ιδανικό μοντέλο θα έχει ακρίβεια και ανάκληση 1, επομένως η βαθμολογία F1 θα είναι επίσης 1.

  Πλεονεκτήματα :

  1. Μπορεί να χρησιμοποιηθεί για σύγκριση μοντέλων, καθώς πρόκειται για μια ενιαία μέτρηση.

  2. Ο αρμονικός μέσος όρος τείνει να είναι πιο κοντά στη χαμηλότερη από τις τιμές, επομένως εάν ένα μοντέλο έχει οποιαδήποτε χαμηλότερη τιμή από τα δύο, Precision ή Recall, τελικά το F1 θα είναι χαμηλότερο εξαιτίας αυτού.

  Μειονεκτήματα :

  1. Είναι το ίδιο με το Precision και το Recall.

- **AU-ROC**

  Το AU-ROC είναι η περιοχή κάτω από την καμπύλη λειτουργίας του δέκτη. Είναι ένα γράφημα που δείχνει την απόδοση ενός μοντέλου, για όλες τις τιμές που θεωρούνται ως κατώφλι.

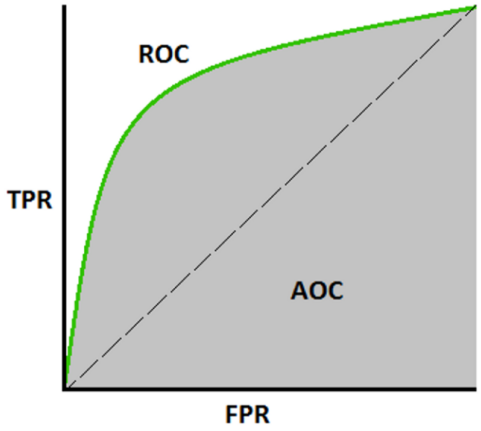
  Καθώς το AU-ROC είναι ένα γράφημα, έχει τον δικό του άξονα X και τον άξονα Y, ενώ ο άξονας X είναι FPR και ο άξονας Y είναι TPR, όπου

  TPR = Αληθώς θετικό / (Αληθώς θετικό + Ψευδώς αρνητικό)

  FPR = Ψευδώς θετικό / (Ψευδώς θετικό + Αληθώς αρνητικό)

  Τα διαγράμματα μιας καμπύλης ROC είναι βασικά TPR έναντι FPR που υπολογίζονται σε διαφορετικά κατώφλια ταξινόμησης. Η μείωση του ορίου ταξινόμησης ταξινομεί περισσότερα στοιχεία ως θετικά, αυξάνοντας έτσι τόσο τα Ψευδώς θετικά όσο και τα Αληθώς θετικά, δηλαδή τις σωστές προβλέψεις.

  Όλες οι τιμές ταξινομούνται και απεικονίζονται σε ένα γράφημα και η περιοχή κάτω από την καμπύλη ROC είναι η πραγματική απόδοση του μοντέλου σε διαφορετικά κατώφλια [1].



Σχήμα 2.3: *AU-ROC μετρική [2].*

Πλεονεκτήματα :

1. Μια απλή γραφική αναπαράσταση της διαγνωστικής ακρίβειας ενός τεστ: όσο πιο κοντά είναι η κορυφή της καμπύλης προς την επάνω αριστερή γωνία, τόσο μεγαλύτερη είναι η διακριτική ικανότητα του τεστ.

2. Επίσης, επιτρέπει ένα πιο σύνθετο (και πιο ακριβές) μέτρο της ακρίβειας μιας δοκιμής, που είναι η AUC.

3. Η AUC με τη σειρά της μπορεί να χρησιμοποιηθεί ως απλή αριθμητική βαθμολογία της ακρίβειας των διαγνωστικών εξετάσεων, η οποία απλοποιεί τη σύγκριση μεταξύ των διαγνωστικών δοκιμών.

Μειονεκτήματα :

1. Τα πραγματικά κατώφλια απόφασης συνήθως δεν εμφανίζονται στη γραφική παράσταση.

2. Καθώς το μέγεθος του δείγματος μειώνεται, η γραφική παράσταση γίνεται πιο οδοντωτή.

3. Δεν ερμηνεύεται εύκολα.

2.3 Θεωρία των Μοντέλων Μηχανικής Μάθησης που χρησιμοποιήθηκαν στην εν λόγω εργασία

Η Μηχανική Μάθηση είναι πεδίο της Επιστήμης των Υπολογιστών, που αναπτύχθηκε από τη μελέτη της αναγνώρισης προτύπων και της υπολογιστικής θεωρίας μάθησης στην Τεχνητή νοημοσύνη. Τα τελευταία χρόνια αναπτύσσεται συνεχώς και έχει κάνει την εμφάνιση της σε αρκετούς τομείς, όπου η χρήση υπολογιστικών συστημάτων ήταν περιορισμένη ή ακόμα και ανύπαρκτη, όπως είναι ο έλεγχος παραγωγής εργοστασίων, η διαφήμιση, η αξιολόγηση των εργαζομένων και πολλά ακόμα. Δεν είναι τυχαίο που όλο και περισσότερες εταιρείες εντάσσουν τη Μηχανική Μάθηση στις λειτουργίες τους, ακολουθώντας τις τεχνολογικές εξελίξεις [22].

Ο ορισμός που πρότεινε ο Tom M. Mitchell [23] για τη Μηχανική Μάθηση είναι ο ακόλουθος: ¨Ένα πρόγραμμα υπολογιστή λέγεται ότι μαθαίνει από μια εμπειρία Ε ως προς μια κλάση εργασιών Τ και ένα μέτρο επίδοσης Ρ, αν η επίδοσή του σε εργασίες της κλάσης Τ, όπως αποτιμάται από το μέτρο Ρ, βελτιώνεται με την εμπειρία Ε¨, δηλαδή ένα υπολογιστικό σύστημα χρησιμοποιεί παλαιότερα δεδομένα και προσπαθεί να ¨μάθει¨ από αυτά τα δεδομένα, να αναγνωρίσει δηλαδή τα πρότυπα και τα μοτίβα που ακολουθούν αυτά, ώστε να ταξινομήσει νέα δεδομένα, να βρει ομοιότητες μεταξύ τους, να τα ομαδοποιήσει κλπ [22].

Υπάρχουν αρκετές κατηγορίες προβλημάτων Μηχανικής Μάθησης, ωστόσο η παρούσα εργασία θα ασχοληθεί αποκλειστικά με την επιβλεπόμενη μάθηση (supervised learning), όπου το υπολογιστικό σύστημα εκπαιδεύεται σε ένα σύνολο παραδειγμάτων, τα οποία συνοδεύονται με τις κατηγορίες-κλάσεις στις οποίες ανήκουν τα δεδομένα. Το σύστημα δηλαδή, χρησιμοποιεί την εμπειρία, παλιά δεδομένα τα οποία είναι διαθέσιμα, με σκοπό να προβλέψει μελλοντικές καταστάσεις, ταξινομώντας τα νέα δεδομένα στην κλάση που πιστεύει ότι ανήκουν. Να αναφερθεί εδώ, ότι γενικότερα στις προβλέψεις γίνεται η υπόθεση ότι το μοτίβο που επικρατεί στα μέχρι τώρα δεδομένα, ή έστω στα τελευταία, θα συνεχίσει να εμφανίζεται και στα επόμενα [22].

Επομένως, είναι πραγματικά δύσκολο να είναι κανείς σίγουρος για τις προβλέψεις του, καθώς μπορεί το μοτίβο που μέχρι τώρα εμφανιζόταν να αλλάξει ή και να εξαφανιστεί. Σε κάθε πρόβλημα της Μηχανικής Μάθησης υπάρχει και το σύνολο δεδομένων dataset με το οποίο θα εργαστεί κάποιος για την ανάπτυξη του συστήματος, που επιλύει το πρόβλημα. Για τη δημιουργία αυτού του συστήματος, το αρχικό dataset διαμερίζεται σε δύο υποσύνολα δεδομένων, το σύνολο δεδομένων εκπαίδευσης (training set), όπου το σύστημα θα προσπαθήσει να ¨μάθει¨ από τα δεδομένα και το σύνολο στο οποίο θα δοκιμαστεί η απόδοση του μοντέλου (test set). Με αυτό τον τρόπο σχεδιάζονται και εκπαιδεύονται οι περισσότεροι αλγόριθμοι Μηχανικής Μάθησης [22]. Η παρούσα εργασία χρησιμοποιεί δεδομένα από αγώνες του NBA, με σκοπό να προβλέψει το τελικό αποτέλεσμα επόμενων αγώνων του ίδιου Πρωταθλήματος.

Τα Μοντέλα Μηχανικής Μάθησης που χρησιμοποήθηκαν στην εν λόγω εργασία είναι τα εξής:

- 'lr' - Logistic Regression
- 'knn' - K Neighbors Classifier
- 'nb' - Naive Bayes
- 'dt' - Decision Tree Classifier
- 'svm' - SVM - Linear Kernel
- 'rbfsvm' - SVM - Radial Kernel
- 'gpc' - Gaussian Process Classifier
- 'mlp' - MLP Classifier
- 'ridge' - Ridge Classifier
- 'rf' - Random Forest Classifier
- 'qda' - Quadratic Discriminant Analysis
- 'ada' - Ada Boost Classifier
- 'gbc' - Gradient Boosting Classifier
- 'lda' - Linear Discriminant Analysis
- 'et' - Extra Trees Classifier
- 'xgboost' - Extreme Gradient Boosting
- 'lightgbm' - Light Gradient Boosting Machine
- 'catboost' - CatBoost Classifier

2.3.1 Τι είναι η λογιστική παλινδρόμηση ('lr' - Logistic Regression)

Αυτός ο τύπος στατιστικού μοντέλου (γνωστό και ως μοντέλο λογιστικής παλινδρόμησης) χρησιμοποιείται συχνά για ταξινόμηση και την προγνωστική ανάλυση. Η λογιστική παλινδρόμηση εκτιμά την πιθανότητα να συμβεί ένα γεγονός, όπως ¨ψηφίστηκε ή δεν ψηφίστηκε¨, με βάση ένα δεδομένο σύνολο ανεξάρτητων μεταβλητών. Εφόσον το αποτέλεσμα είναι μια πιθανότητα, η εξαρτημένη μεταβλητή οριοθετείται μεταξύ 0 και 1. Στη λογιστική παλινδρόμηση, εφαρμόζεται ένας μετασχηματισμός λόγων στις πιθανότητες - δηλαδή, η πιθανότητα επιτυχίας διαιρούμενη με την πιθανότητα αποτυχίας. Αυτό είναι επίσης κοινώς γνωστό ως log odds, ή φυσικός λογάριθμος των πιθανοτήτων, και αυτή η λογιστική συνάρτηση αντιπροσωπεύεται από τους ακόλουθους τύπους [24] :

$$Logit(pi) = \frac{1}{(1 + exp(-pi))}$$

$$ln(\frac{pi}{1-pi}) = Beta_0 + Beta_1 * X_1 + ... + B_k * K_k$$

Σε αυτήν την εξίσωση λογιστικής παλινδρόμησης, η logit(pi) είναι η εξαρτημένη μεταβλητή ή η μεταβλητή απόκρισης και το x είναι η ανεξάρτητη μεταβλητή. Η παράμετρος Beta, ή ο συντελεστής, σε αυτό το μοντέλο εκτιμάται συνήθως μέσω της εκτίμησης μέγιστης πιθανοφάνειας (MLE) [24]. Αυτή η μέθοδος δοκιμάζει διαφορετικές τιμές του Beta, μέσω πολλαπλών επαναλήψεων, για βελτιστοποίηση, για την καλύτερη προσαρμογή των λογαριθμικών πιθανοτήτων. Όλες αυτές οι επαναλήψεις παράγουν τη συνάρτηση λογαριθμικής πιθανότητας και η λογιστική παλινδρόμηση επιδιώκει να μεγιστοποιήσει αυτήν τη συνάρτηση για να βρει την καλύτερη εκτίμηση παραμέτρων. Μόλις βρεθεί ο βέλτιστος συντελεστής (ή οι συντελεστές εάν υπάρχουν περισσότερες από μία ανεξάρτητες μεταβλητές), οι πιθανότητες υπό όρους για κάθε παρατήρηση μπορούν να υπολογιστούν, να καταγραφούν και να αθροιστούν για να προκύψει μια προβλεπόμενη πιθανότητα. Για τη δυαδική ταξινόμηση, μια πιθανότητα μικρότερη από 0,5 θα προβλέψει το 0 ενώ μια πιθανότητα μεγαλύτερη από 0,5 θα προβλέψει το 1. Αφού υπολογιστεί το μοντέλο, είναι καλύτερη πρακτική να αξολογηθεί το πόσο καλά το μοντέλο προβλέπει την εξαρτημένη μεταβλητή, η οποία ονομάζεται goodness of fit. Η δοκιμή Hosmer–Lemeshow [25] είναι μια δημοφιλής μέθοδος για την αξιολόγηση της προσαρμογής του μοντέλου.

Οι λογαριθμικές πιθανότητες μπορεί να είναι δύσκολο να κατανοηθούν σε μια ανάλυση δεδομένων λογιστικής παλινδρόμησης. Ως αποτέλεσμα, η εκθετική εκτίμηση των εκτιμήσεων βήτα είναι συνηθισμένη για τη μετατροπή των αποτελεσμάτων σε αναλογία πιθανοτήτων (OR), διευκολύνοντας την ερμηνεία των αποτελεσμάτων. Το OR αντιπροσωπεύει τις πιθανότητες να προκύψει ένα αποτέλεσμα δεδομένου ενός συγκεκριμένου γεγονότος, σε σύγκριση με τις πιθανότητες να συμβεί το αποτέλεσμα απουσία αυτού του γεγονότος [24]. Εάν το OR είναι μεγαλύτερο από 1, τότε το συμβάν σχετίζεται με υψηλότερες πιθανότητες να δημιουργήσει ένα συγκεκριμένο αποτέλεσμα. Αντίθετα, εάν το OR είναι μικρότερο από 1, τότε το συμβάν σχετίζεται με μικρότερες πιθανότητες να συμβεί αυτό το αποτέλεσμα. Με βάση την παραπάνω εξίσωση, η ερμηνεία ενός λόγου πιθανοτήτων μπορεί να χαρακτηριστεί ως εξής: οι πιθανότητες επιτυχίας αλλάζουν κατά $exp(cB_1)$ φορές για κάθε αύξηση της μονάδας c σε x. Για να χρησιμοποιήσουμε ένα παράδειγμα, ας πούμε ότι έπρεπε να υπολογίσουμε τις πιθανότητες επιβίωσης στον Τιτανικό, δεδομένου ότι το άτομο ήταν άνδρας και η αναλογία πιθανοτήτων για τους άνδρες ήταν 0,0810. Θα ερμηνεύαμε τον λόγο πιθανοτήτων ως ¨οι πιθανότητες επιβίωσης των ανδρών μειώθηκαν κατά έναν παράγοντα 0,0810 σε σύγκριση με τις γυναίκες¨, διατηρώντας όλες τις άλλες μεταβλητές σταθερές [24].

Τόσο η γραμμική όσο και η λογιστική παλινδρόμηση είναι από τα πιο δημοφιλή μοντέλα στην επιστήμη δεδομένων και τα εργαλεία ανοιχτού κώδικα, όπως η Python και η R και κάνουν τον υπολογισμό για αυτές γρήγορο και εύκολο.

Τα μοντέλα γραμμικής παλινδρόμησης χρησιμοποιούνται για τον προσδιορισμό της σχέσης μεταξύ μιας συνεχούς εξαρτημένης μεταβλητής και μιας ή περισσότερων ανεξάρτητων μεταβλητών. Όταν υπάρχει μόνο μία ανεξάρτητη μεταβλητή και μία εξαρτημένη μεταβλητή, είναι γνωστή ως απλή γραμμική παλινδρόμηση, αλλά καθώς αυξάνεται ο αριθμός των ανεξάρτητων μεταβλητών, αναφέρεται ως πολλαπλή γραμμική παλινδρόμηση. Για κάθε τύπο

γραμμικής παλινδρόμησης, επιδιώκει να σχεδιάσει μια γραμμή βέλτιστης προσαρμογής μέσω ενός συνόλου σημείων δεδομένων, τα οποία συνήθως υπολογίζονται χρησιμοποιώντας τη μέθοδο των ελαχίστων τετραγώνων [24].

Παρόμοια με τη γραμμική παλινδρόμηση, η λογιστική παλινδρόμηση χρησιμοποιείται επίσης για την εκτίμηση της σχέσης μεταξύ μιας εξαρτημένης μεταβλητής και μιας ή περισσότερων ανεξάρτητων μεταβλητών, αλλά χρησιμοποιείται για να κάνει μια πρόβλεψη για μια κατηγορική μεταβλητή, έναντι μιας συνεχούς. Μια κατηγορική μεταβλητή μπορεί να είναι αληθής ή ψευδής, ναι ή όχι, 1 ή 0, κ.λπ. Η μονάδα μέτρησης διαφέρει επίσης από τη γραμμική παλινδρόμηση καθώς παράγει μια πιθανότητα, αλλά η συνάρτηση logit μετατρέπει την καμπύλη S σε ευθεία γραμμή [24].

Ενώ και τα δύο μοντέλα χρησιμοποιούνται στην ανάλυση παλινδρόμησης για να γίνουν προβλέψεις για μελλοντικά αποτελέσματα, η γραμμική παλινδρόμηση είναι συνήθως πιο κατανοητή. Η γραμμική παλινδρόμηση επίσης δεν απαιτεί τόσο μεγάλο μέγεθος δείγματος. όσο η λογιστική παλινδρόμηση χρειάζεται ένα επαρκές δείγμα για να αναπαραστήσει τιμές σε όλες τις κατηγορίες απόκρισης. Χωρίς ένα μεγαλύτερο, αντιπροσωπευτικό δείγμα, το μοντέλο μπορεί να μην έχει επαρκή στατιστική ισχύ για να ανιχνεύσει ένα σημαντικό αποτέλεσμα [24].

Σύμφωνα με την πηγή [24] υπάρχουν τρία είδη μοντέλων λογιστικής παλινδρόμησης, τα οποία ορίζονται με βάση την κατηγορική απόκριση.

- **Δυαδική λογιστική παλινδρόμηση:** Σε αυτήν την προσέγγιση, η απόκριση ή η εξαρτημένη μεταβλητή είναι διχοτομικής φύσης—δηλ. έχει μόνο δύο πιθανά αποτελέσματα (π.χ. 0 ή 1). Μερικά δημοφιλή παραδείγματα χρήσης του περιλαμβάνουν την πρόβλεψη εάν ένα μήνυμα ηλεκτρονικού ταχυδρομείου είναι ανεπιθύμητο ή όχι ανεπιθύμητο ή εάν ένας όγκος είναι κακοήθης ή όχι. Στο πλαίσιο της λογιστικής παλινδρόμησης, αυτή είναι η πιο συχνά χρησιμοποιούμενη προσέγγιση, και γενικότερα, είναι ένας από τους πιο συνηθισμένους ταξινομητές για δυαδική ταξινόμηση.

- **Πολυωνυμική λογιστική παλινδρόμηση (Multinomial):** Σε αυτόν τον τύπο μοντέλου λογιστικής παλινδρόμησης, η εξαρτημένη μεταβλητή έχει τρία ή περισσότερα πιθανά αποτελέσματα. Ωστόσο, αυτές οι τιμές δεν έχουν καθορισμένη σειρά. Για παράδειγμα, τα κινηματογραφικά στούντιο θέλουν να προβλέψουν ποιό είδος ταινίας είναι πιθανό να δει ένας θεατής του κινηματογράφου, για να προωθήσει τις ταινίες πιο αποτελεσματικά. Ένα πολυωνυμικό μοντέλο λογιστικής παλινδρόμησης μπορεί να βοηθήσει το στούντιο να προσδιορίσει τη δύναμη επιρροής που μπορεί να έχει η ηλικία, το φύλο και η κατάσταση γνωριμίας ενός ατόμου, στον τύπο της ταινίας που προτιμά. Το στούντιο μπορεί στη συνέχεια να προσανατολίσει μια διαφημιστική καμπάνια μιας συγκεκριμένης ταινίας προς μια ομάδα ανθρώπων που είναι πιθανό να πάνε να τη δουν.

- **Διατεταγμένη λογιστική παλινδρόμηση (Ordinal):** Αυτός ο τύπος μοντέλου λογιστικής παλινδρόμησης χρησιμοποιείται όταν η μεταβλητή απόκρισης έχει τρία ή περισσότερα πιθανά αποτελέσματα, αλλά στην περίπτωση αυτή, αυτές οι τιμές έχουν μια καθορισμένη σειρά. Παραδείγματα διατεταγμένων απαντήσεων περιλαμβάνουν κλίμακες βαθμολόγησης από το Α έως F ή κλίμακες βαθμολόγησης από το 1 έως το 5.

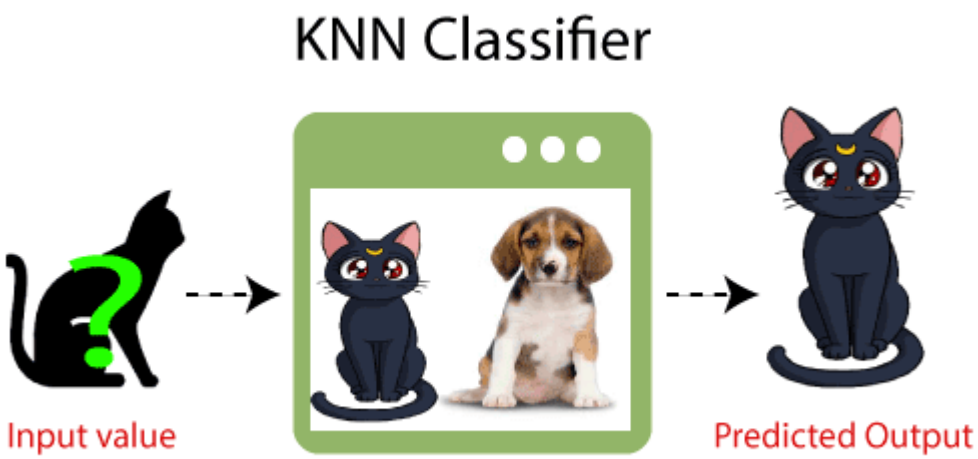
Στο πλαίσιο της μηχανικής μάθησης, η λογιστική παλινδρόμηση ανήκει στην οικογένεια των εποπτευόμενων μοντέλων μηχανικής μάθησης. Θεωρείται επίσης διακριτικό μοντέλο, που σημαίνει ότι επιχειρεί να διακρίνει μεταξύ τάξεων (ή κατηγοριών). Σε αντίθεση με έναν παραγωγικό αλγόριθμο, όπως ο naïve bayes, ο οποίος δεν μπορεί, όπως υπονοεί το όνομα, να δημιουργήσει πληροφορίες, όπως μια εικόνα για να την προβλέψει (π.χ. μια εικόνα μιας γάτας).

Προηγουμένως, αναφέρθηκε πώς η λογιστική παλινδρόμηση μεγιστοποιεί τη συνάρτηση λογαριθμικής πιθανοφάνειας για τον προσδιορισμό των συντελεστών βήτα του μοντέλου. Αυτό αλλάζει ελαφρώς στο πλαίσιο της μηχανικής μάθησης. Στο πλαίσιο της μηχανικής μάθησης, η αρνητική λογαριθμική πιθανοφάνεια χρησιμοποιείται ως συνάρτηση απώλειας, χρησιμοποιώντας τη διαδικασία της καθόδου κλίσης, για να βρεθεί το ολικό μέγιστο.

Η λογιστική παλινδρόμηση μπορεί επίσης να είναι επιρρεπής σε υπερπροσαρμογή, ιδιαίτερα όταν υπάρχει μεγάλος αριθμός μεταβλητών πρόβλεψης μέσα στο μοντέλο. Η κανονικοποίηση χρησιμοποιείται συνήθως για να ¨τιμωρήσει¨ τους μεγάλους συντελεστές των παραμέτρων, όταν το μοντέλο πάσχει από υψηλή διαστατικότητα [24].

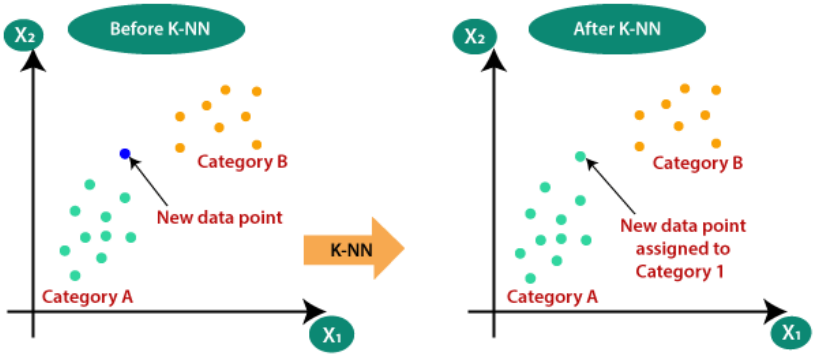
2.3.2 Τι είναι ο αλγόριθμος KNN ('knn' - K Neighbors Classifier)

Ο KNN είναι ένας από τους απλούστερους αλγορίθμους Μηχανικής Μάθησης που βασίζεται στην εποπτευόμενη μάθηση. Ο αλγόριθμος υποθέτει την ομοιότητα μεταξύ μιας νέας περίπτωσης δεδομένων και των διαθέσιμων περιπτώσεων και τοποθετεί τη νέα περίπτωση στην κατηγορία που μοιάζει περισσότερο στις διαθέσιμες κατηγορίες. Αποθηκεύει όλα τα διαθέσιμα δεδομένα και ταξινομεί ένα νέο σημείο δεδομένων με βάση την ομοιότητα. Αυτό σημαίνει ότι όταν εμφανίζονται νέα δεδομένα, τότε μπορούν εύκολα να ταξινομηθούν σε μια κατηγορία, με τη χρήση του αλγορίθμου KNN [3]. Η χρήση του αλγορίθμου είναι κατάλληλη και σε προβλήματα παλινδρόμισης, ωστόσο ενδείκνυται σε προβλήματα ταξινόμησης. Είναι μη παραμετρικός αλγόριθμος, που σημαίνει ότι δεν κάνει καμία υπόθεση για τα υποκείμενα δεδομένα. Ονομάζεται επίσης τεμπέλης μαθητευόμενος αλγόριθμος, καθώς δεν μαθαίνει αμέσως από το σύνολο εκπαίδευσης, αλλά αποθηκεύει το σύνολο δεδομένων και τη στιγμή της ταξινόμησης εκτελεί μια ενέργεια στο σύνολο δεδομένων. Ο αλγόριθμος KNN στη φάση εκπαίδευσης, απλώς αποθηκεύει το σύνολο δεδομένων και όταν λαμβάνει νέα δεδομένα, τότε ταξονιμεί αυτά τα δεδομένα σε μια κατηγορία που μοιάζει πολύ με τα νέα δεδομένα [3]. Για παραδάγμα, ας υποθέσουμε ότι έχουμε μια εικόνα ενός πλάσματος που μοιάζει με γάτα ή σκύλο και θέλουμε να μάθουμε εάν είναι γάτα ή σκύλος. Για αυτή την αναγώριση, μπορούμε με τη χρήση του KNN να αναγνωρίσουμε το σωστό αποτέλεσμα, καθώς χρησιμοποιεί ένα μέτρο ομοιότητας. Το μοντέλο μας θα εντοπίσει τα παρόμοια χαρακτηριστικά του νέου συνόλου δεδομένων για εικόνες γατών και σκύλων και με βάση τα πιο όμοια χαρακτηριστικά θα τα τοποθετήσει στην κατάλληλη κατηγορία [3].



Σχήμα 2.4: *KNN για αναγνώριση προτύπων [3].*

Ας υποθέσουμε ότι υπάρχουν δύο κατηγορίες, δηλαδή η Κατηγορία A και η Κατηγορία B, και έχουμε ένα νέο σημείο δεδομένων x1, οπότε αυτό το σημείο δεδομένων θα βρίσκεται σε κάποια από αυτές τις κατηγορίες. Για να λύσουμε αυτό το είδος προβλήματος, χρειαζόμαστε έναν αλγόριθμο KNN. Με τη βοήθεια του KNN, μπορούμε εύκολα να αναγνωρίσουμε την κατηγορία ή την κλάση ενός συγκεκριμένου συνόλου δεδομένων. Όπως για παράδειγμα στο παρακάτω διάγραμμα :



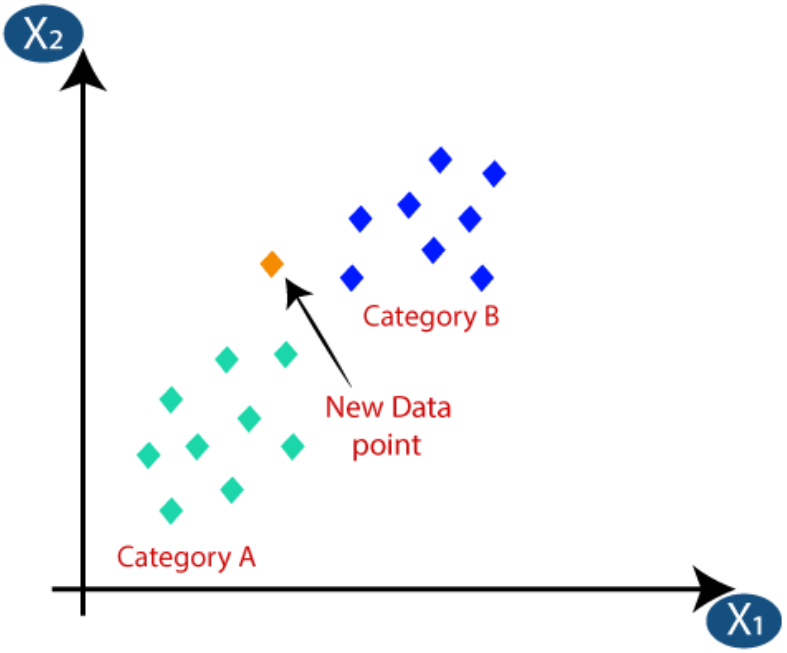
Σχήμα 2.5: *Κατηγοριοποίηση σημείου με KNN [3].*

Πώς λειτουργεί ο αλγόριθμος KNN

- Βήμα-1: Επιλέγουμε τον αριθμό K των γειτόνων
- Βήμα-2: Υπολογίζουμε την Ευκλείδεια απόσταση του K αριθμού γειτόνων
- Βήμα-3: Παίρνουμε τους K πλησιέστερους γείτονες σύμφωνα με την υπολογισμένη Ευκλείδεια απόσταση.
- Βήμα-4: Μεταξύ αυτών των K γειτόνων, μετράμε τον αριθμό των σημείων δεδομένων σε κάθε κατηγορία.

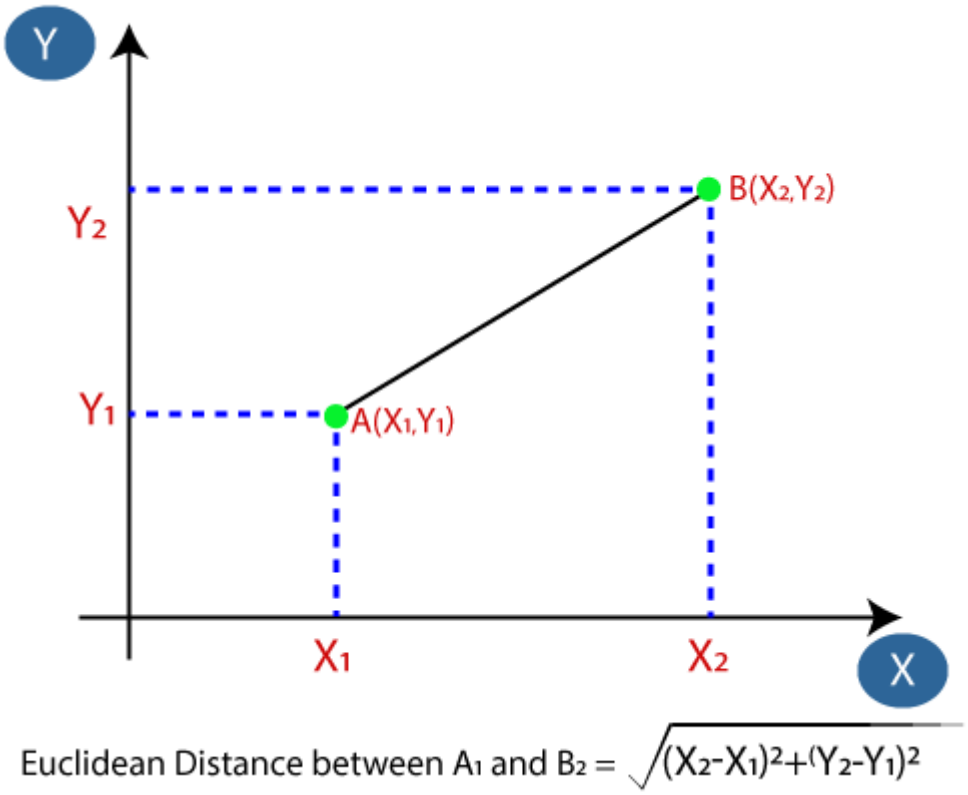
- Βήμα-5: Εκχωρούμε τα νέα σημεία δεδομένων σε εκείνη την κατηγορία για την οποία ο αριθμός του γείτονα είναι μέγιστος.
- Βήμα-6: Το μοντέλο είναι έτοιμο.

Ας υποθέσουμε ότι έχουμε ένα νέο σημείο δεδομένων και πρέπει να το βάλουμε στην απαιτούμενη κατηγορία, σύμφωνα με την παρακάτω εικόνα:



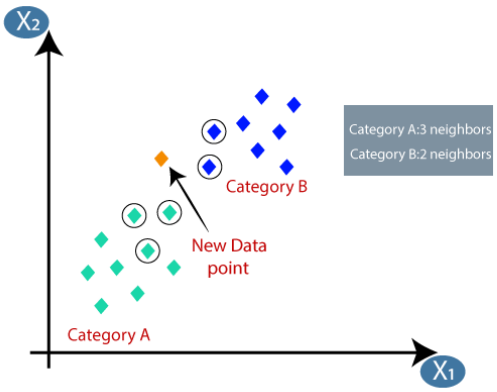
Σχήμα 2.6: *Κατηγοριοποίηση σημείου με KNN βήμα 1ο [3].*

Αρχικά, θα επιλέξουμε τον αριθμό των γειτόνων, άρα θα επιλέξουμε το K=5. Στη συνέχεια, θα υπολογίσουμε την Ευκλείδεια απόσταση μεταξύ των σημείων δεδομένων. Η Ευκλείδεια απόσταση είναι η απόσταση μεταξύ δύο σημείων, την οποία έχουμε ήδη μελετήσει στη γεωμετρία. Μπορεί να υπολογιστεί ως:



Σχήμα 2.7: *Ευκλείδια απόσταση δύο σημείων [3].*

Υπολογίζοντας την Ευκλείδεια απόσταση πήραμε τους πλησιέστερους γείτονες, ως τρεις πλησιέστερους γείτονες στην κατηγορία Α και δύο πλησιέστερους στην κατηγορία Β. Δηλαδή:



Σχήμα 2.8: *Κατηγοριοποίηση σημείου με KNN βήμα 2ο [3].*

Όπως μπορούμε να δούμε οι 3 πλησιέστεροι γείτονες είναι από την κατηγορία Α, επομένως αυτό το νέο σημείο δεδομένων πρέπει να ανήκει στην κατηγορία Α.

Πώς να επιλέξουμε την τιμή του K στον αλγόριθμο KNN

Δεν υπάρχει ιδιαίτερος τρόπος να προσδιορίσουμε την καλύτερη τιμή για το K, επομένως πρέπει να δοκιμάσουμε ορισμένες τιμές για να βρούμε την καλύτερη από αυτές. Η πιο προτιμώμενη τιμή για το K είναι το 5. Μια πολύ χαμηλή τιμή για το K, όπως K=1 ή K=2,

μπορεί να είναι θορυβώδης και να οδηγήσει σε επιδράσεις ακραίων τιμών στο μοντέλο. Οι μεγάλες τιμές για το K είναι καλές, αλλά μπορεί να βρεθούν κάποιες δυσκολίες.

- **Πλεονεκτήματα του αλγορίθμου KNN** :

  Είναι απλός στην εφαρμογή. Είναι ανθεκτικός στα θορυβώδη δεδομένα εκπαίδευσης. Μπορεί να είναι πιο αποτελεσματικός εάν τα δεδομένα εκπαίδευσης είναι μεγάλα.

- **Μειονεκτήματα του αλγορίθμου KNN** :

  Χρειάζεται πάντα να προσδιορίζεται η τιμή του K που μπορεί να είναι σύνθετη κάποια στιγμή. Το κόστος υπολογισμού είναι υψηλό λόγω του υπολογισμού της απόστασης μεταξύ των σημείων δεδομένων για όλα τα δείγματα εκπαίδευσης.

2.3.3 Τι είναι ο αλγόριθμος Naive Bayes ('nb' - Naive Bayes)

Ένας ταξινομητής είναι ένα μοντέλο μηχανικής μάθησης που χρησιμοποιείται για τη διάκριση διαφορετικών αντικειμένων με βάση ορισμένα χαρακτηριστικά.

Ένας ταξινομητής Naive Bayes είναι ένα μοντέλο μηχανικής μάθησης που χρησιμοποιείται για ταξινόμηση. Η ουσία του ταξινομητή βασίζεται στο θεώρημα Bayes [4].

$$P(A|B) = \frac{P(B|A) * P(A)}{P(B)}$$

Χρησιμοποιώντας το θεώρημα Bayes, μπορούμε να βρούμε την πιθανότητα να συμβεί το A, δεδομένου ότι έχει συμβεί το B. Εδώ, το B είναι η απόδειξη και το A είναι η υπόθεση. Η υπόθεση που γίνεται εδώ είναι ότι οι προβλέψεις/χαρακτηριστικά είναι ανεξάρτητα. Δηλαδή η παρουσία ενός συγκεκριμένου χαρακτηριστικού δεν επηρεάζει το άλλο. Εξ ου και η ονομασία αφελής [4].

Ας πάρουμε ένα παράδειγμα για να αποκτήσουμε καλύτερη διαίσθηση. Σκεφτείτε το πρόβλημα του παιχνιδιού γκολφ. Το σύνολο δεδομένων αναπαρίσταται όπως παρακάτω:

	OUTLOOK	TEMPERATURE	HUMIDITY	WINDY	PLAY GOLF
0	Rainy	Hot	High	False	No
1	Rainy	Hot	High	True	No
2	Overcast	Hot	High	False	Yes
3	Sunny	Mild	High	False	Yes
4	Sunny	Cool	Normal	False	Yes
5	Sunny	Cool	Normal	True	No
6	Overcast	Cool	Normal	True	Yes
7	Rainy	Mild	High	False	No
8	Rainy	Cool	Normal	False	Yes
9	Sunny	Mild	Normal	False	Yes
10	Rainy	Mild	Normal	True	Yes
11	Overcast	Mild	High	True	Yes
12	Overcast	Hot	Normal	False	Yes
13	Sunny	Mild	High	True	No

Πίνακας 2.1: *Πίνακας συνόλου δεδομένων παραδείγματος [4].*

Κατατάσσουμε αν η μέρα είναι κατάλληλη για παιχνίδι γκολφ, δεδομένων των χαρακτηριστικών της ημέρας. Οι στήλες αντιπροσωπεύουν αυτά τα χαρακτηριστικά και οι σειρές αντιπροσωπεύουν μεμονωμένες εγγραφές. Αν πάρουμε την πρώτη σειρά του συνόλου δεδομένων, μπορούμε να παρατηρήσουμε ότι δεν είναι κατάλληλο για παιχνίδι γκολφ, εάν η προοπτική είναι βροχερή, η θερμοκρασία είναι ζεστή, η υγρασία είναι υψηλή και δεν φυσάει. Κάνουμε δύο υποθέσεις εδώ, μία όπως αναφέρθηκε παραπάνω θεωρούμε ότι αυτοί οι προγνωστικοί παράγοντες είναι ανεξάρτητοι. Αν δηλαδή η θερμοκρασία είναι ζεστή, δεν σημαίνει απαραίτητα ότι η υγρασία είναι υψηλή. Μια άλλη υπόθεση που γίνεται εδώ είναι ότι όλοι οι προγνωστικοί παράγοντες έχουν ίση επίδραση στο αποτέλεσμα. Δηλαδή, η μέρα που φυσάει δεν έχει μεγαλύτερη σημασία στο να αποφασίσεις να παίξεις γκολφ ή όχι [4]. Από την ίδια πηγή έχουμε ότι :

Σύμφωνα με αυτό το παράδειγμα, το θεώρημα Bayes μπορεί να ξαναγραφτεί ως εξής:

$$P(y|X) = \frac{P(X|y) * P(y)}{P(X)}$$

Η μεταβλητή y είναι η μεταβλητή κλάσης (παίζω γκολφ), η οποία αντιπροσωπεύει αν είναι κατάλληλο να παίξεις γκολφ ή όχι με βάση τις συνθήκες. Η μεταβλητή X αντιπροσωπεύει τις παραμέτρους/χαρακτηριστικά.

Το X δίνεται ως,

$$X = (x_1, x_2, x_3, ..., x_n)$$

Εδώ τα x1, x2...xn αντιπροσωπεύουν τα χαρακτηριστικά, δηλαδή μπορούν να αντιστοιχιστούν σε προοπτικές, θερμοκρασία, υγρασία και άνεμο. Αντικαθιστώντας το X και επεκτείνοντας χρησιμοποιώντας τον κανόνα της αλυσίδας παίρνουμε,

$$P(y|x_1, x_2, x_3, ..., x_n) = \frac{P(x_1|y) * P(x_2|y)... * P(x_n|y) * P(y)}{P(x_1) * P(x_2)... * P(x_n)}$$

Τώρα, μπορεί να λάβει κανείς τις τιμές για το καθένα, κοιτάζοντας το σύνολο δεδομένων και να τις αντικαταστήσει στην εξίσωση. Για όλες τις καταχωρήσεις στο σύνολο δεδομένων, ο παρονομαστής δεν αλλάζει, παραμένει στατικός. Επομένως, μπορεί να αφαιρεθεί ο παρονομαστής και να εισαχθεί μια αναλογία.

$$P(y|x_1, x_2, x_3, ..., x_n) \infty P(y) \prod_{i=1}^{n} P(x_i|y)$$

Στην περίπτωσή μας, η μεταβλητή y κλάσης έχει μόνο δύο αποτελέσματα, ναι ή όχι. Θα μπορούσαν να υπάρξουν περιπτώσεις όπου η ταξινόμηση θα μπορούσε να είναι πολυμεταβλητή. Επομένως, πρέπει να βρούμε την κλάση y με μέγιστη πιθανότητα.

$$y = argmax_y P(y) \prod_{i=1}^{n} P(x_i|y)$$

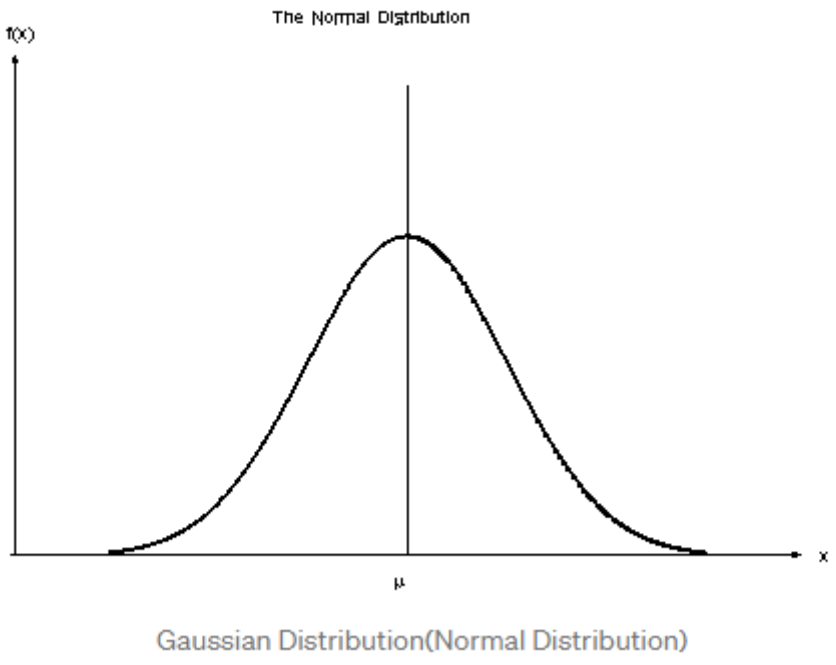
Χρησιμοποιώντας την παραπάνω συνάρτηση, μπορούμε να λάβουμε την κλάση, δεδομένων των προγνωστικών.

Τύποι ταξινομητή Naive Bayes:

- **Πολυωνυμικό Naive Bayes**: Αυτό χρησιμοποιείται κυρίως για πρόβλημα ταξινόμησης εγγράφων, δηλαδή εάν ένα έγγραφο ανήκει στην κατηγορία αθλημάτων, πολιτικής, τεχνολογίας κ.λπ. Τα χαρακτηριστικά/προγνωστικά που χρησιμοποιούνται από τον ταξινομητή είναι η συχνότητα των λέξεων που υπάρχουν στο έγγραφο [4].

- **Bernoulli Naive Bayes**: Αυτό είναι παρόμοιο με τα πολυωνυμικά naive bayes, αλλά οι προβλέψεις είναι δυαδικές μεταβλητές. Οι παράμετροι που χρησιμοποιούμε για να προβλέψουμε τη μεταβλητή κλάσης λαμβάνουν μόνο τιμές ναι ή όχι, για παράδειγμα εάν μια λέξη εμφανίζεται στο κείμενο ή όχι [4].

- **Gaussian Naive Bayes**: Όταν οι προβλέψεις καταλαμβάνουν μια συνεχή τιμή και δεν είναι διακριτές, υποθέτουμε ότι αυτές οι τιμές λαμβάνονται δείγμα από μια γκαουσιανή κατανομή [4].



Σχήμα 2.9: *Γκαουσσιανή-Κανονική κατανομή [4].*

Δεδομένου ότι ο τρόπος με τον οποίο υπάρχουν οι τιμές στο σύνολο δεδομένων αλλάζει, ο τύπος για την υπό όρους πιθανότητα αλλάζει σε:

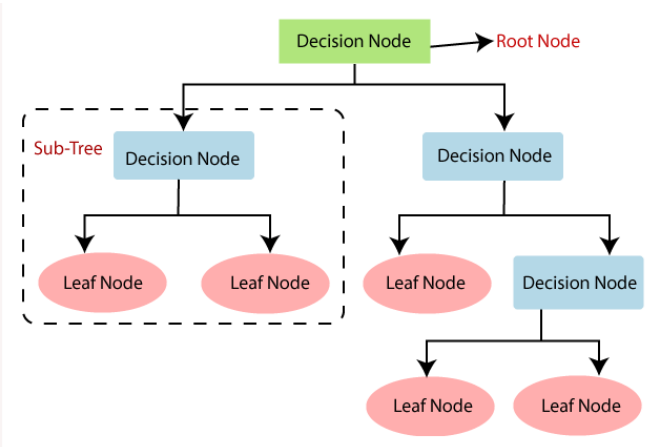
$$P(x_i|y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} exp(-\frac{(x_i - \mu_y)^2}{2\sigma_y^2})$$

Συμπέρασμα:

Οι αλγόριθμοι Naive Bayes χρησιμοποιούνται κυρίως σε ανάλυση συναισθημάτων, φιλτράρισμα ανεπιθύμητων μηνυμάτων, συστήματα συστάσεων κ.λπ. Είναι γρήγοροι και εύκολοι στην εφαρμογή τους, αλλά το μεγαλύτερο μειονέκτημά τους είναι ότι η απαίτηση των προγνωστικών να είναι ανεξάρτητα. Στις περισσότερες από τις πραγματικές περιπτώσεις, οι προγνωστικοί παράγοντες εξαρτώνται, αυτό εμποδίζει την απόδοση του ταξινομητή [4].

2.3.4 Τι είναι ο αλγόριθμος Decision Tree ('dt' - Decision Tree Classifier)

Το Decision Tree είναι μια εποπτευόμενη τεχνική μάθησης που μπορεί να χρησιμοποιηθεί τόσο για προβλήματα ταξινόμησης όσο και για προβλήματα παλινδρόμησης, αλλά κυρίως προτιμάται για την επίλυση προβλημάτων ταξινόμησης. Είναι ένας ταξινομητής με δομή δέντρου, όπου οι εσωτερικοί κόμβοι αντιπροσωπεύουν τα χαρακτηριστικά ενός συνόλου δεδομένων, τα κλαδιά αντιπροσωπεύουν τους κανόνες απόφασης και κάθε κόμβος φύλλου αντιπροσωπεύει το αποτέλεσμα. Σε ένα δέντρο απόφασης, υπάρχουν δύο κόμβοι, οι οποίοι είναι ο κόμβος απόφασης και ο κόμβος φύλλου. Οι κόμβοι απόφασης χρησιμοποιούνται για τη λήψη οποιασδήποτε απόφασης και έχουν πολλαπλά κλαδιά, ενώ οι κόμβοι φύλλων είναι το αποτέλεσμα αυτών των αποφάσεων και δεν περιέχουν περαιτέρω κλαδιά [5]. Οι αποφάσεις ή η δοκιμή εκτελούνται με βάση τα χαρακτηριστικά του δεδομένου συνόλου δεδομένων. Είναι μια γραφική αναπαράσταση για τη λήψη όλων των πιθανών λύσεων σε ένα πρόβλημα/απόφαση με βάση δεδομένες συνθήκες. Ονομάζεται δέντρο απόφασης επειδή, παρόμοια με ένα δέντρο, ξεκινά με τον κόμβο ρίζας, ο οποίος επεκτείνεται σε περαιτέρω κλαδιά και κατασκευάζει μια δομή που μοιάζει με δέντρο. Για να δημιουργήσουμε ένα δέντρο, χρησιμοποιούμε τον αλγόριθμο CART, ο οποίος σημαίνει αλγόριθμος Classification and Regression Tree. Ένα δέντρο απόφασης κάνει απλώς μια ερώτηση και με βάση την απάντηση (Ναι/Όχι), χωρίζει περαιτέρω το δέντρο σε υποδέντρα [5]. Το παρακάτω διάγραμμα εξηγεί τη γενική δομή ενός δέντρου αποφάσεων:



Σχήμα 2.10: *Προσομοίωση Πύλης NOR [5].*

Γιατί να χρησιμοποιήσει κανείς τα Δέντρα απόφασης

Υπάρχουν διάφοροι αλγόριθμοι στη Μηχανική μάθηση, επομένως η επιλογή του καλύτερου αλγόριθμου για το δεδομένο σύνολο δεδομένων και το πρόβλημα είναι τα κύρια σημεία που πρέπει να θυμάται κανείς κατά τη δημιουργία ενός μοντέλου μηχανικής μάθησης [5].

Παρατίθενται οι δύο λόγοι για τη χρήση του δέντρου απόφασης:

Τα δέντρα απόφασης συνήθως μιμούνται την ανθρώπινη ικανότητα σκέψης κατά τη λήψη μιας απόφασης, επομένως είναι εύκολο να γίνει κατανοητό. Η λογική πίσω από το δέντρο αποφάσεων μπορεί να γίνει εύκολα κατανοητή επειδή δείχνει μια δομή που μοιάζει με δέντρο.

Ορολογίες Δένδρου Αποφάσεων

- **Κόμβος ρίζας**: Ο κόμβος ρίζας είναι το σημείο από όπου ξεκινά το δέντρο αποφάσεων. Αντιπροσωπεύει ολόκληρο το σύνολο δεδομένων, το οποίο περαιτέρω χωρίζεται σε δύο ή περισσότερα ομοιογενή σύνολα.
- **Κόμβος φύλλου**: Οι κόμβοι φύλλων είναι ο τελικός κόμβος εξόδου και το δέντρο δεν μπορεί να διαχωριστεί περαιτέρω, μετά τη λήψη ενός κόμβου φύλλου.
- **Διαίρεση**: Ο διαχωρισμός είναι η διαδικασία διαίρεσης του κόμβου απόφασης/κόμβου ρίζας σε υποκόμβους σύμφωνα με τις δεδομένες συνθήκες.
- **Κλάδος/Υποδέντρο**: Ένα δέντρο που σχηματίζεται από το σχίσιμο του δέντρου.
- **Κλάδεμα**: Το κλάδεμα είναι η διαδικασία αφαίρεσης των ανεπιθύμητων κλαδιών από το δέντρο.
- **Κόμβος Γονέας/Παιδί**: Ο ριζικός κόμβος του δέντρου ονομάζεται γονικός κόμβος και άλλοι κόμβοι ονομάζονται θυγατρικοί κόμβοι.

Πώς λειτουργεί ο αλγόριθμος Decision Tree

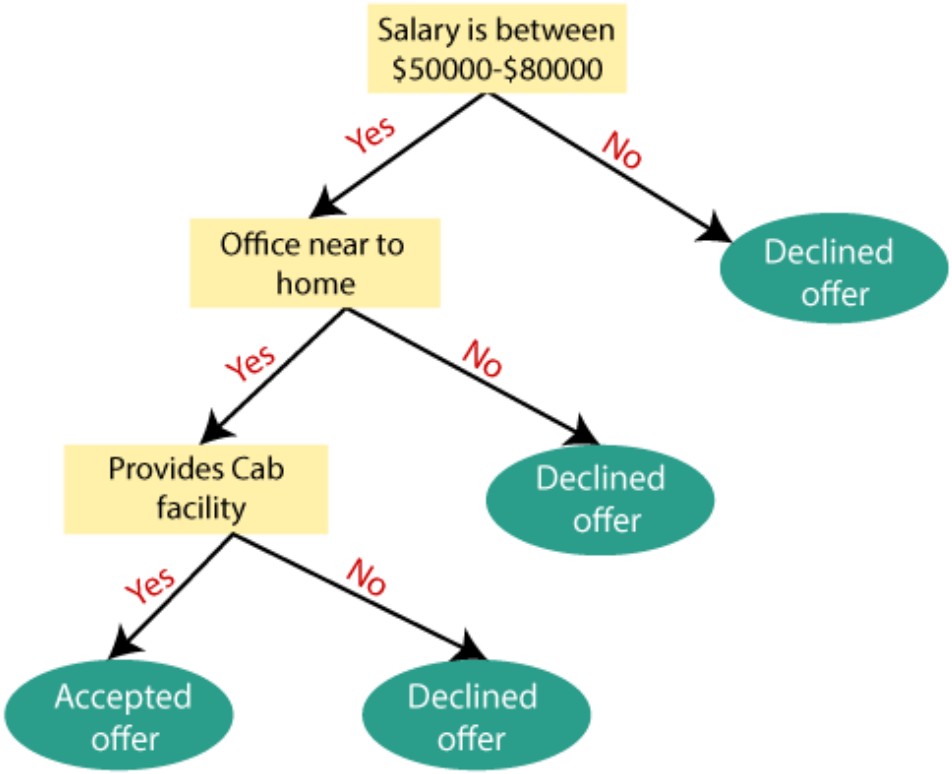
Σε ένα δέντρο αποφάσεων, για την πρόβλεψη της κλάσης του δεδομένου συνόλου δεδομένων, ο αλγόριθμος ξεκινά από τον κόμβο ρίζας του δέντρου. Αυτός ο αλγόριθμος συγκρίνει τις τιμές του χαρακτηριστικού root με το χαρακτηριστικό record (πραγματικό σύνολο δεδομένων) και, με βάση τη σύγκριση, ακολουθεί το κλαδί και μεταβαίνει στον επόμενο κόμβο [5].

Για τον επόμενο κόμβο, ο αλγόριθμος συγκρίνει ξανά την τιμή του χαρακτηριστικού με τους άλλους υπο-κόμβους και προχωρά περαιτέρω. Συνεχίζει τη διαδικασία μέχρι να φτάσει στον κόμβο φύλλου του δέντρου. Η πλήρης διαδικασία μπορεί να γίνει καλύτερα κατανοητή χρησιμοποιώντας τον παρακάτω αλγόριθμο [5]:

- **Βήμα-1**: Ξεκινήστε το δέντρο με τον ριζικό κόμβο, S, ο οποίος περιέχει το πλήρες σύνολο δεδομένων.
- **Βήμα 2**: Βρείτε το καλύτερο χαρακτηριστικό στο σύνολο δεδομένων χρησιμοποιώντας το Μέτρο Επιλογής Χαρακτηριστικού (ASM).
- **Βήμα-3**: Διαιρέστε το S σε υποσύνολα που περιέχουν πιθανές τιμές για τα καλύτερα χαρακτηριστικά.
- **Βήμα-4**: Δημιουργήστε τον κόμβο δέντρου αποφάσεων, ο οποίος περιέχει το καλύτερο χαρακτηριστικό.

- **Βήμα-5**: Δημιουργήστε αναδρομικά νέα δέντρα αποφάσεων χρησιμοποιώντας τα υποσύνολα του συνόλου δεδομένων που δημιουργήθηκαν στο βήμα -3. Συνεχίστε αυτή τη διαδικασία μέχρι να φτάσετε σε ένα στάδιο όπου δεν μπορείτε να ταξινομήσετε περαιτέρω τους κόμβους και να καλέσετε τον τελικό κόμβο ως κόμβο φύλλου.

Ας υποθέσουμε ότι υπάρχει ένας υποψήφιος που έχει μια προσφορά εργασίας και θέλει να αποφασίσει αν θα αποδεχτεί την προσφορά ή όχι. Έτσι, για να λυθεί αυτό το πρόβλημα, το δέντρο αποφάσεων ξεκινά με τον ριζικό κόμβο (χαρακτηριστικό μισθού κατά ASM). Ο ριζικός κόμβος χωρίζεται περαιτέρω στον επόμενο κόμβο απόφασης (απόσταση από το γραφείο) και σε έναν κόμβο φύλλου με βάση τις αντίστοιχες ετικέτες. Ο επόμενος κόμβος απόφασης χωρίζεται περαιτέρω σε έναν κόμβο απόφασης (εγκατάσταση Cab) και έναν κόμβο φύλλου. Τέλος, ο κόμβος απόφασης χωρίζεται σε δύο κόμβους φύλλων (Αποδεκτές προσφορές και Απορριφθείσες προσφορές), όπως στο παρακάτω διάγραμμα [5]:



Σχήμα 2.11: *Παράδειγμα Δέντρου αποφάσεων [5].*

Μέτρα επιλογής χαρακτηριστικών

Κατά την υλοποίηση ενός δέντρου απόφασης, το κύριο ζήτημα αφορά το πώς να επιλέξει κανείς το καλύτερο χαρακτηριστικό για τον ριζικό κόμβο και για τους υποκόμβους. Έτσι, για την επίλυση τέτοιων προβλημάτων υπάρχει μια τεχνική που ονομάζεται μέτρο επιλογής χαρακτηριστικών ή ASM [5]. Με αυτή τη μέτρηση, μπορούμε εύκολα να επιλέξουμε το καλύτερο χαρακτηριστικό για τους κόμβους του δέντρου. Υπάρχουν δύο δημοφιλείς τεχνικές για το ASM, οι οποίες είναι:

- Κέρδος πληροφοριών (Information Gain)
- Δείκτης Τζίνι (Gini Index)

1. Κέρδος πληροφοριών:

Το κέρδος πληροφοριών είναι η μέτρηση των αλλαγών στην εντροπία μετά την τμηματοποίηση ενός συνόλου δεδομένων με βάση ένα χαρακτηριστικό. Υπολογίζει πόσες πληροφορίες μας παρέχει ένα χαρακτηριστικό για μια τάξη. Σύμφωνα με την τιμή του κέρδους πληροφοριών, χωρίζουμε τον κόμβο και κατασκευάζουμε το δέντρο αποφάσεων. Ένας αλγόριθμος δέντρου αποφάσεων προσπαθεί πάντα να μεγιστοποιήσει την τιμή του κέρδους πληροφοριών και ένας κόμβος/χαρακτηριστικό που έχει το υψηλότερο κέρδος πληροφοριών διαχωρίζεται πρώτα. Μπορεί να υπολογιστεί χρησιμοποιώντας τον παρακάτω τύπο [5]:

$$InformationGain = Entropy(S) - [(WeightedAvg) * Entropy(eachfeature)]$$

Εντροπία: Η εντροπία είναι μια μέτρηση για τη μέτρηση της ακαθαρσίας σε ένα δεδομένο χαρακτηριστικό. Καθορίζει την τυχαιότητα στα δεδομένα. Η εντροπία μπορεί να υπολογιστεί ως [5]:

$$Entropy(s) = -P(yes) * log_2 P(yes) - P(no) * log_2 P(no)$$

όπου,

- S = Συνολικός αριθμός δειγμάτων
- P(yes) = Πιθανότητα του Ναι
- P(no) = Πιθανότητα του Όχι

2. Δείκτης Τζίνι: Ο δείκτης Gini είναι ένα μέτρο ακαθαρσίας ή καθαρότητας που χρησιμοποιείται κατά τη δημιουργία ενός δέντρου αποφάσεων στον αλγόριθμο CART (Classification and Regression Tree). Ένα χαρακτηριστικό με χαμηλό δείκτη Gini θα πρέπει να προτιμάται σε σύγκριση με τον υψηλό δείκτη Gini [5]. Δημιουργεί μόνο δυαδικούς διαχωρισμούς και ο αλγόριθμος CART χρησιμοποιεί τον δείκτη Gini για τη δημιουργία δυαδικών διαχωρισμών. Ο δείκτης Gini μπορεί να υπολογιστεί χρησιμοποιώντας τον παρακάτω τύπο:

$$GiniIndex = 1 - \Sigma_j P_j^2$$

Κλάδεμα: Εύρεση ενός βέλτιστου δέντρου απόφασης

Το κλάδεμα είναι μια διαδικασία διαγραφής των περιττών κόμβων από ένα δέντρο προκειμένου να ληφθεί το βέλτιστο δέντρο αποφάσεων.

Ένα πολύ μεγάλο δέντρο αυξάνει τον κίνδυνο υπερβολικής προσαρμογής και ένα μικρό δέντρο μπορεί να μην καταγράφει όλα τα σημαντικά χαρακτηριστικά του συνόλου δεδομένων. Επομένως, μια τεχνική που μειώνει το μέγεθος του δέντρου εκμάθησης χωρίς να μειώνει την ακρίβεια είναι γνωστή ως Κλάδεμα. Υπάρχουν κυρίως δύο τύποι τεχνολογίας κλαδέματος δέντρων που χρησιμοποιούνται [5]:

- Κλάδεμα πολυπλοκότητας κόστους
- Μειωμένο Κλάδεμα Σφάλματος

Πλεονεκτήματα του Δέντρου Αποφάσεων

- Είναι εύκολο να γίνει κατανοητό, καθώς ακολουθεί την ίδια διαδικασία που ακολουθεί ένας άνθρωπος ενώ παίρνει οποιαδήποτε απόφαση στην πραγματική ζωή.
- Μπορεί να είναι πολύ χρήσιμο για την επίλυση προβλημάτων που σχετίζονται με αποφάσεις.
- Βοηθά να σκεφτόμαστε όλα τα πιθανά αποτελέσματα για ένα πρόβλημα.
- Υπάρχει λιγότερη απαίτηση καθαρισμού δεδομένων σε σύγκριση με άλλους αλγόριθμους.

Μειονεκτήματα του δέντρου απόφασης

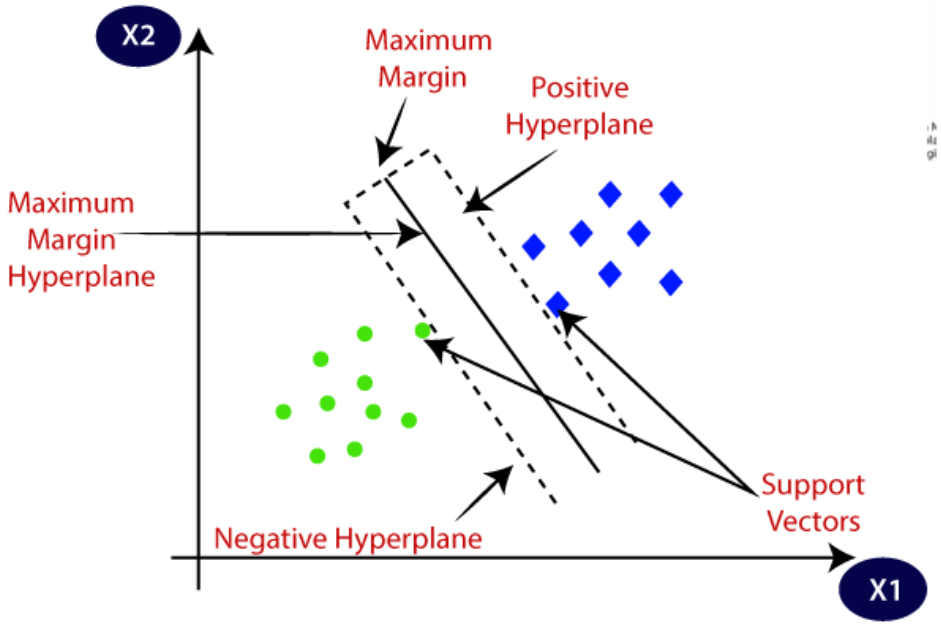
- Το δέντρο αποφάσεων περιέχει πολλά επίπεδα, γεγονός που το καθιστά περίπλοκο.
- Μπορεί να έχει πρόβλημα υπερπροσαρμογής, το οποίο μπορεί να επιλυθεί χρησιμοποιώντας τον αλγόριθμο Random Forest.
- Για περισσότερες ετικέτες κλάσεων, η υπολογιστική πολυπλοκότητα του δέντρου αποφάσεων μπορεί να αυξηθεί.

2.3.5 Τι είναι ο αλγόριθμος Support Vector Machine με Linear kernel ('svm' - SVM - Linear Kernel)

Το Support Vector Machine ή SVM είναι ένας από τους πιο δημοφιλείς αλγόριθμους εποπτευόμενης μάθησης, ο οποίος χρησιμοποιείται για προβλήματα ταξινόμησης και παλινδρόμησης. Ωστόσο, κατά κύριο λόγο, χρησιμοποιείται για προβλήματα ταξινόμησης στη Μηχανική Μάθηση [6].

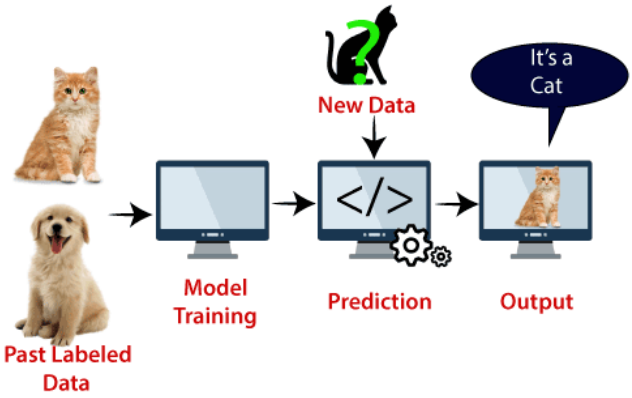
Ο στόχος του αλγόριθμου SVM είναι να δημιουργήσει την καλύτερη γραμμή ή όριο απόφασης που μπορεί να διαχωρίσει το χώρο n-διαστάσεων σε κλάσεις, ώστε να μπορούμε εύκολα να βάλουμε το νέο σημείο δεδομένων στη σωστή κατηγορία στο μέλλον. Αυτό το όριο καλύτερης απόφασης ονομάζεται υπερεπίπεδο.

Το SVM επιλέγει τα ακραία σημεία/διανύσματα που βοηθούν στη δημιουργία του υπερεπίπεδου. Αυτές οι ακραίες περιπτώσεις ονομάζονται διανύσματα υποστήριξης και ως εκ τούτου ο αλγόριθμος ονομάζεται Support Vector Machine [6]. Ας εξετάσουμε το παρακάτω διάγραμμα στο οποίο υπάρχουν δύο διαφορετικές κατηγορίες που ταξινομούνται χρησιμοποιώντας ένα όριο απόφασης ή ένα υπερεπίπεδο:



Σχήμα 2.12: *Ανάλυση ενός SVM [6].*

Παράδειγμα [6]: Το SVM μπορεί να γίνει κατανοητό με το παράδειγμα που χρησιμοποιήσαμε στον ταξινομητή KNN. Ας υποθέσουμε ότι βλέπουμε μια παράξενη γάτα που έχει επίσης κάποια χαρακτηριστικά σκύλων, οπότε αν θέλουμε ένα μοντέλο που μπορεί να προσδιορίσει με ακρίβεια αν είναι γάτα ή σκύλος, τότε ένα τέτοιο μοντέλο μπορεί να δημιουργηθεί χρησιμοποιώντας τον αλγόριθμο SVM. Πρώτα θα εκπαιδεύσουμε το μοντέλο μας με πολλές εικόνες γατών και σκύλων, ώστε να μπορεί να μάθει για τα διαφορετικά χαρακτηριστικά των γατών και των σκύλων και μετά θα το δοκιμάσουμε με αυτό το άγνωστο πλάσμα. Έτσι, καθώς το διάνυσμα υποστήριξης δημιουργεί ένα όριο απόφασης μεταξύ αυτών των δύο δεδομένων (γάτας και σκύλου) και επιλέγει ακραίες περιπτώσεις (διανύσματα υποστήριξης), θα δει την ακραία περίπτωση γάτας και σκύλου. Με βάση τα διανύσματα υποστήριξης, θα την ταξινομήσει ως γάτα. Σύμφωνα με το παρακάτω διάγραμμα:



Σχήμα 2.13: *Αναγνώριση μιας γάτας με SVM [6].*

Ο αλγόριθμος SVM μπορεί να χρησιμοποιηθεί για ανίχνευση προσώπου, ταξινόμηση εικόνων, κατηγοριοποίηση κειμένου κ.λπ.

Τύποι SVM

Τα SVM μπορεί να είναι δύο τύπων [6]:

- **Γραμμικό SVM**: Το γραμμικό SVM χρησιμοποιείται για γραμμικά διαχωρίσιμα δεδομένα, πράγμα που σημαίνει ότι εάν ένα σύνολο δεδομένων μπορεί να ταξινομηθεί σε δύο κατηγορίες χρησιμοποιώντας μια ευθεία γραμμή, τότε αυτά τα δεδομένα ονομάζονται γραμμικά διαχωρίσιμα δεδομένα και ο ταξινομητής χρησιμοποιείται ως Γραμμικός ταξινομητής SVM.

- **Μη γραμμικό SVM**: Το μη γραμμικό SVM χρησιμοποιείται για μη γραμμικά διαχωρισμένα δεδομένα, πράγμα που σημαίνει ότι εάν ένα σύνολο δεδομένων δεν μπορεί να ταξινομηθεί χρησιμοποιώντας μια ευθεία γραμμή, τότε αυτά τα δεδομένα ονομάζονται μη γραμμικά δεδομένα και ο ταξινομητής που χρησιμοποιείται ονομάζεται Μη-γραμμικός ταξινομητής SVM.

Υπερεπίπεδο και διανύσματα υποστήριξης στον αλγόριθμο SVM [6]:

- **Υπερεπίπεδο:** Μπορεί να υπάρχουν πολλές γραμμές/όρια απόφασης για να διαχωριστούν οι κλάσεις σε ν-διάστατο χώρο, αλλά πρέπει να βρούμε το καλύτερο όριο απόφασης που βοηθά στην ταξινόμηση των σημείων δεδομένων. Αυτό το καλύτερο όριο είναι γνωστό ως υπερεπίπεδο του SVM.

  Οι διαστάσεις του υπερεπίπεδου εξαρτώνται από τα χαρακτηριστικά που υπάρχουν στο σύνολο δεδομένων, πράγμα που σημαίνει ότι εάν υπάρχουν 2 χαρακτηριστικά (όπως φαίνεται στην εικόνα), τότε το υπερεπίπεδο θα είναι μια ευθεία γραμμή. Και αν υπάρχουν 3 χαρακτηριστικά, τότε το υπερεπίπεδο θα είναι ένα επίπεδο 2 διαστάσεων.

  Δημιουργούμε πάντα ένα υπερεπίπεδο που έχει μέγιστο περιθώριο, που σημαίνει τη μέγιστη απόσταση μεταξύ των σημείων δεδομένων.

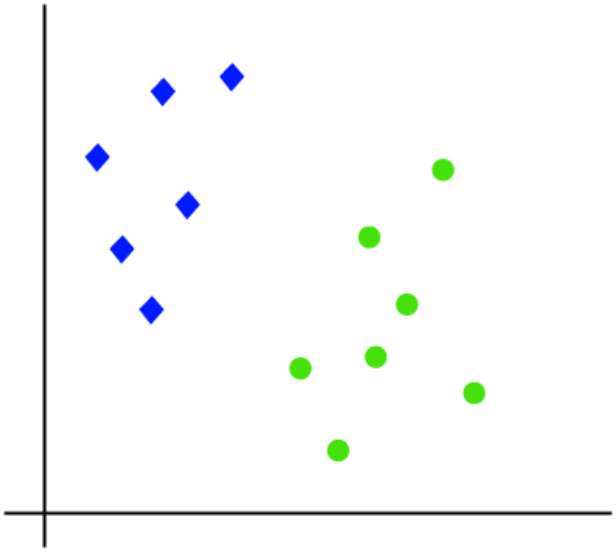
- **Διανύσματα υποστήριξης:**

  Τα σημεία δεδομένων ή διανύσματα που είναι πιο κοντά στο υπερεπίπεδο και τα οποία επηρεάζουν τη θέση του υπερεπίπεδου ονομάζονται διανύσματα υποστήριξης. Δεδομένου ότι αυτά τα διανύσματα υποστηρίζουν το υπερεπίπεδο, επομένως ονομάζεται διάνυσμα υποστήριξης.

Πώς λειτουργεί το SVM

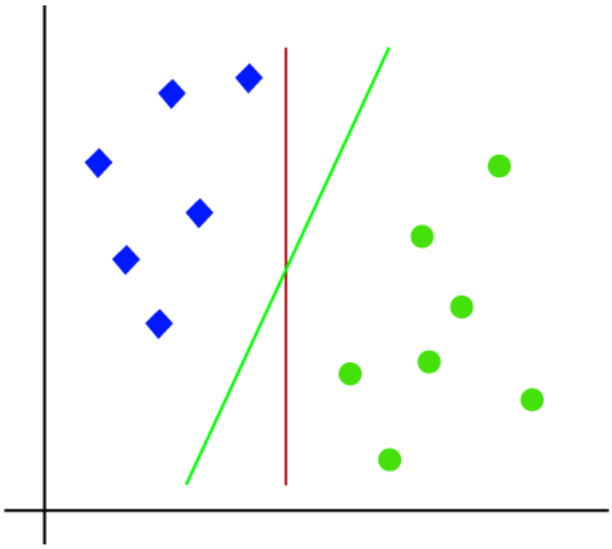
Γραμμικό SVM:

Η λειτουργία του αλγορίθμου SVM μπορεί να γίνει κατανοητή χρησιμοποιώντας ένα παράδειγμα. Ας υποθέσουμε ότι έχουμε ένα σύνολο δεδομένων που έχει δύο ετικέτες (πράσινη και μπλε) και το σύνολο δεδομένων έχει δύο χαρακτηριστικά x1 και x2. Θέλουμε έναν ταξινομητή που να μπορεί να ταξινομήσει το ζεύγος (x1, x2) των συντεταγμένων είτε σε πράσινο είτε σε μπλε. Σύμφωνα με την παρακάτω εικόνα:



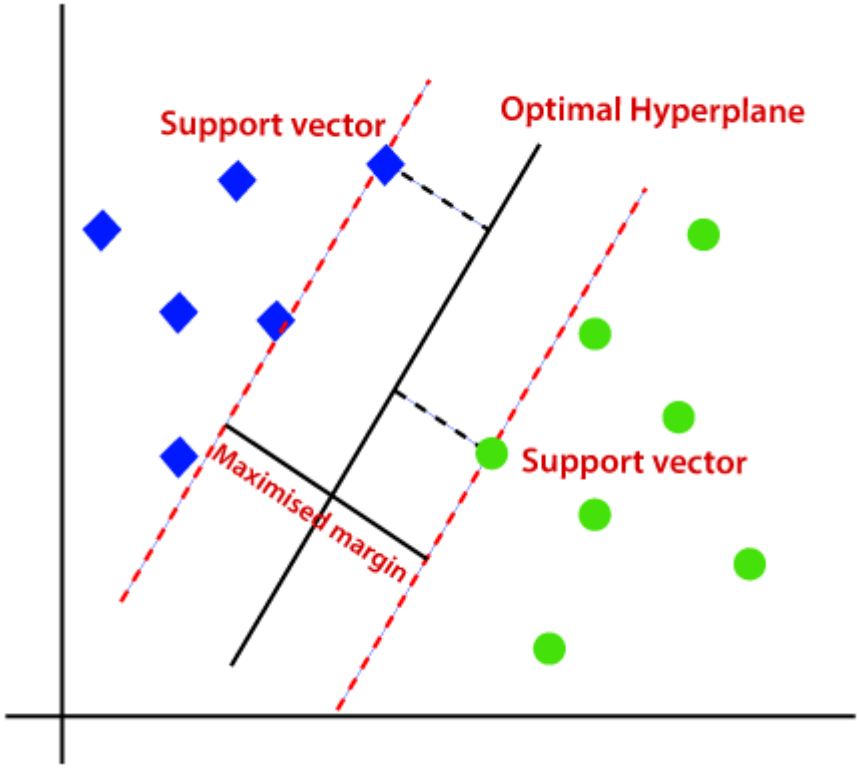
Σχήμα 2.14: *Ταξινόμηση χρωμάτων με SVM [6].*

Έτσι, καθώς είναι 2-d διάστημα, έτσι απλά χρησιμοποιώντας μια ευθεία γραμμή, μπορούμε εύκολα να διαχωρίσουμε αυτές τις δύο κατηγορίες. Αλλά μπορεί να υπάρχουν πολλές γραμμές που μπορούν να διαχωρίσουν αυτές τις κλάσεις. Σύμφωνα με την παρακάτω εικόνα :



Σχήμα 2.15: *Εύρεση βέλτιστης διαχωριστικής γραμμής - υπερεπιπέδου [6].*

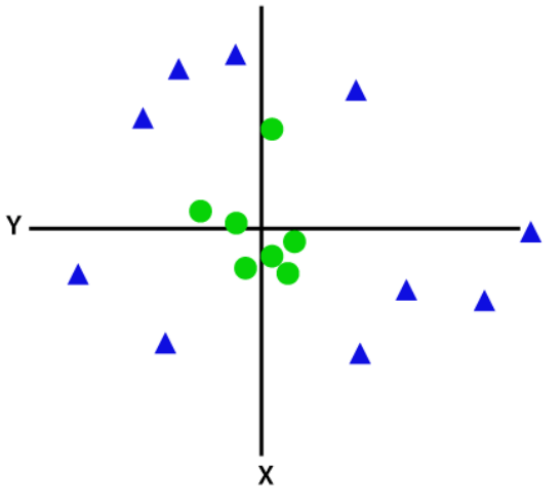
Ως εκ τούτου, ο αλγόριθμος SVM βοηθά στην εύρεση της καλύτερης γραμμής ή ορίου απόφασης. Αυτό το καλύτερο όριο ή περιοχή ονομάζεται υπερεπίπεδο. Ο αλγόριθμος SVM βρίσκει το πλησιέστερο σημείο των γραμμών και από τις δύο κλάσεις. Αυτά τα σημεία ονομάζονται διανύσματα υποστήριξης. Η απόσταση μεταξύ των διανυσμάτων και του υπερεπίπεδου ονομάζεται περιθώριο. Και ο στόχος του SVM είναι να μεγιστοποιήσει αυτό το περιθώριο. Το υπερεπίπεδο με μέγιστο περιθώριο ονομάζεται βέλτιστο υπερεπίπεδο.



Σχήμα 2.16: *Μεγιστοποίηση περιθωρίου - βέλτιστο υπερπίπεδο [6].*

Μη Γραμμικό SVM:

Εάν τα δεδομένα είναι γραμμικά διατεταγμένα, τότε μπορούμε να τα διαχωρίσουμε χρησιμοποιώντας μια ευθεία γραμμή, αλλά για τα μη γραμμικά δεδομένα, δεν μπορούμε να σχεδιάσουμε μια ευθεία γραμμή. Σύμφωνα με την παρακάτω εικόνα:

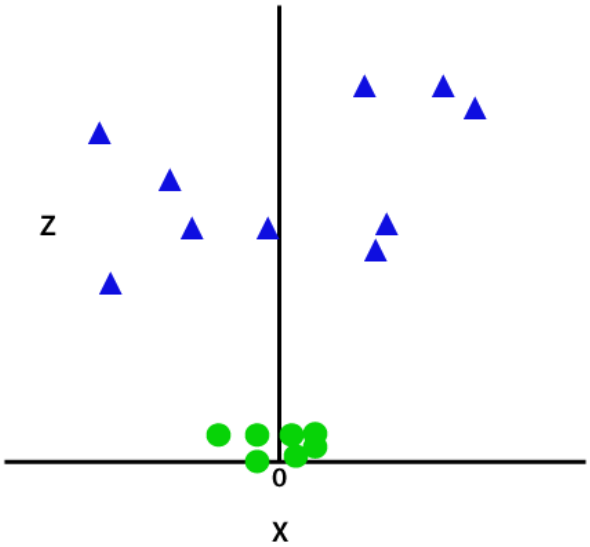


Σχήμα 2.17: *Μη γραμμικά διαχωρίσιμα δεδομένα [6].*

Για να διαχωρίσουμε λοιπόν αυτά τα σημεία δεδομένων, πρέπει να προσθέσουμε μια ακόμη διάσταση. Για γραμμικά δεδομένα, χρησιμοποιήσαμε δύο διαστάσεις x και y, επομένως για μη γραμμικά δεδομένα, θα προσθέσουμε μια τρίτη διάσταση z. Μπορεί να υπολογιστεί ως:

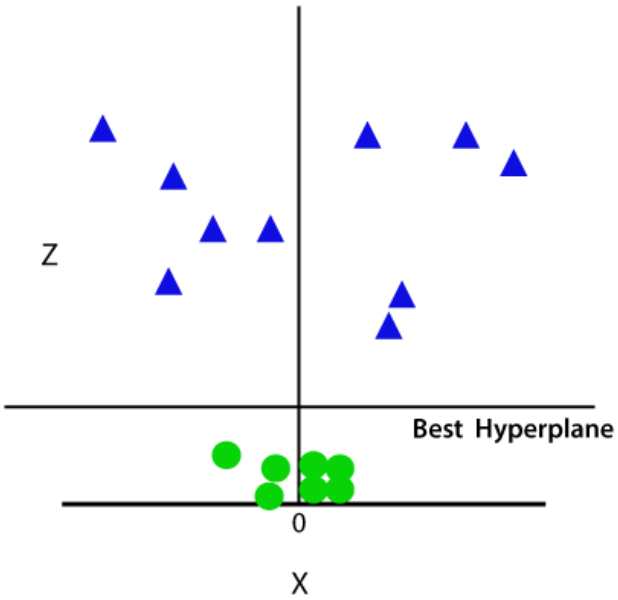
$$z = x^2 + y^2$$

Προσθέτοντας την τρίτη διάσταση, ο χώρος του δείγματος θα γίνει όπως η παρακάτω εικόνα:



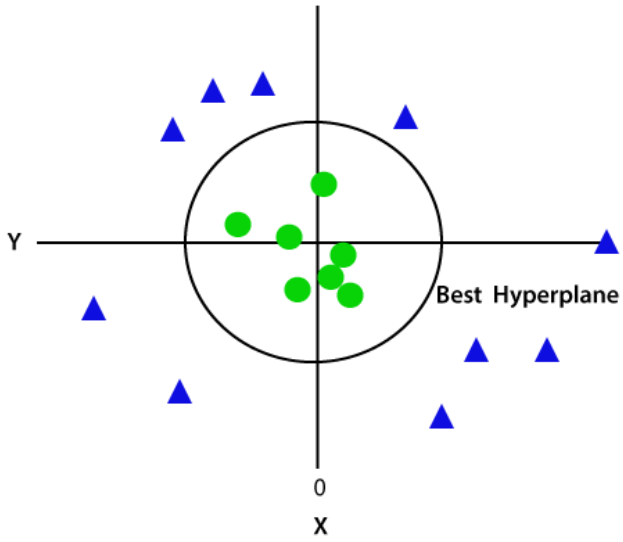
Σχήμα 2.18: *Μετατροπή μη διαχωρίσιμων δεδομένων σε διαχωρίσιμα [6].*

Τώρα λοιπόν, το SVM θα διαιρέσει τα σύνολα δεδομένων σε κλάσεις με τον ακόλουθο τρόπο. Σύμφωνα με την παρακάτω εικόνα:



Σχήμα 2.19: *Εφαρμογή SVM στα πλέον γραμμικά διαχωρίσιμα δεδομένα [6].*

Εφόσον βρισκόμαστε σε 3-d διάστημα, επομένως μοιάζει με ένα επίπεδο παράλληλο στον άξονα x. Αν το μετατρέψουμε σε 2d διάστημα με z=1, τότε θα γίνει ως:



Σχήμα 2.20: *Επιστροφή σε δισδιάστατο χώρο [6].*

Ως εκ τούτου, παίρνουμε μια περιφέρεια ακτίνας 1 σε περίπτωση μη γραμμικών δεδομένων.

2.3.6 Τι είναι ο αλγόριθμος Support Vector Machine με Radial Kernel ('svm' - SVM - Radial Kernel)

Οι μηχανές διανυσμάτων υποστήριξης είναι μια διάσημη και πολύ ισχυρή τεχνική ταξινόμησης που δεν χρησιμοποιεί κανενός είδους πιθανολογικό μοντέλο, όπως οποιοσδήποτε άλλος ταξινομητής, αλλά απλώς δημιουργεί υπερεπίπεδα ή απλά βάζοντας γραμμές, για να

διαχωρίσει και να ταξινομήσει τα δεδομένα σε κάποιο χώρο χαρακτηριστικών σε διαφορετικές περιοχές [26].

Οι διανυσματικοί ταξινομητές υποστήριξης χρησιμοποιούνται κυρίως για την επίλυση προβλημάτων δυαδικής ταξινόμησης όπου έχουμε μόνο 2 ετικέτες κλάσης που λένε Y= [1, 1] και μερικούς προγνωστικούς παράγοντες Xi. Και αυτό που κάνει το SVM είναι ότι δημιουργεί υπερεπίπεδα που με απλά λόγια είναι απλώς ευθείες γραμμές ή επίπεδα ή είναι μη γραμμικές καμπύλες, και αυτές οι γραμμές χρησιμοποιούνται για να διαχωρίσουν τα δεδομένα ή να χωρίσουν τα δεδομένα σε 2 ή περισσότερες κατηγορίες ανάλογα με τον τύπο της ταξινόμησης πρόβλημα [26].

Μια άλλη σημαντική έννοια στο SVM είναι οι ταξινομητές μέγιστου περιθωρίου. Αυτό σημαίνει ότι μεταξύ ενός συνόλου διαχωριστικών υπερεπιπέδων το SVM στοχεύει στην εύρεση αυτού που μεγιστοποιεί το περιθώριο M. Αυτό σημαίνει απλώς ότι θέλουμε να μεγιστοποιήσουμε το χάσμα ή την απόσταση μεταξύ των 2 κλάσεων από το όριο απόφασης (διαχωριστικό επίπεδο). Αυτή η έννοια του γραμμικού διαχωρισμού των δεδομένων σε 2 διαφορετικές κατηγορίες χρησιμοποιώντας ένα γραμμικό διαχωριστή ή μια ευθεία γραμμική γραμμή ονομάζεται γραμμική διαχωρισιμότητα [26].

Ο όρος διανύσματα υποστήριξης στο SVM είναι τα σημεία δεδομένων ή τα παραδείγματα εκπαίδευσης που χρησιμοποιούνται για τον καθορισμό ή τη μεγιστοποίηση του περιθωρίου. Τα διανύσματα υποστήριξης είναι τα σημεία που βρίσκονται κοντά στο όριο απόφασης ή στη λάθος πλευρά του ορίου [26].

Αναλυτικότερα, συχνά συναντάται ότι οι γραμμικοί διαχωριστές και τα μη γραμμικά όρια απόφασης αποτυγχάνουν λόγω των μη γραμμικών αλληλεπιδράσεων στα δεδομένα και της μη γραμμικής εξάρτησης μεταξύ των χαρακτηριστικών στο χώρο χαρακτηριστικών.

Το ¨κόλπο¨ εδώ είναι να κάνει κανείς την επέκταση χαρακτηριστικών. Έτσι, ο τρόπος που λύνουμε αυτό το πρόβλημα είναι κάνοντας έναν μη γραμμικό μετασχηματισμό στα χαρακτηριστικά Xi και τη μετατροπή τους σε χώρο υψηλότερων διαστάσεων (ας πούμε 2-D σε 3-D χώρο) που ονομάζεται χώρος χαρακτηριστικών. Τώρα με αυτόν τον μετασχηματισμό, είμαστε σε θέση να διαχωρίσουμε μη γραμμικά δεδομένα χρησιμοποιώντας ένα μη γραμμικό όριο απόφασης. Ένας διάσημος και πιο γενικός τρόπος για την προσθήκη μη γραμμικοτήτων σε ένα μοντέλο για την καταγραφή μη γραμμικών αλληλεπιδράσεων είναι απλώς η χρήση όρων υψηλότερου βαθμού, όπως τετραγωνικοί και κυβικοί πολυωνυμικοί όροι.

$$y_i = \beta_0 + \beta_1 X_1 + \beta_2 X_1^2 + \beta_3 X_2 + \beta_4 X_2^2 + \beta_5 X_2^3 \ldots = 0$$

Η παραπάνω εξίσωση είναι η εξίσωση του μη γραμμικού υπερεπίπεδου που δημιουργείται εάν χρησιμοποιήσουμε πολυωνυμικούς όρους υψηλότερου βαθμού για να προσαρμόσουμε τα δεδομένα για να πάρουμε ένα μη γραμμικό όριο απόφασης. Αυτό που στην πραγματικότητα κάνουμε είναι ότι διευρύνουμε τον χώρο των χαρακτηριστικών προσαρμόζοντας μη γραμμικές συναρτήσεις σε προγνωστικούς παράγοντες Xi . Αλλά το πρόβλημα με τα πολυώνυμα είναι ότι σε υψηλότερη διάσταση, δηλαδή όταν έχουν πολλούς προγνωστικούς παράγοντες, γίνονται άγρια και γενικά υπερπροσαρμόζονται σε υψηλότερους βαθμούς του πολυωνύμου.

Ως εκ τούτου, υπάρχει ένας άλλος κομψός τρόπος προσθήκης μη γραμμικοτήτων στο

SVM είναι με τη χρήση του ¨κόλπου¨ του πυρήνα [26].

Συνάρτηση Πυρήνα. Η συνάρτηση πυρήνα είναι συνάρτηση της μορφής [26] :

$$K(x, y) = (1 + \sum_{j=1}^{p} x_{ij} y_{ij})^d$$

όπου d είναι ο βαθμός του πολυωνύμου.

Τώρα ο τύπος της συνάρτησης Kernel που θα χρησιμοποιήσουμε εδώ είναι ένας Radial kernel. Έχει μορφή

$$K(x, y) = exp(-\gamma \sum_{j=1}^{p} (x_{ij} - y_{ij})^2)$$

και γ εδώ είναι μια παράμετρος συντονισμού που υπολογίζει την ομαλότητα του ορίου απόφασης και ελέγχει τη διακύμανση του μοντέλου. Αν γ είναι πολύ μεγάλο, τότε έχουμε ήσυχα κυμαινόμενα και ¨τρελά¨ όρια απόφασης, τα οποία ευθύνονται για την υψηλή διακύμανση και την υπερπροσαρμογή. Αν γ είναι μικρό, η γραμμή απόφασης ή το όριο είναι πιο ομαλό και έχει χαμηλή διακύμανση.

Έτσι τώρα η εξίσωση του ταξινομητή διανυσμάτων υποστήριξης γίνεται :

$$f(x) = \beta_0 + \sum_{i \in S} a_i K(x_i, y_i)$$

Εδώ Σ είναι τα διανύσματα υποστήριξης και α είναι απλώς μια τιμή βάρους που είναι μη μηδενική για όλα τα διανύσματα υποστήριξης, διαφορετικά 0.

2.3.7 Τι είναι ο αλγόριθμος Gaussian Process Classifier ('gpc' - Gaussian Process Classifier)

Ο ταξινομητής διαδικασιών Gauss είναι ένας αλγόριθμος μηχανικής μάθησης ταξινόμησης. Οι διεργασίες Gauss είναι μια γενίκευση της Gaussian κατανομής πιθανοτήτων και μπορούν να χρησιμοποιηθούν ως βάση για εξελιγμένους μη παραμετρικούς αλγόριθμους μηχανικής μάθησης για ταξινόμηση και παλινδρόμηση.

Είναι ένας τύπος μοντέλου πυρήνα, όπως τα SVM, και σε αντίθεση με τα SVM, είναι ικανός να προβλέψει εξαιρετικά βαθμονομημένες πιθανότητες ιδιότητας μέλους κλάσης, αν και η επιλογή και η διαμόρφωση του πυρήνα που χρησιμοποιείται στην καρδιά της μεθόδου είναι ¨προκλητική¨ [27].

Οι διεργασίες Gauss, ή GP για συντομία, είναι μια γενίκευση της κατανομής πιθανοτήτων Gauss (π.χ. η συνάρτηση σε σχήμα καμπάνας). Οι συναρτήσεις κατανομής πιθανότητας Gauss συνοψίζουν την κατανομή τυχαίων μεταβλητών, ενώ οι διαδικασίες Gauss συνοψίζουν τις ιδιότητες των συναρτήσεων, π.χ. τις παραμέτρους των συναρτήσεων. Ως εκ τούτου, μπορεί κανείς να σκεφτεί τις διαδικασίες Gauss ως ένα επίπεδο αφαίρεσης ή έμμεσης κατεύθυνσης πάνω από τις Gaussian συναρτήσεις [27].

Μια Gaussian διαδικασία είναι μια γενίκευση της Gaussian κατανομής πιθανοτήτων. Ενώ μια κατανομή πιθανότητας περιγράφει τυχαίες μεταβλητές που είναι βαθμωτές ή δια-

νύσματα (για πολυμεταβλητές κατανομές), μια στοχαστική διαδικασία διέπει τις ιδιότητες των συναρτήσεων.

Οι διεργασίες Gauss μπορούν να χρησιμοποιηθούν ως αλγόριθμος μηχανικής μάθησης για την προγνωστική μοντελοποίηση ταξινόμησης. Οι διεργασίες Gauss είναι ένας τύπος μεθόδου πυρήνα, όπως τα SVM, αν και είναι σε θέση να προβλέψουν πολύ βαθμονομημένες πιθανότητες, σε αντίθεση με τα SVM [27]. Οι διεργασίες Gauss απαιτούν τον καθορισμό ενός πυρήνα που ελέγχει τον τρόπο συσχέτισης των παραδειγμάτων μεταξύ τους. Συγκεκριμένα, ορίζει τη συνάρτηση συνδιακύμανσης των δεδομένων. Αυτό ονομάζεται λανθάνουσα συνάρτηση ή συνάρτηση «ενόχλησης». Η λανθάνουσα συνάρτηση f παίζει το ρόλο μιας ενοχλητικής συνάρτησης: δεν παρατηρούμε τις τιμές της ίδιας της f (παρατηρούμε μόνο τις εισόδους X και τις ετικέτες κλάσης y) και δεν μας ενδιαφέρουν ιδιαίτερα οι τιμές της f. Ο τρόπος με τον οποίο ομαδοποιούνται τα παραδείγματα χρησιμοποιώντας τον πυρήνα ελέγχει τον τρόπο με τον οποίο το μοντέλο ¨αντιλαμβάνεται¨ τα παραδείγματα, δεδομένου ότι υποθέτει ότι τα παραδείγματα που είναι ¨κοντά¨ το ένα στο άλλο έχουν την ίδια ετικέτα κλάσης. Ως εκ τούτου, είναι σημαντικό να δοκιμάσει κανείς διαφορετικές συναρτήσεις πυρήνα για το μοντέλο και διαφορετικές διαμορφώσεις για εξελιγμένες συναρτήσεις πυρήνα [27].

Απαιτεί επίσης μια συνάρτηση σύνδεσης που ερμηνεύει την εσωτερική αναπαράσταση και προβλέπει την πιθανότητα συμμετοχής στην κλάση. Η λογιστική συνάρτηση μπορεί να χρησιμοποιηθεί, επιτρέποντας τη μοντελοποίηση μιας διωνυμικής κατανομής πιθανότητας για δυαδική ταξινόμηση.

Για τη δυαδική διακριτική περίπτωση, μια απλή ιδέα είναι να μετατραπεί η έξοδος ενός μοντέλου παλινδρόμησης σε μια πιθανότητα κλάσης χρησιμοποιώντας μια συνάρτηση απόκρισης (το αντίστροφο μιας συνάρτησης σύνδεσης), η οποία ¨σκουπίζει¨ το όρισμά της, το οποίο μπορεί να βρίσκεται στον τομέα (-inf , inf), στην περιοχή [0, 1], που εγγυάται μια έγκυρη πιθανολογική ερμηνεία [27].

2.3.8 Τι είναι ο αλγόριθμος MLP Classifier ('mlp' - MLP Classifier)

Ένα multilayer perceptron (MLP), είναι μια πλήρως συνδεδεμένη κατηγορία τεχνητού νευρωνικού δικτύου (ANN). Ο όρος MLP χρησιμοποιείται διφορούμενα, μερικές φορές ¨χαλαρά¨ για να σημαίνει οποιοδήποτε feedforward ANN, μερικές φορές ¨αυστηρά¨ για να αναφέρεται σε δίκτυα που αποτελούνται από πολλαπλά στρώματα perceptrons (με ενεργοποίηση κατωφλίου). Τα πολυεπίπεδα perceptron μερικές φορές αναφέρονται στην καθομιλουμένη ως νευρωνικά δίκτυα ¨βανίλια¨, ειδικά όταν έχουν ένα μόνο κρυφό στρώμα [7].

Ένα MLP αποτελείται από τουλάχιστον τρία επίπεδα κόμβων: ένα στρώμα εισόδου, ένα κρυφό στρώμα και ένα στρώμα εξόδου. Εκτός από τους κόμβους εισόδου, κάθε κόμβος είναι ένας νευρώνας που χρησιμοποιεί μια μη γραμμική συνάρτηση ενεργοποίησης. Το MLP χρησιμοποιεί μια εποπτευόμενη τεχνική εκμάθησης που ονομάζεται backpropagation για εκπαίδευση. Τα πολλαπλά στρώματα και η μη γραμμική ενεργοποίησή του διακρίνουν το MLP από ένα γραμμικό perceptron. Μπορεί να διακρίνει δεδομένα που δεν διαχωρίζονται γραμμικά [7].

Συνάρτηση ενεργοποίησης

Εάν ένα πολυεπίπεδο perceptron έχει μια γραμμική συνάρτηση ενεργοποίησης σε όλους

τους νευρώνες, δηλαδή μια γραμμική συνάρτηση που χαρτογραφεί τις σταθμισμένες εισόδους στην έξοδο κάθε νευρώνα. Τότε η γραμμική άλγεβρα δείχνει ότι οποιοσδήποτε αριθμός στρωμάτων μπορεί να μειωθεί σε ένα διεπίπεδο μοντέλο εισόδου-εξόδου. Στα MLP ορισμένοι νευρώνες χρησιμοποιούν μια μη γραμμική συνάρτηση ενεργοποίησης που αναπτύχθηκε για να μοντελοποιήσει τη συχνότητα των δυναμικών δράσης ή πυροδότησης βιολογικών νευρώνων [7].

Οι δύο πιο καθιερωμένες συναρτήσεις ενεργοποίησης είναι και οι δύο σιγμοειδείς και περιγράφονται από :

$$y(v_i) = tanh(v_i)$$

και

$$y(v_i) = (1 + e^{-v_i})^{-1}$$

Στις πρόσφατες εξελίξεις της βαθιάς μάθησης η γραμμική μονάδα ανορθωτή (ReLU) χρησιμοποιείται συχνότερα ως ένας από τους πιθανούς τρόπους για να ξεπεραστούν τα αριθμητικά προβλήματα που σχετίζονται με τα σιγμοειδή [7].

Η πρώτη είναι μια υπερβολική εφαπτομένη που κυμαίνεται από -1 έως 1, ενώ η άλλη είναι η λογιστική συνάρτηση, η οποία είναι παρόμοια σε σχήμα αλλά κυμαίνεται από 0 έως 1. Εδώ yi είναι η έξοδος του i-οστού κόμβου (νευρώνας) και vi είναι το σταθμισμένο άθροισμα των συνδέσεων εισόδου. Έχουν προταθεί εναλλακτικές λειτουργίες ενεργοποίησης, συμπεριλαμβανομένων των λειτουργιών ανορθωτή και softplus. Πιο εξειδικευμένες λειτουργίες ενεργοποίησης περιλαμβάνουν συναρτήσεις ακτινικής βάσης (χρησιμοποιούνται σε δίκτυα ακτινικής βάσης, μια άλλη κατηγορία μοντέλων εποπτευόμενων νευρωνικών δικτύων).

Επίπεδα

Το MLP αποτελείται από τρία ή περισσότερα επίπεδα (ένα στρώμα εισόδου και ένα στρώμα εξόδου με ένα ή περισσότερα κρυφά επίπεδα) κόμβων που δεν ενεργοποιούνται γραμμικά. Εφόσον τα MLP είναι πλήρως συνδεδεμένα, κάθε κόμβος σε ένα επίπεδο συνδέεται με ένα συγκεκριμένο βάρος wij σε κάθε κόμβο στο επόμενο επίπεδο [7].

Εκμάθηση

Η εκμάθηση λαμβάνει χώρα στο perceptron αλλάζοντας τα βάρη σύνδεσης μετά την επεξεργασία κάθε τμήματος δεδομένων, με βάση το μέγεθος του σφάλματος στην έξοδο σε σύγκριση με το αναμενόμενο αποτέλεσμα. Αυτό είναι ένα παράδειγμα εποπτευόμενης μάθησης και πραγματοποιείται μέσω της οπισθοδιάδοσης, μιας γενίκευσης του αλγορίθμου των ελαχίστων μέσων τετραγώνων στο γραμμικό perceptron [7].

Σύμφωνα με την ίδια πηγή, μπορούμε να αναπαραστήσουμε τον βαθμό σφάλματος σε έναν κόμβο εξόδου j στο n-οστό σημείο δεδομένων (παράδειγμα εκπαίδευσης) με

$$e_j(n) = d_j(n) - y_j(n),$$

όπου d είναι η τιμή στόχος και y είναι η τιμή που παράγεται από το perceptron. Τα βάρη των κόμβων μπορούν στη συνέχεια να προσαρμοστούν με βάση διορθώσεις που ελαχιστοποιούν το σφάλμα σε ολόκληρη την έξοδο, που δίνονται από

$$\varepsilon(n) = \frac{1}{2}\sum_j e_j^2(n),$$

Χρησιμοποιώντας κλίση κατάβασης, η αλλαγή σε κάθε βάρος είναι

$$\Delta w_{ji}(n) = -\eta\frac{\vartheta\varepsilon(n)}{\vartheta v_j(n)}y_i(n),$$

όπου yi είναι η έξοδος του προηγούμενου νευρώνα και η είναι ο ρυθμός εκμάθησης, ο οποίος επιλέγεται για να διασφαλίσει ότι τα βάρη συγκλίνουν γρήγορα σε μια απόκριση, χωρίς ταλαντώσεις .

Η παράγωγος που θα υπολογιστεί εξαρτάται από το επαγόμενο τοπικό πεδίο vi, το οποίο από μόνο του ποικίλλει. Είναι εύκολο να αποδειχθεί ότι για έναν κόμβο εξόδου αυτή η παράγωγος μπορεί να απλοποιηθεί σε

$$-\frac{\vartheta\varepsilon(n)}{\vartheta v_j(n)} = e_j(n)\phi'(v_j(n))$$

όπου φ′ είναι το παράγωγο της συνάρτησης ενεργοποίησης που περιγράφεται παραπάνω, η οποία από μόνη της δεν ποικίλλει. Η ανάλυση είναι πιο δύσκολη για την αλλαγή των βαρών σε έναν κρυφό κόμβο, αλλά μπορεί να φανεί ότι η σχετική παράγωγος είναι

$$-\frac{\vartheta\varepsilon(n)}{\vartheta v_j(n)} = \phi'(v_j(n))\sum_k -\frac{\vartheta\varepsilon(n)}{\vartheta v_k(n)}w_{kj}(n)$$

Αυτό εξαρτάται από την αλλαγή στα βάρη των κ-οστών κόμβων, που αντιπροσωπεύουν το επίπεδο εξόδου. Έτσι, για να αλλάξουμε τα βάρη των κρυφών επιπέδων, τα βάρη του στρώματος εξόδου αλλάζουν ανάλογα με την παράγωγο της συνάρτησης ενεργοποίησης, και έτσι αυτός ο αλγόριθμος αντιπροσωπεύει μια αντίστροφη διάδοση της συνάρτησης ενεργοποίησης [7].

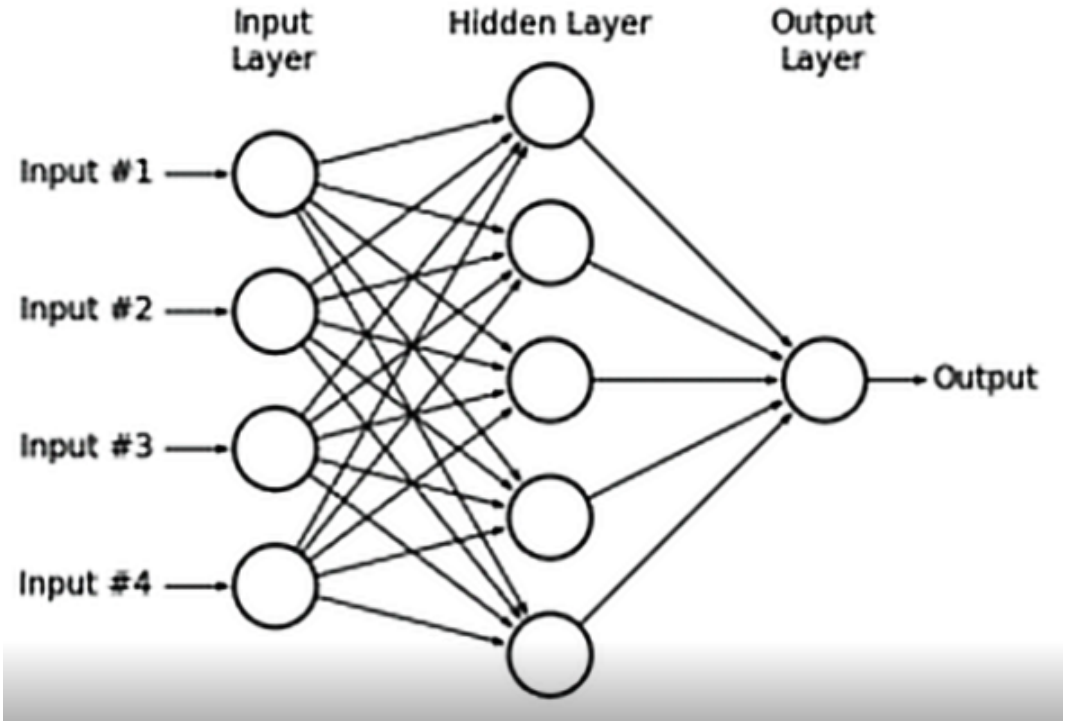
Εφαρμογές

- Τα MLP είναι χρήσιμα στην έρευνα για την ικανότητά τους να επιλύουν προβλήματα στοχαστικά, κάτι που συχνά επιτρέπει κατά προσέγγιση λύσεις για εξαιρετικά πολύπλοκα προβλήματα όπως η προσέγγιση της φυσικής κατάστασης.

- Τα MLP είναι καθολικές προσεγγίσεις συναρτήσεων όπως φαίνεται από το θεώρημα του Cybenko, έτσι ώστε να μπορούν να χρησιμοποιηθούν για τη δημιουργία μαθηματικών μοντέλων με ανάλυση παλινδρόμησης. Καθώς η ταξινόμηση είναι μια ιδιαίτερη περίπτωση παλινδρόμησης όταν η μεταβλητή απόκρισης είναι κατηγορική. Τα MLP δημιουργούν καλούς αλγόριθμους ταξινομητή.

- Τα MLP ήταν μια δημοφιλής λύση μηχανικής μάθησης στη δεκαετία του 1980, βρίσκοντας εφαρμογές σε διάφορα πεδία όπως η αναγνώριση ομιλίας, η αναγνώριση εικόνας και το λογισμικό μηχανικής μετάφρασης, αλλά στη συνέχεια αντιμετώπισαν ισχυρό ανταγωνισμό από πολύ απλούστερες διανυσματικές μηχανές υποστήριξης . Το ενδιαφέρον για τα δίκτυα backpropagation επέστρεψε λόγω των επιτυχιών της βαθιάς

μάθησης.



Σχήμα 2.21: *Αρχιτεκτονική ενός MLP [7].*

2.3.9 Τι είναι ο αλγόριθμος Ridge Classifier ('ridge' - Ridge Classifier)

Η παλινδρόμηση είναι μια εργασία μοντελοποίησης που περιλαμβάνει την πρόβλεψη μιας αριθμητικής τιμής δεδομένης μιας εισόδου.

Η γραμμική παλινδρόμηση είναι ο τυπικός αλγόριθμος για την παλινδρόμηση που προϋποθέτει μια γραμμική σχέση μεταξύ των εισόδων και της μεταβλητής στόχου. Μια επέκταση στη γραμμική παλινδρόμηση επικαλείται την προσθήκη κυρώσεων στη συνάρτηση απώλειας κατά τη διάρκεια της εκπαίδευσης που ενθαρρύνει απλούστερα μοντέλα που έχουν μικρότερες τιμές συντελεστών. Αυτές οι επεκτάσεις αναφέρονται ως κανονικοποιημένη (regularized) γραμμική παλινδρόμηση ή τιμωρημένη γραμμική παλινδρόμηση [28].

Η παλινδρόμηση κορυφογραμμής (Ridge Regression) είναι ένας δημοφιλής τύπος κανονικοποιημένη γραμμικής παλινδρόμησης που περιλαμβάνει ποινή L2. Αυτό έχει ως αποτέλεσμα τη συρρίκνωση των συντελεστών για εκείνες τις μεταβλητές εισόδου που δεν συμβάλλουν πολύ στην εργασία πρόβλεψης [28].

Παλινδρόμηση κορυφογραμμής - (Ridge Regression)

Η γραμμική παλινδρόμηση αναφέρεται σε ένα μοντέλο που υποθέτει μια γραμμική σχέση μεταξύ των μεταβλητών εισόδου και της μεταβλητής στόχου.

Με μια μεμονωμένη μεταβλητή εισόδου, αυτή η σχέση είναι μια γραμμή και με υψηλότερες διαστάσεις. Αυτή η σχέση μπορεί να θεωρηθεί ως ένα υπερεπίπεδο που συνδέει

τις μεταβλητές εισόδου με τη μεταβλητή στόχο. Οι συντελεστές του μοντέλου βρίσκονται μέσω μιας διαδικασίας βελτιστοποίησης που επιδιώκει να ελαχιστοποιήσει το αθροιστικό τετραγωνικό σφάλμα μεταξύ των προβλέψεων (yhat) και των αναμενόμενων τιμών στόχου (y) [28].

$$loss = \sum_{i=0}^{n} (y_i - yhat_i)^2$$

Ένα πρόβλημα με τη γραμμική παλινδρόμηση είναι ότι οι εκτιμώμενοι συντελεστές του μοντέλου μπορεί να γίνουν μεγάλοι, καθιστώντας το μοντέλο ευαίσθητο στις εισροές και πιθανώς ασταθές. Αυτό ισχύει ιδιαίτερα για προβλήματα με λίγες παρατηρήσεις (δείγματα) ή λιγότερα δείγματα (n) από ό,τι οι προβλέψεις εισόδου (p) ή οι μεταβλητές.

Μια προσέγγιση για την αντιμετώπιση της σταθερότητας των μοντέλων παλινδρόμησης είναι η αλλαγή της συνάρτησης απώλειας ώστε να συμπεριλάβει πρόσθετο κόστος για ένα μοντέλο που έχει μεγάλους συντελεστές. Τα μοντέλα γραμμικής παλινδρόμησης που χρησιμοποιούν αυτές τις τροποποιημένες συναρτήσεις απώλειας κατά τη διάρκεια της εκπαίδευσης αναφέρονται συλλογικά ως τιμωρημένη (penalized) γραμμική παλινδρόμηση [28].

Μια δημοφιλής ποινή είναι η τιμωρία ενός μοντέλου που βασίζεται στο άθροισμα των τετραγωνικών τιμών των συντελεστών (beta). Αυτό ονομάζεται ποινή L2.

$$L2_{penalty} = \sum_{j=0}^{p} \beta_j^2$$

Μια ποινή L2 ελαχιστοποιεί το μέγεθος όλων των συντελεστών, αν και αποτρέπει την αφαίρεση οποιωνδήποτε συντελεστών από το μοντέλο, επιτρέποντας την τιμή τους να μηδενιστεί.

Το αποτέλεσμα αυτής της ποινής είναι ότι οι εκτιμήσεις παραμέτρων επιτρέπεται να γίνουν μεγάλες μόνο εάν υπάρχει αναλογική μείωση της SSE. Στην πραγματικότητα, αυτή η μέθοδος συρρικνώνει τις εκτιμήσεις προς το 0 καθώς η ποινή λάμδα γίνεται μεγάλη (αυτές οι τεχνικές μερικές φορές ονομάζονται ¨μέθοδοι συρρίκνωσης¨) [28].

Αυτή η ποινή μπορεί να προστεθεί στη συνάρτηση κόστους για γραμμική παλινδρόμηση και αναφέρεται ως κανονικοποίηση Tikhonov (από τον συγγραφέα) ή Ridge Regression γενικότερα.

Χρησιμοποιείται μια υπερπαράμετρος που ονομάζεται ¨λάμδα¨ που ελέγχει τη στάθμιση της ποινής στη συνάρτηση απώλειας. Μια προεπιλεγμένη τιμή 1,0 θα επιβαρύνει πλήρως την ποινή. Η τιμή 0 αποκλείει την ποινή. Πολύ μικρές τιμές λάμδα, όπως 1e-3 ή μικρότερες είναι συνηθισμένες [28].

$$ridge_{loss} = loss + (lambda * l2_{penalty})$$

2.3.10 Τι είναι ο αλγόριθμος Random Forest Classifier ('rf' - Random Forest Classifier)

Τα τυχαία δάση είναι ένας εποπτευόμενος αλγόριθμος εκμάθησης. Μπορεί να χρησιμοποιηθεί τόσο για ταξινόμηση όσο και για παλινδρόμηση. Είναι επίσης ο πιο ευέλικτος και εύχρηστος αλγόριθμος. Ένα δάσος αποτελείται από δέντρα. Λέγεται ότι όσο περισσότερα δέντρα έχει, τόσο πιο εύρωστο είναι ένα δάσος. Τα Random Forests δημιουργούν δέντρα αποφάσεων σε τυχαία επιλεγμένα δείγματα δεδομένων, λαμβάνουν πρόβλεψη από κάθε δέντρο και επιλέγουν την καλύτερη λύση μέσω ψηφοφορίας. Παρέχουν, επίσης, μια αρκετά καλή ένδειξη της σημασίας των χαρακτηριστικών. Τα Random Forests έχουν μια ποικιλία εφαρμογών, όπως μηχανές συστάσεων, ταξινόμηση εικόνων και επιλογή χαρακτηριστικών. Μπορεί να χρησιμοποιηθούν για την ταξινόμηση πιστών αιτούντων δανείου, τον εντοπισμό δόλιας δραστηριότητας και την πρόβλεψη ασθενειών. Βρίσκονται στη βάση του αλγορίθμου Boruta, ο οποίος επιλέγει σημαντικά χαρακτηριστικά σε ένα σύνολο δεδομένων [8].

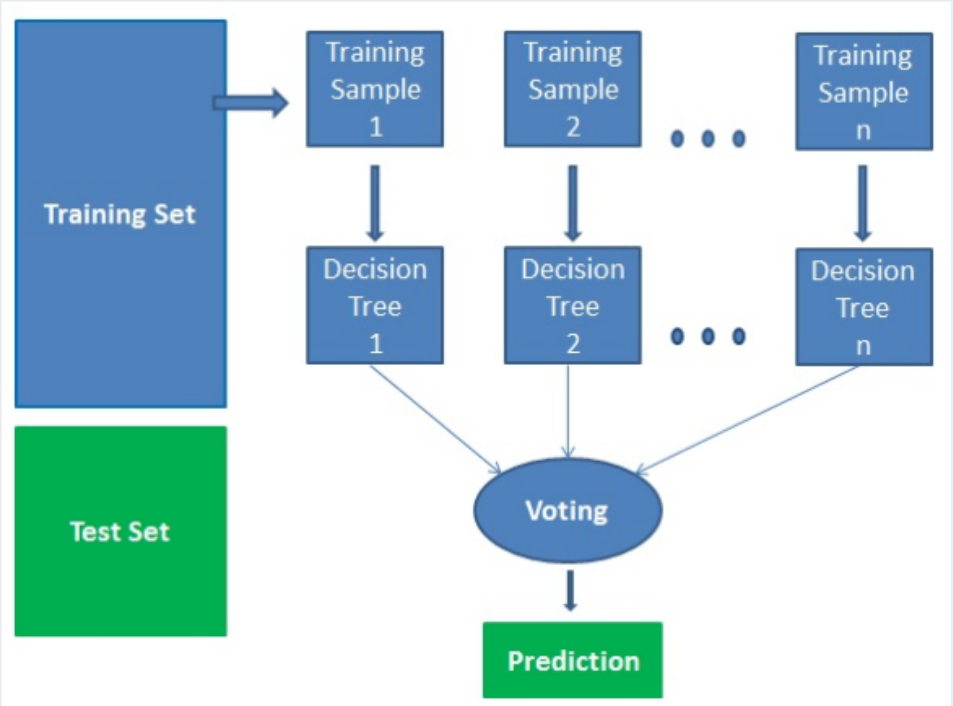
Ο αλγόριθμος των τυχαίων δασών

Ας υποθέσουμε ότι θέλει κάποιος να πάει ένα ταξίδι και θα ήθελε να ταξιδέψει σε ένα μέρος που θα απολαύσει. Τι κάνει λοιπόν για να βρεί ένα μέρος που θα του/της αρέσει. Μπορεί να κάνει αναζήτηση στο διαδίκτυο, να διαβάσει κριτικές σε ταξιδιωτικά ιστολόγια και πύλες ή μπορεί επίσης να ρωτήσει τους φίλους του/της. Ας υποθέσουμε ότι αποφασίζει να ρωτήσει τους φίλους και μίλησε μαζί τους για την προηγούμενη ταξιδιωτική εμπειρία τους σε διάφορα μέρη. Θα λάβει μερικές συστάσεις από κάθε φίλο. Τώρα πρέπει να κάνει μια λίστα με αυτά τα προτεινόμενα μέρη. Στη συνέχεια, τους ζητάει να ψηφίσουν (ή να επιλέξουν ένα καλύτερο μέρος για το ταξίδι) από τη λίστα με τα προτεινόμενα μέρη που πραγματοποίησε [8]. Το μέρος με τον μεγαλύτερο αριθμό ψήφων θα είναι η τελική του/της επιλογή για το ταξίδι. Στην παραπάνω διαδικασία απόφασης, υπάρχουν δύο μέρη. Πρώτα, ρωτάει τους φίλους του/της για την ατομική τους ταξιδιωτική εμπειρία και λαμβάνει μία σύσταση από πολλά μέρη που έχουν επισκεφτεί. Αυτό το μέρος είναι σαν να χρησιμοποιεί τον αλγόριθμο του δέντρου αποφάσεων. Εδώ, κάθε φίλος κάνει μια επιλογή από τα μέρη που έχει επισκεφτεί μέχρι τώρα. Το δεύτερο μέρος, αφού συγκεντρωθούν όλες οι συστάσεις, είναι η διαδικασία ψηφοφορίας για την επιλογή της καλύτερης θέσης στη λίστα συστάσεων. Όλη αυτή η διαδικασία λήψης συστάσεων από φίλους και ψηφοφορίας για την εύρεση του καλύτερου μέρους είναι γνωστή ως αλγόριθμος τυχαίων δασών. Τεχνικά είναι μια μέθοδος συνόλου (βασισμένη στην προσέγγιση διαίρει και βασίλευε) δέντρων αποφάσεων που δημιουργούνται σε ένα τυχαία διαχωρισμένο σύνολο δεδομένων. Αυτή η συλλογή ταξινομητών δέντρων απόφασης είναι επίσης γνωστή ως το δάσος. Τα μεμονωμένα δέντρα απόφασης δημιουργούνται χρησιμοποιώντας έναν δείκτη επιλογής χαρακτηριστικών όπως κέρδος πληροφοριών, αναλογία κέρδους και δείκτη Gini για κάθε χαρακτηριστικό. Κάθε δέντρο εξαρτάται από ένα ανεξάρτητο τυχαίο δείγμα. Σε ένα πρόβλημα ταξινόμησης, κάθε δέντρο ψηφίζει και η πιο δημοφιλής κατηγορία επιλέγεται ως τελικό αποτέλεσμα. Στην περίπτωση παλινδρόμησης, ο μέσος όρος όλων των εξόδων του δέντρου θεωρείται ως τελικό αποτέλεσμα. Είναι απλούστερο και πιο ισχυρό σε σύγκριση με τους άλλους αλγόριθμους μη γραμμικής ταξινόμησης [8].

Πώς λειτουργεί ο αλγόριθμος

Λειτουργεί σε τέσσερα βήματα:

- Επιλέξτε τυχαία δείγματα από ένα δεδομένο σύνολο δεδομένων.
- Κατασκευάστε ένα δέντρο απόφασης για κάθε δείγμα και λάβετε ένα αποτέλεσμα πρόβλεψης από κάθε δέντρο απόφασης.
- Εκτελέστε ψηφοφορία για κάθε προβλεπόμενο αποτέλεσμα.
- Επιλέξτε το αποτέλεσμα της πρόβλεψης με τις περισσότερες ψήφους ως τελική πρόβλεψη.



Σχήμα 2.22: *Αλγόριθμος τυχαίου δάσους [8].*

Πλεονεκτήματα:

- Τα τυχαία δάση θεωρούνται ως μια εξαιρετικά ακριβής και ισχυρή μέθοδος λόγω του αριθμού των δέντρων απόφασης που συμμετέχουν στη διαδικασία.
- Δεν πάσχουν από το πρόβλημα της υπερπροσαρμογής. Ο κύριος λόγος είναι ότι παίρνουν το μέσο όρο όλων των προβλέψεων, γεγονός που ακυρώνει τις προκαταλήψεις.
- Ο αλγόριθμος μπορεί να χρησιμοποιηθεί τόσο σε προβλήματα ταξινόμησης όσο και σε προβλήματα παλινδρόμησης.
- Τα τυχαία δάση μπορούν επίσης να χειριστούν τιμές που λείπουν. Υπάρχουν δύο τρόποι για να τα χειριστεί κανείς: η χρήση διαμέσου τιμών για την αντικατάσταση συνεχών μεταβλητών και ο υπολογισμός του σταθμισμένου μέσου όρου των τιμών που λείπουν.

- Μπορεί κανείς να λάβει τη σχετική σημασία χαρακτηριστικών, η οποία βοηθά στην επιλογή των χαρακτηριστικών που συμβάλλουν περισσότερο στον ταξινομητή.

Μειονεκτήματα:

- Τα τυχαία δάση είναι αργά στη δημιουργία προβλέψεων επειδή έχουν πολλαπλά δέντρα απόφασης. Κάθε φορά που κάνουν μια πρόβλεψη, όλα τα δέντρα στο δάσος πρέπει να κάνουν μια πρόβλεψη για την ίδια δεδομένη είσοδο και στη συνέχεια να την ψηφίσουν. Όλη αυτή η διαδικασία είναι χρονοβόρα.

- Το μοντέλο είναι δύσκολο να ερμηνευτεί σε σύγκριση με ένα δέντρο αποφάσεων, όπου μπορεί κανείς εύκολα να πάρει μια απόφαση ακολουθώντας τη διαδρομή στο δέντρο.

Εύρεση σημαντικών χαρακτηριστικών

Τα τυχαία δάση προσφέρουν επίσης έναν καλό δείκτη επιλογής χαρακτηριστικών. Το Scikit-learn παρέχει μια επιπλέον μεταβλητή με το μοντέλο, η οποία δείχνει τη σχετική σημασία ή συμβολή κάθε χαρακτηριστικού στην πρόβλεψη. Υπολογίζει αυτόματα τη βαθμολογία συνάφειας κάθε χαρακτηριστικού στη φάση εκπαίδευσης. Στη συνέχεια, μειώνει τη συνάφεια έτσι ώστε το άθροισμα όλων των βαθμολογιών να είναι 1 [8].

Αυτή η βαθμολογία βοηθάει να επιλέξει κανείς τα πιο σημαντικά χαρακτηριστικά και να απορρίψει τα λιγότερο σημαντικά για την κατασκευή μοντέλων.

Το τυχαίο δάσος χρησιμοποιεί τη σημασία τζίνι ή τη μέση μείωση της ακαθαρσίας (MDI) για να υπολογίσει τη σημασία κάθε χαρακτηριστικού. Η σημασία του Gini είναι επίσης γνωστή ως η συνολική μείωση της ακαθαρσίας του κόμβου. Αυτό είναι το πόσο μειώνεται η προσαρμογή ή η ακρίβεια του μοντέλου όταν ρίχνει μια μεταβλητή. Όσο μεγαλύτερη είναι η μείωση, τόσο πιο σημαντική είναι η μεταβλητή. Εδώ, η μέση μείωση είναι μια σημαντική παράμετρος για την επιλογή μεταβλητών. Ο δείκτης Gini μπορεί να περιγράψει τη συνολική επεξηγηματική ισχύ των μεταβλητών [8].

Random Forests εναντίον Decision Trees

- Τα τυχαία δάση είναι ένα σύνολο πολλών δέντρων αποφάσεων.

- Τα βαθιά δέντρα απόφασης μπορεί να υποφέρουν από υπερπροσαρμογή, αλλά τα τυχαία δάση αποτρέπουν την υπερπροσαρμογή δημιουργώντας δέντρα σε τυχαία υποσύνολα.

- Τα δέντρα αποφάσεων είναι υπολογιστικά πιο γρήγορα.

- Τα τυχαία δάση είναι δύσκολο να ερμηνευτούν, ενώ ένα δέντρο αποφάσεων είναι εύκολα ερμηνεύσιμο και μπορεί να μετατραπεί σε κανόνες.

2.3.11 Τι είναι ο αλγόριθμος Quadratic Discriminant Analysis ('qda' - Quadratic Discriminant Analysis)

Ανάλυση τετραγωνικής διάκρισης - Quadratic Discriminant Analysis

Το QDA είναι μια παραλλαγή του LDA Linear Discriminant Analysis (θα αναλυθεί στη συνέχεια) στην οποία υπολογίζεται ένας ατομικός πίνακας συνδιακύμανσης για κάθε κατηγορία παρατηρήσεων. Το QDA είναι ιδιαίτερα χρήσιμο εάν υπάρχει προηγούμενη γνώση ότι μεμονωμένες τάξεις εμφανίζουν διακριτές συνδιακυμάνσεις. Ένα μειονέκτημα του QDA είναι ότι δεν μπορεί να χρησιμοποιηθεί ως τεχνική μείωσης διαστάσεων [29].

Αναλυτικότερα στο QDA, πρέπει να υπολογίσουμε το Σ_k για κάθε κλάση $k \in 1, \ldots, K$ αντί να υποθέσουμε Σκ=Σ όπως στο LDA. Η διακριτική συνάρτηση του LDA είναι τετραγωνική στο x:

$$\delta_k(x) = -\frac{1}{2} log|\Sigma_k| - \frac{1}{2}(x - \mu_k)^T \Sigma_k^{-1}(x - \mu_k) + log\pi_k$$

Εφόσον το QDA εκτιμά έναν πίνακα συνδιακύμανσης για κάθε τάξη, έχει μεγαλύτερο αριθμό αποτελεσματικών παραμέτρων από το LDA. Μπορούμε να εξαγάγουμε τον αριθμό των παραμέτρων με τον ακόλουθο τρόπο [29]:

- Χρειαζόμαστε K class priors π_k. Εφόσον $\sum_{i=1}^{K} \pi_k = 1$ δεν χρειαζόμαστε παράμετρο για ένα από τα προηγούμενα. Έτσι, υπάρχουν ελεύθερες παράμετροι K-1 για τα priors.
- Εφόσον υπάρχουν K κεντροειδή, μ_k, με p καταχωρήσεις το καθένα, υπάρχουν παράμετροι K_p που σχετίζονται με τους μέσους όρους.
- Από τον πίνακα συνδιακύμανσης, Σ_k, χρειάζεται μόνο να εξετάσουμε τη διαγώνιο και το άνω ορθογώνιο τρίγωνο. Αυτή η περιοχή του πίνακα συνδιακύμανσης έχει στοιχεία p(p+1)2. Εφόσον K τέτοιοι πίνακες πρέπει να εκτιμηθούν, υπάρχουν παράμετροι K_p(p+1)2 που σχετίζονται με τους πίνακες συνδιακύμανσης.

Έτσι, ο αποτελεσματικός αριθμός παραμέτρων QDA είναι :

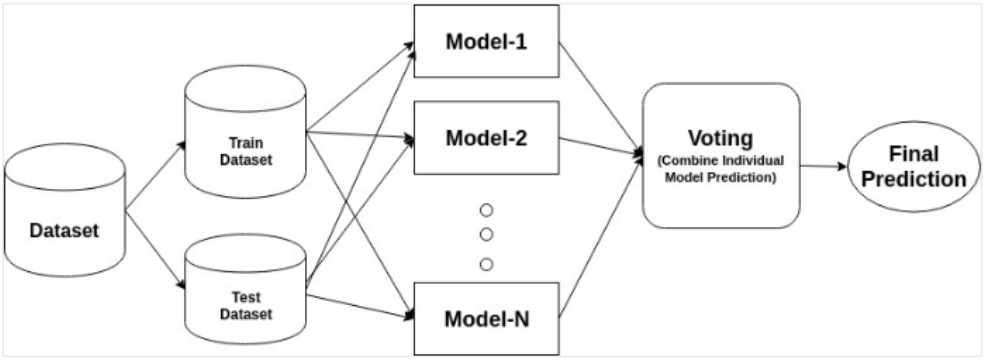
$$K - 1 + K_p + K\frac{p(p+1)}{2}$$

Δεδομένου ότι ο αριθμός των παραμέτρων QDA είναι τετραγωνικός σε p, το QDA θα πρέπει να χρησιμοποιείται με προσοχή όταν ο χώρος χαρακτηριστικών είναι μεγάλος.

2.3.12 Τι είναι ο αλγόριθμος Ada Boost Classifier ('ada' - Ada Boost Classifier)

Προσέγγιση συνόλου Μηχανικής Μάθησης (ensemble approach)

Ένα σύνολο είναι ένα σύνθετο μοντέλο, που συνδυάζει μια σειρά από ταξινομητές χαμηλής απόδοσης με στόχο τη δημιουργία ενός βελτιωμένου ταξινομητή. Τα σύνολα προσφέρουν μεγαλύτερη ακρίβεια από τον ατομικό ή τον βασικό ταξινομητή. Οι μέθοδοι συνόλου μπορούν να παραλληλιστούν με την κατανομή κάθε βασικού μαθητευόμενου σε διαφορετικές μηχανές. Τέλος, μπορεί κανείς να πει ότι οι μέθοδοι εκμάθησης συνόλου είναι μετα-αλγόριθμοι που συνδυάζουν διάφορες μεθόδους μηχανικής μάθησης σε ένα ενιαίο μοντέλο πρόβλεψης για να αυξήσουν την απόδοση. Οι μέθοδοι συνόλου μπορούν να μειώσουν τη διακύμανση χρησιμοποιώντας την προσέγγιση σακουλών (bagging), την προκατάληψη χρησιμοποιώντας μια προσέγγιση ενίσχυσης (boosting) ή να βελτιώσουν τις προβλέψεις χρησιμοποιώντας την προσέγγιση στοίβαξης (stacking) [9].

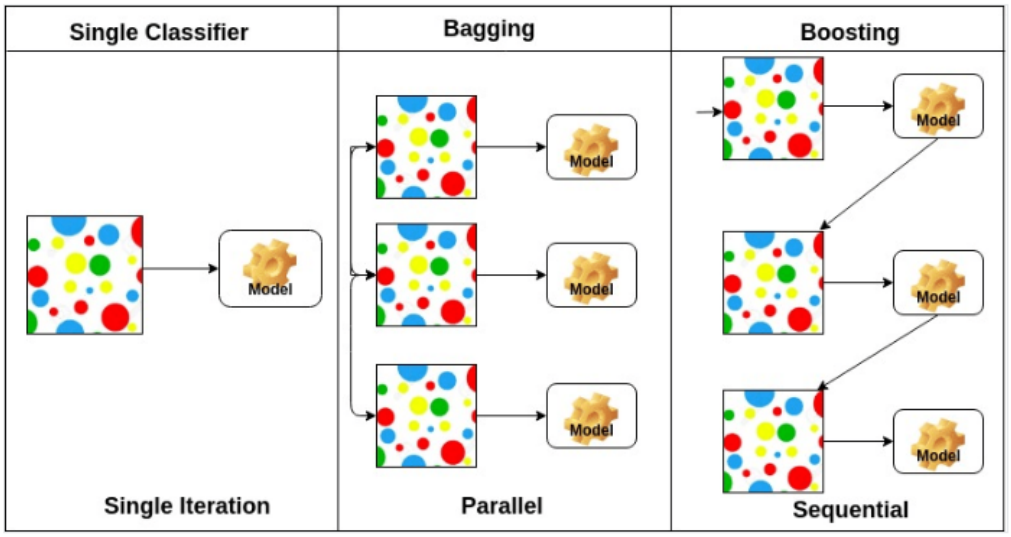


Σχήμα 2.23: *Μέθοδοι Ensemble [9].*

Bagging σημαίνει πρακτικά bootstrap agregation. Συνδυάζει πολλούς μαθητευόμενους με τρόπο που να μειώνει τη διακύμανση των εκτιμήσεων. Για παράδειγμα, το τυχαίο δάσος εκπαιδεύει το M Decision Tree. Μπορεί κανείς να εκπαιδεύσει M διαφορετικά δέντρα σε διαφορετικά τυχαία υποσύνολα δεδομένων και να εκτελέσει ψηφοφορία για την τελική πρόβλεψη. Οι μέθοδοι των συνόλων συσκευασίας είναι Random Forest και Extra Trees [9]. Οι αλγόριθμοι ενίσχυσης είναι ένα σύνολο ταξινομητή χαμηλής ακρίβειας για τη δημιουργία ενός ταξινομητή υψηλής ακρίβειας. Ο ταξινομητής χαμηλής ακρίβειας (ή αδύναμος ταξινομητής) προσφέρει ακρίβεια καλύτερη από τη ρίψη ενός νομίσματος. Ο υψηλής ακρίβειας ταξινομητής (ή ισχυρός ταξινομητής) προσφέρει ποσοστό σφάλματος κοντά στο 0. Ο αλγόριθμος ενίσχυσης μπορεί να παρακολουθεί το μοντέλο που απέτυχε στην ακριβή πρόβλεψη. Οι αλγόριθμοι ενίσχυσης επηρεάζονται λιγότερο από το πρόβλημα της υπερπροσαρμογής. Οι ακόλουθοι τρεις αλγόριθμοι έχουν κερδίσει τεράστια δημοτικότητα σε διαγωνισμούς επιστήμης δεδομένων [9].

- AdaBoost
- Gradient Tree Boosting
- XGBoost

Η στοίβαξη (ή η στοιβαγμένη γενίκευση) είναι μια τεχνική εκμάθησης συνόλου που συνδυάζει πολλαπλές προβλέψεις μοντέλων ταξινόμησης βάσης σε ένα νέο σύνολο δεδομένων. Αυτά τα νέα δεδομένα αντιμετωπίζονται ως δεδομένα εισόδου για έναν άλλο ταξινομητή. Αυτός ο ταξινομητής χρησιμοποιήθηκε για την επίλυση αυτού του προβλήματος. Η στοίβαξη αναφέρεται συχνά ως ανάμειξη (blending) [9].



Σχήμα 2.24: *Bagging vs Boosting [9].*

Με βάση τη διάταξη των βασικών μαθητών, οι μέθοδοι συνόλου μπορούν να χωριστούν σε δύο ομάδες: Για παράδειγμα, στις μεθόδους παράλληλων συνόλων, οι βασικοί εκπαιδευόμενοι δημιουργούνται παράλληλα, όπως στο Τυχαίο Δάσος. Στις μεθόδους διαδοχικών συνόλων, οι βασικοί μαθητές δημιουργούνται διαδοχικά, για παράδειγμα ο AdaBoost.

Με βάση τον τύπο των βασικών μαθητών, οι μέθοδοι συνόλου μπορούν να χωριστούν σε δύο ομάδες: η μέθοδος ομογενούς συνόλου χρησιμοποιεί τον ίδιο τύπο βασικού μαθητή σε κάθε επανάληψη. Η μέθοδος ετερογενούς συνόλου χρησιμοποιεί τον διαφορετικό τύπο βασικού μαθητή σε κάθε επανάληψη [9].

Ταξινομητής AdaBoost

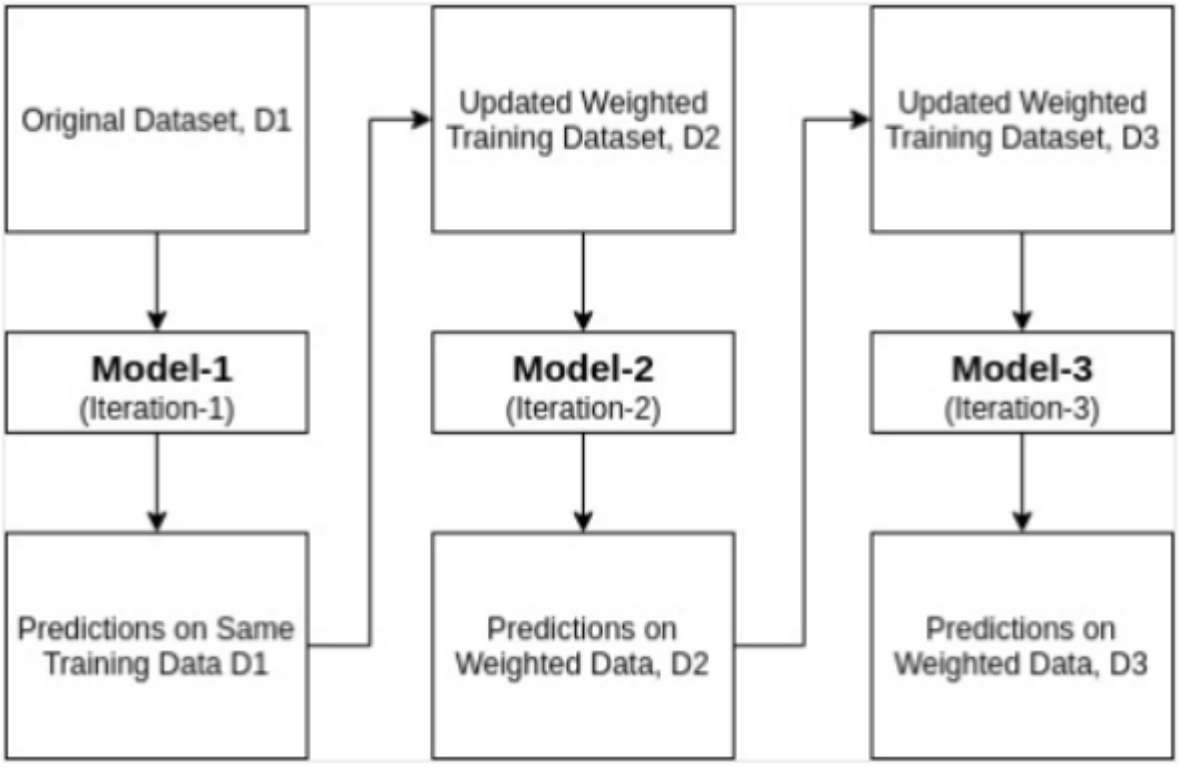
Ο Ada-boost ή Adaptive Boosting είναι ένας από τους ταξινομητές ενίσχυσης συνόλου που προτάθηκε από τους Yoav Freund και Robert Schapire το 1996. Συνδυάζει πολλαπλούς ταξινομητές για να αυξήσει την ακρίβεια των ταξινομητών. Ο AdaBoost είναι μια επαναληπτική μέθοδος συνόλου. Ο ταξινομητής AdaBoost δημιουργεί έναν ισχυρό ταξινομητή συνδυάζοντας πολλούς ταξινομητές με κακή απόδοση, έτσι ώστε να έχει κανείς έναν ισχυρό ταξινομητή υψηλής ακρίβειας. Η βασική ιδέα πίσω από τον Adaboost είναι να ορίσει κανείς τα βάρη των ταξινομητών και να εκπαιδεύσει το δείγμα δεδομένων σε κάθε επανάληψη έτσι ώστε να διασφαλίζει τις ακριβείς προβλέψεις ασυνήθιστων παρατηρήσεων. Οποιοσδήποτε αλγόριθμος μηχανικής μάθησης μπορεί να χρησιμοποιηθεί ως βασικός ταξινομητής εάν δέχεται βάρη στο σετ εκπαίδευσης. Ο Adaboost θα πρέπει να πληροί δύο προϋποθέσεις [9]:

- Ο ταξινομητής θα πρέπει να εκπαιδεύεται διαδραστικά σε διάφορα σταθμισμένα παραδείγματα εκπαίδευσης.
- Σε κάθε επανάληψη, προσπαθεί να παρέχει εξαιρετική εφαρμογή σε αυτά τα παραδείγματα ελαχιστοποιώντας τα σφάλματα εκπαίδευσης.

Πώς λειτουργεί ο αλγόριθμος AdaBoost

Λειτουργεί στα εξής βήματα:

- Αρχικά, ο Adaboost επιλέγει ένα υποσύνολο εκπαίδευσης τυχαία.
- Εκπαιδεύει επαναληπτικά το μοντέλο μηχανικής εκμάθησης AdaBoost επιλέγοντας το σετ εκπαίδευσης με βάση την ακριβή πρόβλεψη της τελευταίας εκπαίδευσης.
- Αποδίδει το μεγαλύτερο βάρος σε λανθασμένες ταξινομημένες παρατηρήσεις, έτσι ώστε στην επόμενη επανάληψη αυτές οι παρατηρήσεις να έχουν την υψηλή πιθανότητα ταξινόμησης.
- Επίσης, εκχωρεί το βάρος στον εκπαιδευμένο ταξινομητή σε κάθε επανάληψη σύμφωνα με την ακρίβεια του ταξινομητή. Ο πιο ακριβής ταξινομητής θα έχει μεγάλο βάρος.
- Αυτή η διαδικασία επαναλαμβάνεται μέχρι να χωρέσουν τα πλήρη δεδομένα εκπαίδευσης χωρίς κανένα σφάλμα ή μέχρι να φτάσει στον καθορισμένο μέγιστο αριθμό εκτιμητών.
- Για ταξινόμηση, εκτελεί μια ¨ψηφοφορία¨ σε όλους τους αλγόριθμους εκμάθησης που δημιούργησε.



Σχήμα 2.25: *Αλγόριθμος AdaBoost [9].*

2.3.13 Τι είναι ο αλγόριθμος Gradient Boosting Classifier ('gbc' - Gradient Boosting Classifier)

Τι είναι το Gradient Boosting

Όπως υποδηλώνει το όνομα, η εκμάθηση συνόλου περιλαμβάνει τη δημιουργία ενός ισχυρού μοντέλου χρησιμοποιώντας ένα σύνολο "ασθενέστερων" μοντέλων. Η ενίσχυση κλίσης εμπίπτει στην κατηγορία των μεθόδων ενίσχυσης, οι οποίες μαθαίνουν επαναληπτικά από κάθε έναν από τους αδύναμους μαθητές να χτίσουν ένα ισχυρό μοντέλο [30]. Μπορεί να βελτιστοποιήσει τα εξής:

- Regression
- Classification
- Ranking

Το πεδίο εφαρμογής αυτής της εργασίας θα περιοριστεί ιδιαίτερα στην ταξινόμηση. Η ιδέα πίσω από την ενίσχυση προέρχεται από τη διαίσθηση ότι οι αδύναμοι μαθητές θα μπορούσαν να τροποποιηθούν προκειμένου να γίνουν καλύτεροι. Ο AdaBoost ήταν ο πρώτος αλγόριθμος ενίσχυσης. Το AdaBoost και οι σχετικοί αλγόριθμοι χρησιμοποιήθηκαν για πρώτη φορά σε ένα στατιστικό πλαίσιο από τον Leo Breiman (1997), ο οποίος έθεσε τα θεμέλια για άλλους ερευνητές όπως ο Jerome H. Friedman να τροποποιήσουν αυτήν την εργασία στην ανάπτυξη του αλγορίθμου ενίσχυσης κλίσης για παλινδρόμηση. Στη συνέχεια, πολλοί ερευνητές ανέπτυξαν αυτόν τον αλγόριθμο ενίσχυσης για πολλά περισσότερα πεδία μηχανικής μάθησης και στατιστικών, πολύ πέρα από τις αρχικές εφαρμογές στην παλινδρόμηση και την ταξινόμηση [30].

Ο όρος "Gradient" στο Gradient Boosting αναφέρεται στο γεγονός ότι υπάρχουν δύο ή περισσότερες παράγωγοι της ίδιας συνάρτησης. Το Gradient Boosting είναι ένας επαναληπτικός αλγόριθμος λειτουργικής κλίσης, δηλαδή ένας αλγόριθμος που ελαχιστοποιεί μια συνάρτηση απώλειας επιλέγοντας επαναληπτικά μια συνάρτηση που δείχνει προς την αρνητική κλίση [30].

Gradient Boosting στην ταξινόμηση. Το Gradient Boosting έχει τρία κύρια συστατικά:

- Συνάρτηση απώλειας (Loss Function) - Ο ρόλος της συνάρτησης απώλειας είναι να εκτιμήσει πόσο καλό είναι το μοντέλο στο να κάνει προβλέψεις με τα υπάρχοντα δεδομένα. Αυτό μπορεί να διαφέρει ανάλογα με το πρόβλημα. Για παράδειγμα, αν προσπαθούμε να προβλέψουμε το βάρος ενός ατόμου ανάλογα με ορισμένες μεταβλητές εισόδου (πρόβλημα παλινδρόμησης), τότε η συνάρτηση απώλειας θα είναι κάτι που μας βοηθά να βρούμε τη διαφορά μεταξύ των προβλεπόμενων βαρών και των παρατηρούμενων βαρών. Από την άλλη πλευρά, εάν προσπαθούμε να κατηγοριοποιήσουμε εάν σε ένα άτομο αρέσει μια συγκεκριμένη ταινία με βάση την προσωπικότητά του, θα χρειαστούμε μια συνάρτηση απώλειας που θα μας βοηθά να κατανοήσουμε πόσο ακριβές είναι το μοντέλο μας στην ταξινόμηση ατόμων που το έκαναν ή δεν το έκαναν. Όπως ορισμένες ταινίες [30].
- Αδύναμος μαθητής (Weak Learner) - Αδύναμος μαθητής είναι αυτός που ταξινομεί τα δεδομένα μας αλλά το κάνει άσχημα, ίσως όχι καλύτερα από την τυχαία εικασία. Με

άλλα λόγια, έχει υψηλό ποσοστό σφαλμάτων. Αυτά είναι συνήθως δέντρα απόφασης [30].

- Προσθετικό μοντέλο (Additive Model) - Αυτή είναι η επαναληπτική και διαδοχική προσέγγιση της προσθήκης των δέντρων (αδύναμοι μαθητές) ένα βήμα τη φορά. Μετά από κάθε επανάληψη, πρέπει να είμαστε πιο κοντά στο τελικό μας μοντέλο. Με άλλα λόγια, κάθε επανάληψη θα πρέπει να μειώνει την αξία της συνάρτησης απώλειας [30].

Η θεωρία πίσω από το Gradient Boosting

Ο ταξινομητής ενίσχυσης κλίσης εξαρτάται από μια συνάρτηση απώλειας. Μπορεί να χρησιμοποιηθεί μια προσαρμοσμένη συνάρτηση απώλειας και πολλές τυποποιημένες συναρτήσεις απώλειας υποστηρίζονται από ταξινομητές ενίσχυσης κλίσης, αλλά η συνάρτηση απώλειας πρέπει να είναι διαφοροποιήσιμη [10].

Οι αλγόριθμοι ταξινόμησης χρησιμοποιούν συχνά λογαριθμική απώλεια, ενώ οι αλγόριθμοι παλινδρόμησης μπορούν να χρησιμοποιήσουν τετράγωνα σφάλματα. Τα συστήματα ενίσχυσης κλίσης δεν χρειάζεται να παράγουν μια νέα συνάρτηση απώλειας κάθε φορά που προστίθεται ο αλγόριθμος ενίσχυσης, αλλά μπορεί να εφαρμοστεί στο σύστημα οποιαδήποτε συνάρτηση διαφοροποιήσιμης απώλειας [10].

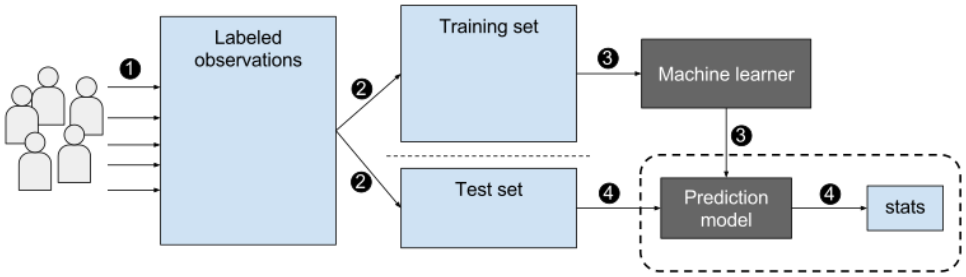
Τα συστήματα ενίσχυσης κλίσης έχουν δύο άλλα απαραίτητα μέρη: έναν αδύναμο μαθητή και ένα πρόσθετο στοιχείο. Τα συστήματα ενίσχυσης κλίσης χρησιμοποιούν δέντρα απόφασης ως αδύναμους μαθητές τους. Τα δέντρα παλινδρόμησης χρησιμοποιούνται για τους αδύναμους μαθητές και αυτά τα δέντρα παλινδρόμησης εξάγουν πραγματικές τιμές. Επειδή οι έξοδοι είναι πραγματικές τιμές, καθώς οι νέοι μαθητές προστίθενται στο μοντέλο, η έξοδος των δέντρων παλινδρόμησης μπορεί να προστεθεί μαζί για να διορθωθούν λάθη στις προβλέψεις [10].

Το πρόσθετο στοιχείο ενός μοντέλου ενίσχυσης κλίσης προέρχεται από το γεγονός ότι τα δέντρα προστίθενται στο μοντέλο με την πάροδο του χρόνου και όταν συμβαίνει αυτό, τα υπάρχοντα δέντρα δεν χρησιμοποιούνται, οι τιμές τους παραμένουν σταθερές.

Χρησιμοποιείται μια διαδικασία παρόμοια με την gradient descent για την ελαχιστοποίηση του σφάλματος μεταξύ των δεδομένων παραμέτρων. Αυτό γίνεται λαμβάνοντας την υπολογιζόμενη απώλεια και εκτελώντας κλίση κατάβασης για να μειωθεί αυτή η απώλεια. Στη συνέχεια, οι παράμετροι του δέντρου τροποποιούνται για να μειωθεί η υπολειπόμενη απώλεια.

Στη συνέχεια, η έξοδος του νέου δέντρου προσαρτάται στην έξοδο των προηγούμενων δέντρων που χρησιμοποιήθηκαν στο μοντέλο. Αυτή η διαδικασία επαναλαμβάνεται έως ότου επιτευχθεί ένας προκαθορισμένος αριθμός δέντρων ή η απώλεια μειωθεί κάτω από ένα συγκεκριμένο όριο [10].

Τα βήματα για την εφαρμογή του Gradient Boosting



Σχήμα 2.26: *Gradient Boosting [10].*

Για να εφαρμόσουμε έναν ταξινομητή ενίσχυσης κλίσης, θα χρειαστεί να εκτελέσουμε μια σειρά από διαφορετικά βήματα. Θα χρειαστεί να:

- Τοποθετήσουμε το μοντέλο
- Συντονίσουμε τις παραμέτρους και τις υπερπαραμέτρους του μοντέλου
- Κάνουμε προβλέψεις
- Ερμηνεύσουμε τα αποτελέσματα

Η τοποθέτηση μοντέλων με το Scikit-Learn είναι αρκετά εύκολη, καθώς συνήθως πρέπει απλώς να καλέσουμε την εντολή fit() μετά τη ρύθμιση του μοντέλου [10].

Ωστόσο, ο συντονισμός των υπερπαραμέτρων του μοντέλου απαιτεί κάποια ενεργή λήψη αποφάσεων από την πλευρά μας. Υπάρχουν διάφορα ορίσματα/υπερπαράμετροι που μπορούμε να συντονίσουμε για να προσπαθήσουμε να επιτύχουμε την καλύτερη ακρίβεια για το μοντέλο. Ένας από τους τρόπους με τους οποίους μπορούμε να το κάνουμε αυτό είναι αλλάζοντας τον ρυθμό εκμάθησης του μοντέλου. Θα θέλαμε να ελέγξουμε την απόδοση του μοντέλου στο σετ εκπαίδευσης με διαφορετικούς ρυθμούς εκμάθησης και, στη συνέχεια, να χρησιμοποιήσουμε τον καλύτερο ρυθμό εκμάθησης για να κάνουμε προβλέψεις.

Οι προβλέψεις μπορούν να γίνουν στο Scikit-Learn πολύ απλά χρησιμοποιώντας τη συνάρτηση predict() μετά την τοποθέτηση του ταξινομητή. Μπορεί κανείς να προβλέπει τα χαρακτηριστικά του συνόλου δεδομένων δοκιμής και, στη συνέχεια, να συγκρίνει τις προβλέψεις με τις πραγματικές ετικέτες. Η διαδικασία αξιολόγησης ενός ταξινομητή τυπικά περιλαμβάνει τον έλεγχο της ακρίβειας του ταξινομητή και στη συνέχεια την προσαρμογή των παραμέτρων/υπερπαραμέτρων του μοντέλου έως ότου ο ταξινομητής έχει μια ακρίβεια με την οποία είναι ικανοποιημένος ο χρήστης [10].

Δίαφορα μοντέλα βελτιωμένου Gradient Boosting είναι τα Penalized Learning, Tree Constraints, Random Sampling/Stochastic Boosting, Shrinkage/Weighted Updates και XGBoost.

2.3.14 Τι είναι ο αλγόριθμος Linear Discriminant Analysis ('lda' - Linear Discriminant Analysis)

Η διακριτική ανάλυση περιλαμβάνει μεθόδους που μπορούν να χρησιμοποιηθούν τόσο για ταξινόμηση όσο και για μείωση διαστάσεων. Η ανάλυση γραμμικής διάκρισης (LDA) είναι ιδιαίτερα δημοφιλής επειδή είναι ταυτόχρονα ένας ταξινομητής και μια τεχνική μείωσης διαστάσεων. Η ανάλυση τετραγωνικής διάκρισης (QDA) είναι μια παραλλαγή του LDA που επιτρέπει τον μη γραμμικό διαχωρισμό δεδομένων [31].

Linear discriminant analysis

Το LDA είναι μια τεχνική ταξινόμησης και μείωσης διαστάσεων, η οποία μπορεί να ερμηνευθεί από δύο οπτικές γωνίες. Το πρώτο είναι ότι η ερμηνεία είναι πιθανολογική και η δεύτερη, η πιο διαδικαστική ερμηνεία, οφείλεται στον Fisher. Η πρώτη ερμηνεία είναι χρήσιμη για την κατανόηση των υποθέσεων του LDA. Η δεύτερη ερμηνεία επιτρέπει την καλύτερη κατανόηση του τρόπου με τον οποίο το LDA εκτελεί τη μείωση διαστάσεων [31].

Έστω ένα πρόβλημα ταξινόμησης με μια τυχαία μεταβλητή X, η οποία προέρχεται από μία από τις κλάσεις K, με ορισμένες πυκνότητες πιθανότητας για συγκεκριμένη κατηγορία f(x). Ένας κανόνας διάκρισης προσπαθεί να διαιρέσει το χώρο δεδομένων σε K ασυνεχείς περιοχές που αντιπροσωπεύουν όλες τις τάξεις (όπως τα κουτιά σε μια σκακιέρα). Με αυτές τις περιοχές, η ταξινόμηση με ανάλυση διάκρισης σημαίνει απλώς ότι εκχωρούμε το x στην κλάση j εάν το x βρίσκεται στην περιοχή j. Το ερώτημα είναι τότε, πώς γνωρίζουμε σε ποια περιοχή εμπίπτουν τα δεδομένα x. Φυσικά, μπορούμε να ακολουθήσουμε δύο κανόνες κατανομής [31]:

Κανόνας μέγιστης πιθανότητας: Αν υποθέσουμε ότι κάθε κλάση θα μπορούσε να εμφανιστεί με ίσες πιθανότητες, τότε κατανείμεται το x στην κλάση j αν

$$j = argmax_i f_i(x)$$

Κανόνας Bayes: Αν γνωρίζουμε τις προηγούμενες πιθανότητες της κλάσης, π, τότε εκχωρούμε το x στην κλάση j αν

$$j = argmax_i \pi_i f_i(x)$$

Αν υποθέσουμε ότι τα δεδομένα προέρχονται από πολυμεταβλητή κατανομή Gauss, δηλαδή η κατανομή του X μπορεί να χαρακτηριστεί από τον μέσο όρο (μ) και τη συνδιακύμανσή του (Σ), μπορούν να ληφθούν ρητές μορφές των παραπάνω κανόνων κατανομής. Ακολουθώντας τον κανόνα Bayes, ταξινομούμε τα δεδομένα x στην κλάση j εάν έχουν την υψηλότερη πιθανότητα μεταξύ όλων των κλάσεων K για i = 1,. . . ,K:

$$\delta_i(x) = log f_i(x) + log \pi_i$$

Η παραπάνω συνάρτηση ονομάζεται συνάρτηση διάκρισης. Σημαντική είναι η χρήση του log-likelihood σε αυτό το σημείο. Με μια άλλη λέξη, η συνάρτηση διάκρισης μας λέει πόσο πιθανά είναι τα δεδομένα x από κάθε κλάση. Το όριο απόφασης που χωρίζει οποιεσδήποτε δύο κλάσεις, k και l, επομένως, είναι το σύνολο του x όπου δύο διακριτικές συναρτήσεις

έχουν την ίδια τιμή. Επομένως, τυχόν δεδομένα που εμπίπτουν στο όριο απόφασης είναι εξίσου πιθανά να ανήκουν στις δύο κατηγορίες [31].

Το LDA προκύπτει στην περίπτωση όπου υποθέτουμε ίση συνδιακύμανση μεταξύ των κλάσεων Κ. Δηλαδή, αντί για έναν πίνακα συνδιακύμανσης ανά κλάση, όλες οι κλάσεις έχουν τον ίδιο πίνακα συνδιακύμανσης. Τότε μπορούμε να λάβουμε την ακόλουθη συνάρτηση διάκρισης:

$$\delta_k(x) = X^T\Sigma^{-1}\mu_k - \frac{1}{2}\mu_k^T\Sigma^{-1}\mu_k + log\pi_k$$

2.3.15 Τι είναι ο αλγόριθμος Extra Trees Classifier ('et' - Extra Trees Classifier)

Το Extra Trees είναι ένας αλγόριθμος μηχανικής εκμάθησης συνόλου που συνδυάζει τις προβλέψεις από πολλά δέντρα αποφάσεων. Σχετίζεται με τον ευρέως χρησιμοποιούμενο αλγόριθμο τυχαίων δασών. Συχνά μπορεί να επιτύχει εξίσου καλή ή καλύτερη απόδοση από τον αλγόριθμο τυχαίων δασών, αν και χρησιμοποιεί έναν απλούστερο αλγόριθμο για την κατασκευή των δέντρων αποφάσεων που χρησιμοποιούνται ως μέλη του συνόλου. Είναι επίσης εύκολο στη χρήση δεδομένου ότι έχει λίγες βασικές υπερπαραμέτρους και λογικές ευρετικές μεθόδους για τη διαμόρφωση αυτών των υπερπαραμέτρων [32].

Αλγόριθμος Extra Trees

Το Extremely Randomized Trees, ή Extra Trees για συντομία, είναι ένας αλγόριθμος μηχανικής εκμάθησης συνόλου. Συγκεκριμένα, είναι ένα σύνολο δέντρων αποφάσεων και σχετίζεται με άλλα σύνολα αλγορίθμων δέντρων απόφασης όπως το bootstrap aggregation (bagging) και το τυχαίο δάσος. Ο αλγόριθμος Extra Trees λειτουργεί δημιουργώντας έναν μεγάλο αριθμό μη κλαδευμένων δέντρων αποφάσεων από το σύνολο δεδομένων εκπαίδευσης. Οι προβλέψεις γίνονται με τον μέσο όρο της πρόβλεψης των δέντρων απόφασης σε περίπτωση παλινδρόμησης ή χρησιμοποιώντας την πλειοψηφία στην περίπτωση ταξινόμησης [32].

- Παλινδρόμηση: Προβλέψεις που γίνονται με τον μέσο όρο των προβλέψεων από τα δέντρα αποφάσεων.
- Ταξινόμηση: Προβλέψεις που γίνονται με πλειοψηφία από δέντρα αποφάσεων.

Οι προβλέψεις των δέντρων συγκεντρώνονται για να δώσουν την τελική πρόβλεψη, με πλειοψηφία στα προβλήματα ταξινόμησης και με αριθμητικό μέσο όρο στα προβλήματα παλινδρόμησης.

Σε αντίθεση με το bagging και το τυχαίο δάσος που αναπτύσσουν κάθε δέντρο αποφάσεων από ένα δείγμα bootstrap του συνόλου δεδομένων εκπαίδευσης, ο αλγόριθμος Extra Trees ταιριάζει σε κάθε δέντρο απόφασης σε ολόκληρο το σύνολο δεδομένων εκπαίδευσης. Όπως και το τυχαίο δάσος, ο αλγόριθμος Extra Trees θα δειγματίσει τυχαία τα χαρακτηριστικά σε κάθε σημείο διαχωρισμού ενός δέντρου αποφάσεων. Σε αντίθεση με το τυχαίο δάσος, το οποίο χρησιμοποιεί έναν άπληστο αλγόριθμο για να επιλέξει ένα βέλτιστο σημείο διαχωρισμού, ο αλγόριθμος Extra Trees επιλέγει ένα σημείο διαίρεσης τυχαία. Ο αλγόριθμος Extra-Trees

δημιουργεί ένα σύνολο μη κλαδευμένων δέντρων αποφάσεων ή παλινδρόμησης σύμφωνα με την κλασική διαδικασία από πάνω προς τα κάτω. Οι δύο κύριες διαφορές του με άλλες μεθόδους συνόλου που βασίζονται σε δέντρα είναι ότι χωρίζει τους κόμβους επιλέγοντας σημεία αποκοπής εντελώς τυχαία και ότι χρησιμοποιεί ολόκληρο το δείγμα εκμάθησης (και όχι ένα αντίγραφο bootstrap) για την ανάπτυξη των δέντρων [32].

Ως εκ τούτου, υπάρχουν τρεις κύριες υπερπαράμετροι για συντονισμό στον αλγόριθμο. Είναι ο αριθμός των δέντρων απόφασης στο σύνολο, ο αριθμός των χαρακτηριστικών εισόδου που πρέπει να επιλεγούν τυχαία και να ληφθούν υπόψη για κάθε σημείο διαίρεσης και ο ελάχιστος αριθμός δειγμάτων που απαιτούνται σε έναν κόμβο για τη δημιουργία ενός νέου σημείου διαχωρισμού. Έχει δύο παραμέτρους: K, τον αριθμό των χαρακτηριστικών που επιλέγονται τυχαία σε κάθε κόμβο και nmin, το ελάχιστο μέγεθος δείγματος για τη διαίρεση ενός κόμβου. Συμβολίζουμε με M τον αριθμό των δέντρων αυτού του συνόλου [32].

Η τυχαία επιλογή των σημείων διαχωρισμού καθιστά τα δέντρα απόφασης στο σύνολο λιγότερο συσχετισμένα, αν και αυτό αυξάνει τη διακύμανση του αλγορίθμου. Αυτή η αύξηση της διακύμανσης μπορεί να αντιμετωπιστεί αυξάνοντας τον αριθμό των δέντρων που χρησιμοποιούνται στο σύνολο. Οι παράμετροι K, nmin και M έχουν διαφορετικά αποτελέσματα: K καθορίζει την ισχύ της διαδικασίας επιλογής χαρακτηριστικών, nmin την ισχύ του μέσου όρου του θορύβου εξόδου και M την ισχύ της μείωσης της διακύμανσης της συνάθροισης του μοντέλου συνόλου [32].

2.3.16 Τι είναι ο αλγόριθμος Extreme Gradient Boosting ('xgboost' - Extreme Gradient Boosting')

Σε αντίθεση με την κανονική ενίσχυση κλίσης, το XGBoost χρησιμοποιεί τη δική του μέθοδο κατασκευής δέντρων όπου το Similarity Score και το Gain καθορίζουν τον καλύτερο διαχωρισμό κόμβων [33].

$$SimilarityScore = \frac{(\sum_{i=1}^{n} Residual_i)^2}{\sum_{i=1}^{n}[PreviousProbability_i * (1 - PreviousProbability_i)] + \lambda}$$

- Το υπόλοιπο είναι η πραγματική (παρατηρούμενη) τιμή μείον την προβλεπόμενη τιμή.
- Προηγούμενη πιθανότητα είναι η πιθανότητα ενός συμβάντος που υπολογίστηκε σε προηγούμενο βήμα. Η αρχική πιθανότητα θεωρείται ότι είναι 0,5 για κάθε παρατήρηση, η οποία χρησιμοποιείται για την κατασκευή του πρώτου δέντρου. Για οποιαδήποτε επόμενα δέντρα, η προηγούμενη πιθανότητα υπολογίζεται εκ νέου με βάση την αρχική πρόβλεψη και τις προβλέψεις από όλα τα προηγούμενα δέντρα, όπως φαίνεται στον χάρτη διεργασιών.
- Το λάμδα είναι μια παράμετρος κανονικοποίησης. Η αύξηση του λάμδα μειώνει δυσανάλογα την επίδραση των μικρών φύλλων (αυτά με λίγες παρατηρήσεις) ενώ έχει μόνο μικρή επίδραση στα μεγαλύτερα φύλλα (αυτά με πολλές παρατηρήσεις).

Μόλις έχουμε το Similarity Score για κάθε φύλλο, υπολογίζουμε το Gain χρησιμοποιώντας τον ακόλουθο τύπο [33]:

$$Gain = Leftleaf_{similarity} + Rightleaf_{similarity} - Root_{similarity}$$

Στη συνέχεια, ο διαχωρισμός κόμβου με το υψηλότερο κέρδος επιλέγεται ως ο καλύτερος διαχωρισμός για το δέντρο.

Υπολογισμός τιμών εξόδου

Ο τρόπος υπολογισμού της τιμής εξόδου για κάθε φύλλο είναι σχεδόν πανομοιότυπος μεταξύ των δύο αλγορίθμων, με μόνη διαφορά την υπερπαράμετρο λάμδα [33].

Ο αλγόριθμος Gradient Boosting χρησιμοποιεί τον ακόλουθο τύπο:

$$Value = \frac{\sum_{i=1}^{n} Residual_i}{\sum_{i=1}^{n}[PreviousProbability_i * (1 - PreviousProbability_i)]}$$

Εν τω μεταξύ, το XGBoost χρησιμοποιεί:

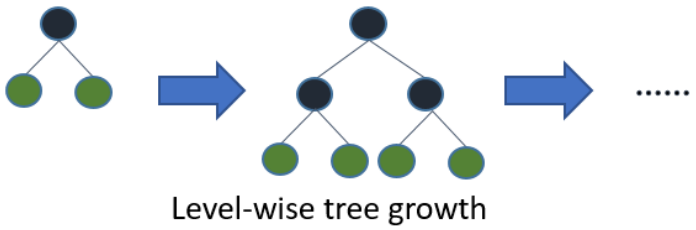
$$Value = \frac{\sum_{i=1}^{n} Residual_i}{\sum_{i=1}^{n}[PreviousProbability_i * (1 - PreviousProbability_i)] + \lambda}$$

2.3.17 Τι είναι ο αλγόριθμος Light Gradient Boosting Machine ('lightgbm' - Light Gradient Boosting Machine)

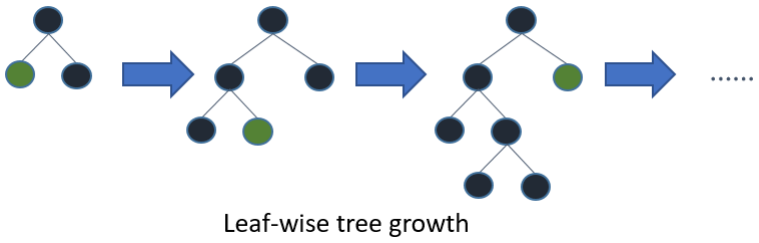
Το Light GBM είναι ένα γρήγορο, κατανεμημένο πλαίσιο ενίσχυσης κλίσης υψηλής απόδοσης που βασίζεται στον αλγόριθμο δέντρου αποφάσεων, που χρησιμοποιείται για την κατάταξη, την ταξινόμηση και πολλές άλλες εργασίες μηχανικής μάθησης.

Δεδομένου ότι βασίζεται σε αλγόριθμους δέντρων αποφάσεων, διαχωρίζει το δέντρο κατά φύλλα με την καλύτερη προσαρμογή, ενώ άλλοι αλγόριθμοι ενίσχυσης διαχωρίζουν το δέντρο κατά βάθος ή στάθμη αντί για φύλλα. Έτσι, όταν μεγαλώνει στο ίδιο φύλλο στο Light GBM, ο αλγόριθμος με βάση τα φύλλα μπορεί να μειώσει περισσότερες απώλειες από τον αλγόριθμο στάθμης και ως εκ τούτου οδηγεί σε πολύ καλύτερη ακρίβεια που σπάνια μπορεί να επιτευχθεί με οποιονδήποτε από τους υπάρχοντες αλγόριθμους ενίσχυσης. Επίσης, είναι εκπληκτικά πολύ γρήγορο, εξ ου και η λέξη ¨Φως¨ [11].

Ακολούθως παρατίθεται μια διαγραμματική αναπαράσταση από τους κατασκευαστές του Light GBM για να εξηγηθεί με σαφήνεια η διαφορά.



Σχήμα 2.27: *Ανάπτυξη δέντρων ανά επίπεδο στο XGBOOST [11].*



Σχήμα 2.28: *Ανάπτυξη δέντρων ανά φύλλο στο Light GBM [11].*

Πλεονεκτήματα του Light GBM

- Γρηγορότερη ταχύτητα εκπαίδευσης και υψηλότερη απόδοση: Το Light GBM χρησιμοποιεί αλγόριθμο που βασίζεται σε ιστογράμματα, δηλαδή τοποθετεί τις τιμές συνεχών χαρακτηριστικών σε ξεχωριστούς κάδους που στερεώνουν τη διαδικασία εκπαίδευσης.
- Χαμηλότερη χρήση μνήμης: Αντικαθιστά τις συνεχείς τιμές σε διακριτούς κάδους που οδηγούν σε χαμηλότερη χρήση μνήμης.
- Καλύτερη ακρίβεια από οποιονδήποτε άλλον αλγόριθμο ενίσχυσης: Παράγει πολύ πιο πολύπλοκα δέντρα ακολουθώντας την προσέγγιση σοφής διαχωρισμού των φύλλων αντί για μια προσέγγιση σε επίπεδο επίπεδο που είναι ο κύριος παράγοντας για την επίτευξη υψηλότερης ακρίβειας. Ωστόσο, μερικές φορές μπορεί να οδηγήσει σε υπερπροσαρμογή η οποία μπορεί να αποφευχθεί ορίζοντας την παράμετρο $\max_{depth}$.
- Συμβατότητα με μεγάλα σύνολα δεδομένων: Είναι σε θέση να αποδίδει εξίσου καλά με μεγάλα σύνολα δεδομένων με σημαντική μείωση του χρόνου εκπαίδευσης σε σύγκριση με το XGBOOST.
- Υποστηρίζεται η παράλληλη μάθηση.

2.3.18 Τι είναι ο αλγόριθμος CatBoost Classifier ('catboost' - CatBoost Classifier)

Ο CatBoost είναι ένας πρόσφατα ανοιχτού κώδικα αλγόριθμος μηχανικής εκμάθησης από τη Yandex. Μπορεί εύκολα να ενσωματωθεί με πλαίσια βαθιάς μάθησης όπως το TensorFlow της Google και το Core ML της Apple. Μπορεί να λειτουργήσει με διάφορους τύπους δεδομένων για να βοηθήσει στην επίλυση ενός ευρέος φάσματος προβλημάτων που αντιμετωπίζουν οι επιχειρήσεις σήμερα. Παρέχει ακόμη την καλύτερη ακρίβεια στην κατηγορία του [34].

Είναι ιδιαίτερα ισχυρό με δύο τρόπους:

Αποδίδει αποτελέσματα αιχμής χωρίς εκτενή εκπαίδευση δεδομένων που συνήθως απαιτείται από άλλες μεθόδους μηχανικής εκμάθησης και παρέχει ισχυρή υποστήριξη για τις πιο περιγραφικές μορφές δεδομένων που συνοδεύουν πολλά επιχειρηματικά προβλήματα. Το όνομα "CatBoost" προέρχεται από δύο λέξεις ¨Κατηγορία¨ και ¨Ενίσχυση¨ [34].

Όπως αναφέρθηκε, η βιβλιοθήκη λειτουργεί καλά με πολλές κατηγορίες δεδομένων, όπως ήχος, κείμενο, εικόνα, συμπεριλαμβανομένων ιστορικών δεδομένων.

Το "Boost" προέρχεται από τον αλγόριθμο μηχανικής εκμάθησης που ενισχύει τη διαβάθμιση, καθώς αυτή η βιβλιοθήκη βασίζεται στη βιβλιοθήκη ενίσχυσης κλίσης. Η ενίσχυση κλίσης είναι ένας ισχυρός αλγόριθμος μηχανικής εκμάθησης που εφαρμόζεται ευρέως σε πολλαπλούς τύπους επιχειρηματικών προκλήσεων όπως ο εντοπισμός απάτης, τα στοιχεία συστάσεων, η πρόβλεψη και έχει επίσης καλή απόδοση. Μπορεί επίσης να επιστρέψει πολύ καλό αποτέλεσμα με σχετικά λιγότερα δεδομένα, σε αντίθεση με τα μοντέλα DL που πρέπει να μάθουν από έναν τεράστιο όγκο δεδομένων [34].

Πλεονεκτήματα της βιβλιοθήκης CatBoost

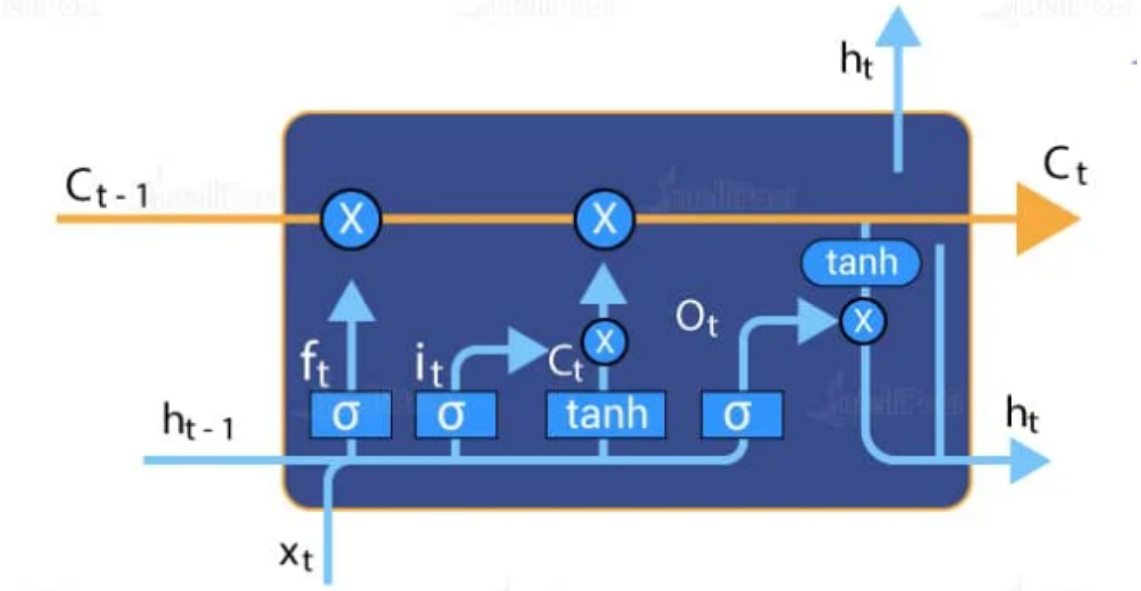
- Απόδοση: Το CatBoost παρέχει αποτελέσματα αιχμής και είναι ανταγωνιστικό με οποιονδήποτε κορυφαίο αλγόριθμο μηχανικής εκμάθησης στο μέτωπο της απόδοσης.
- Αυτόματος χειρισμός κατηγοριών χαρακτηριστικών: Μπορούμε να χρησιμοποιήσουμε το CatBoost χωρίς καμία ρητή προεπεξεργασία για να μετατρέψουμε κατηγορίες σε αριθμούς. Το CatBoost μετατρέπει τις κατηγορικές τιμές σε αριθμούς χρησιμοποιώντας διάφορα στατιστικά στοιχεία για συνδυασμούς κατηγορικών χαρακτηριστικών και συνδυασμούς κατηγορικών και αριθμητικών χαρακτηριστικών.
- Στιβαρό/Ανθεκτικό: Μειώνει την ανάγκη για εκτεταμένο συντονισμό υπερπαραμέτρων και μειώνει τις πιθανότητες υπερβολικής προσαρμογής, γεγονός που οδηγεί σε πιο γενικευμένα μοντέλα. Παρόλο που, το CatBoost έχει πολλαπλές παραμέτρους για συντονισμό και περιέχει παραμέτρους όπως ο αριθμός των δέντρων, ο ρυθμός εκμάθησης, η κανονικοποίηση, το βάθος δέντρου, το μέγεθος διπλώματος, η θερμοκρασία σάκου και άλλες.
- Εύκολο στη χρήση: Μπορεί κανείς να χρησιμοποιήσει το CatBoost από τη γραμμή εντολών, χρησιμοποιώντας ένα φιλικό προς το χρήστη API τόσο για την Python όσο και για την R.

2.4 Θεωρία των Μοντέλων Βαθιάς Μάθησης που χρησιμοποιήθηκαν στην εν λόγω εργασία

2.4.1 Τι είναι ένα LSTM (Long short-term memory)

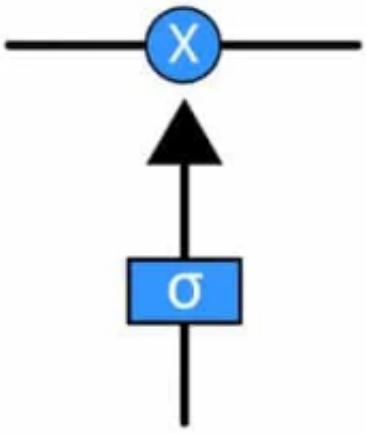
Πρώτα, πρέπει να αναρωτηθεί κανείς ¨Τι σημαίνει το LSTM¨. Το LSTM αντιπροσωπεύει τα δίκτυα μακροπρόθεσμης βραχυπρόθεσμης μνήμης, που χρησιμοποιούνται στον τομέα της βαθιάς μάθησης. Είναι μια ποικιλία επαναλαμβανόμενων νευρωνικών δικτύων (RNN) που είναι ικανά να μαθαίνουν μακροπρόθεσμες εξαρτήσεις, ειδικά σε προβλήματα πρόβλεψης ακολουθίας. Το LSTM έχει συνδέσεις ανάδρασης, δηλαδή, είναι ικανό να επεξεργάζεται ολόκληρη τη σειρά δεδομένων, εκτός από μεμονωμένα σημεία δεδομένων, όπως εικόνες. Αυτό βρίσκει εφαρμογή στην αναγνώριση ομιλίας, στη μηχανική μετάφραση κ.λπ. Το LSTM είναι ένα ειδικό είδος RNN, το οποίο παρουσιάζει εξαιρετική απόδοση σε μια μεγάλη ποικιλία προβλημάτων [12].

Ο κεντρικός ρόλος ενός μοντέλου LSTM κατέχει μια κυψέλη μνήμης γνωστή ως «κατάσταση κυψέλης» που διατηρεί την κατάστασή της με την πάροδο του χρόνου. Η κατάσταση κελιού είναι η οριζόντια γραμμή που διατρέχει την κορυφή του παρακάτω διαγράμματος. Μπορεί να απεικονιστεί ως ένας μεταφορικός ιμάντας μέσω του οποίου οι πληροφορίες απλώς ρέουν, αμετάβλητες [12].



Σχήμα 2.29: *Αρχιτεκτονική ενός LSTM [12].*

Οι πληροφορίες μπορούν να προστεθούν ή να αφαιρεθούν από την κατάσταση κυψέλης στο LSTM και ρυθμίζονται από πύλες. Αυτές οι πύλες προαιρετικά επιτρέπουν στις πληροφορίες να ρέουν μέσα και έξω από το κελί. Περιέχει μια σημειακή λειτουργία πολλαπλασιασμού και ένα στρώμα σιγμοειδούς νευρικού δικτύου που βοηθούν τον μηχανισμό [12].



Σχήμα 2.30: *Πύλες ενός LSTM [12].*

Το σιγμοειδές στρώμα δίνει αριθμούς μεταξύ μηδέν και ενός, όπου το μηδέν σημαίνει «δεν πρέπει να περάσει τίποτα» και το ένα σημαίνει «όλα πρέπει να περάσουν».

Εφαρμογές LSTM Τα δίκτυα LSTM βρίσκουν χρήσιμες εφαρμογές στους ακόλουθους τομείς:

- Μοντελοποίηση γλώσσας
- Μηχανική μετάφραση
- Αναγνώριση χειρογράφου
- Λεζάντα εικόνας
- Δημιουργία εικόνας με χρήση μοντέλων προσοχής
- Ερώτηση απάντηση
- Μετατροπή βίντεο σε κείμενο
- Πολυμορφική μουσική μοντελοποίηση
- Σύνθεση λόγου
- Πρόβλεψη δευτερογενούς δομής πρωτεΐνης

2.4.2 Τι είναι ένα TCN (Temporal Convolutional Network)

Στο σημείο αυτό θα αναλύσουμε τη δομή του TCN και τα βασικά αρχιτεκτονικά του στοιχεία. Είναι εμπνευσμένο από πρόσφατες συνελικτικές αρχιτεκτονικές για διαδοχικά δεδομένα και συνδυάζει απλότητα, αυτοπαλινδρομική πρόβλεψη και πολύ μεγάλη μνήμη. Το TCN έχει σχεδιαστεί με βάση δύο βασικές αρχές [13]:

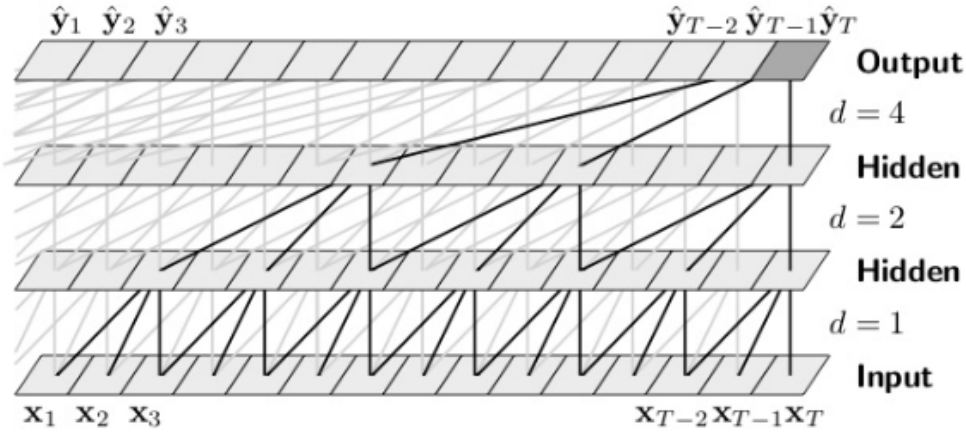
- Οι συνελίξεις είναι αιτιώδεις, που σημαίνει ότι δεν υπάρχει διαρροή πληροφοριών από το μέλλον στο παρελθόν.
- Η αρχιτεκτονική μπορεί να πάρει μια ακολουθία οποιουδήποτε μήκους και να την αντιστοιχίσει σε μια ακολουθία εξόδου του ίδιου μήκους όπως και με ένα RNN.

Για να επιτύχει το πρώτο σημείο, το TCN χρησιμοποιεί συνελίξεις αιτιατού, δηλ. συνελίξεις όπου μια έξοδος τη στιγμή t συνελίσσεται μόνο με στοιχεία από τη στιγμή t και νωρίτερα στο προηγούμενο επίπεδο. Για να ολοκληρώσει το δεύτερο σημείο, το TCN χρησιμοποιεί μια 1D πλήρως συνελικτική αρχιτεκτονική δικτύου, όπου κάθε κρυφό στρώμα έχει το ίδιο μήκος με το επίπεδο εισόδου [13].

Οι απλές αιτιώδεις συνελίξεις έχουν το μειονέκτημα να κοιτάζουν πίσω στην ιστορία μόνο με γραμμικό μέγεθος στο βάθος του δικτύου, δηλαδή το δεκτικό πεδίο μεγαλώνει γραμμικά με κάθε επιπλέον στρώμα. Για να παρακάμψει αυτό το γεγονός, η αρχιτεκτονική χρησιμοποιεί διευρυμένες περιελίξεις που επιτρέπουν ένα εκθετικά μεγάλο δεκτικό πεδίο. Πιο επίσημα, για μια ακολουθία εισαγωγής $x \in R^T$ και ένα φίλτρο $h : \{0, ..., k-1\} \Rightarrow R$, η λειτουργία διευρυμένης συνέλιξης H στο στοιχείο x της ακολουθίας ορίζεται ως

$$H(x) = (x *_d h)(x) = \sum_{i=0}^{k-1} f(i) x_{s-d*i} \tag{2.1}$$

όπου $d = 2^v$ είναι ο παράγοντας διαστολής, με v το επίπεδο του δικτύου και k είναι το μέγεθος του φίλτρου. Ο όρος s-d*i αντιπροσωπεύει την κατεύθυνση του παρελθόντος. Η διαστολή ισοδυναμεί με την εισαγωγή ενός σταθερού βήματος μεταξύ κάθε δύο παρακείμενες βρύσες φίλτρου, όπως φαίνεται στο παρακάτω σχήμα [13]:



Σχήμα 2.31: *Αρχιτεκτονική ενός TCN [13].*

Η χρήση μεγαλύτερης διαστολής επιτρέπει σε μια έξοδο στο ανώτερο επίπεδο να αντιπροσωπεύει ένα ευρύτερο φάσμα εισόδων, επεκτείνοντας έτσι αποτελεσματικά το δεκτικό πεδίο ενός CNN. Υπάρχουν δύο τρόποι για να αυξήσει κανείς το πεδίο λήψης ενός TCN: να επιλέξει μεγαλύτερα μεγέθη φίλτρων k και αύξηση του παράγοντα διαστολής d , αφού το αποτελεσματικό ιστορικό ενός στρώματος είναι (k-1)d .

Υπολειμματικό Μπλοκ (Residual Block)

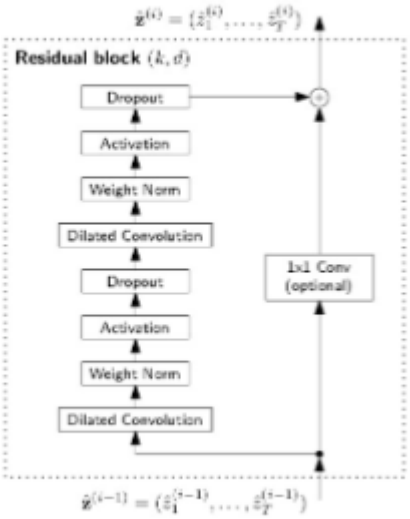
Ένα άλλο αρχιτεκτονικό στοιχείο ενός TCN είναι οι υπολειπόμενες συνδέσεις. Στη θέση ενός συνελικτικού στρώματος, τα TCN χρησιμοποιούν μια γενική υπολειμματική ενότητα. Κάθε υπολειπόμενο μπλοκ περιέχει έναν κλάδο που οδηγεί σε μια σειρά μετασχηματισμών F, των οποίων οι έξοδοι προστίθενται στην είσοδο x του μπλοκ [13].

$$o = Activation(x + F(x)) \tag{2.2}$$

Αυτό επιτρέπει αποτελεσματικά στα επίπεδα να μάθουν τροποποιήσεις στη χαρτογράφηση ταυτότητας αντί για ολόκληρο τον μετασχηματισμό, ο οποίος έχει αποδειχθεί ότι ωφελεί τα βαθιά νευρωνικά δίκτυα. Ειδικά για πολύ βαθιά δίκτυα, η σταθεροποίηση είναι σημαντική, για παράδειγμα, στην περίπτωση που η πρόβλεψη εξαρτάται από μεγάλο μέγεθος ιστορικού ($> 2^{12}$) με μια ακολουθία εισόδου υψηλών διαστάσεων.

Ένα υπολειπόμενο μπλοκ έχει δύο στρώματα διευρυμένων αιτιακών συνελίξεων και διορθωμένες γραμμικές μονάδες (ReLU) ως μη γραμμικότητες, όπως φαίνεται στο ακόλουθο σχήμα:

Η ομαλοποίηση βάρους εφαρμόζεται στα συνελικτικά φίλτρα και προστίθεται μια χωρική εγκατάλειψη μετά από κάθε διευρυμένη συνέλιξη για κανονικοποίηση, που σημαίνει ότι σε κάθε βήμα εκπαίδευσης μηδενίζεται ένα ολόκληρο κανάλι [13].



Σχήμα 2.32: *Αρχιτεκτονική υπολειπόμενου μπλοκ ενός TCN [13].*

Συμπεράσματα

Τα TCN μπορούν να κατασκευαστούν για να έχουν πολύ μεγάλα μεγέθη αποτελεσματικού ιστορικού, πράγμα που σημαίνει ότι έχουν την ικανότητα να κοιτάζουν πολύ μακριά στο παρελθόν για να κάνουν μια πρόβλεψη. Για το σκοπό αυτό, αναπτύσσεται ένας συνδυασμός πολύ βαθιών δικτύων επαυξημένων με υπολειμματικά στρώματα και διευρυμένες περιελίξεις [13].

Η αρχιτεκτονική TCN φαίνεται όχι μόνο να είναι πιο ακριβής από τα κανονικά επαναλαμβανόμενα δίκτυα όπως τα LSTM και τα GRU, αλλά περιέχει επίσης τις ακόλουθες ιδιότητες [13]:

- Παραλληλισμός: Σε αντίθεση με τα RNN όπου οι προβλέψεις για μεταγενέστερα χρονικά βήματα πρέπει να περιμένουν να ολοκληρωθούν οι προκάτοχοί τους, οι συνελίξεις μπορούν να υπολογιστούν παράλληλα επειδή χρησιμοποιείται το ίδιο φίλτρο σε κάθε επίπεδο. Επομένως, τόσο στην εκπαίδευση όσο και στην αξιολόγηση, μια μεγάλη ακολουθία εισόδου μπορεί να υποβληθεί σε επεξεργασία ως σύνολο, αντί για διαδοχικά όπως στα RNN.

- Ευέλικτο μέγεθος δεκτικού πεδίου: Το μέγεθος του δεκτικού πεδίου μπορεί να αλλάξει με πολλούς τρόπους. Για παράδειγμα, η στοίβαξη περισσότερων διεσταλμένων συνελικτικών στρωμάτων, η χρήση μεγαλύτερων συντελεστών διαστολής ή η αύξηση του μεγέθους του φίλτρου είναι όλες βιώσιμες επιλογές. Έτσι, τα TCN παρέχουν καλύτερο έλεγχο του μεγέθους της μνήμης του μοντέλου και προσαρμόζονται εύκολα σε διαφορετικούς τομείς.

- Χαμηλή απαίτηση μνήμης για εκπαίδευση: Ειδικά στην περίπτωση μιας μεγάλης ακολουθίας εισόδου, τα LSTM και τα GRU μπορούν εύκολα να χρησιμοποιήσουν πολλή

μνήμη για να αποθηκεύσουν τα μερικά αποτελέσματα για τις πολλαπλές πύλες κυψελών τους. Στα TCN, ωστόσο, τα φίλτρα μοιράζονται σε ένα επίπεδο, με τη διαδρομή πίσω διάδοσης να εξαρτάται μόνο από το βάθος του δικτύου.

Κεφάλαιο 3

Βιβλιογραφική Ανασκόπηση

Στο κεφάλαιο αυτό γίνεται μια πλήρης βιβλιογρφική ανασκόπηση στις πηγές που χρησιμοποιήθηκαν κατάλληλα, για την ορθότερη πρόβλεψη ενός αγώνα NBA με τη χρήση αλγορίθμων Μηχανικής Μάθησης. Οι πηγές αυτές αποτέλεσαν κινητήριο μοχλό της εργασίας και έμπνευση για την προσέγγιση που ακολουθήθηκε. Χρησιμοποιήθηκε σύνολο δεδομένων διαφορετικό από τη βιβλιογραφία και συνεπώς οι ανάγκες των δεδομένων μας οδήγησαν και σε ανάλογη προσέγγιση του προβλήματος, καθιστώντας μοναδική τη μεθοδολογία επίλυσής του.

3.1 Σχετικές εργασίες

Αν και η μηχανική μάθηση έχει ήδη χρησιμοποιηθεί στο χώρο της ανάλυσης αθλητικών γεγονότων, υπάρχει ακόμη πολύ μεγάλο περιθώριο βελτίωσης των μοντέλων που χρησιμοποιεί αυτή η τεχνολογία. Σε αυτή την ενότητα, το ενδιαφέρον μας συγκεντρώνεται σε τεχνικές μηχανικής μάθησης που χρησιμοποιούνται, με απώτερο σκοπό την πρόβλεψη του αποτελέσματος ενός αγώνα NBA.

Ο συγγραφέας Cao [35] κατασκεύασε ένα προβλεπτικό μοντέλο, με τη χρήση ιστορικών δεδομένων που αφορούσαν αγώνες του NBA, βασισμένο στον Naive Bayes ταξινομητή. Η μετρική στην οποία βασίστηκε η αξιολόγηση του μοντέλου ήταν και εδώ, όπως και στην εργασία μας αυτή, η ορθότητα (accuracy). Η ορθότητα του ταξινομητή στο τεστ σύνολο δεδομένων έφτασε το 65.82%

Μια ακόμη έρευνα που αφορά την πρόβλεψη του αποτελέσματος ενός αγώνα NBA πραγματοποιήθηκε με τη χρήση ενός τεχνητού Νευρωνικού Δικτύου από τους Loeffelhoz et al. [36]. Ένα υποσύνολο των αρχικών χαρακτηριστικών μεταγλωττίστηκε από τους συγγραφείς από μη επεξεργασμένα δεδομένα και χρησίμευσε ως είσοδος στα νευρωνικά δίκτυα. Εν τέλη, πολλοί ειδικοί στο χώρο της καλαθοσφαίρισης έφτασαν στο σημείο να χρησιμοποιούν αυτό το ANN μοντέλο για να συγκρίνουν το αποτέλεσμα του με τις αρχικές εκτιμήσεις τους. Το ANN ξεπέρασε το ποσοστό προβλέψεων των ειδικών και άγγιξε προβλέψεις μέχρι και 74.33% των συνολικών αγώνων [36]. Η μέθοδος παλινδρόμησης που κατασκευάστηκε πάνω σε ιστορικά δεδομένα, άνοιξε το δρόμο για αποτελεσματικές προβλέψεις στο NFL (National Football League). Πλέον μέσω αυτής της έρευνας μπορούν να προβλεφθούν οι επόμενοι νικητές του NFL.

Μέσω της οπισθοδιάδωσης, των αυτο-οργανούμενων χαρτών (SOM) και ορισμένων νευ-

ρωνικών δομών, επιτεύχθηκε η πρόβλεψη της έκβασης ενός αγώνα ποδοσφαίρου NFL από τον Purucker [37]. Κατόπιν δοκιμών ποικίλων μεθόδων εκπαίδευσης δεδομένων, κατέληξε στο συμπέρασμα ότι η οπισθοδιάδοση ήταν ο καλύτερος τρόπος για να παράξει κανείς ορθότητα μοντέλου, μεγαλύτερη από κάθε άλλο ειδικό στο άθλημα. Οι ειδικοί συγκεκριμένα βρίσκονταν γύρω στο 61%, ενώ το μοντέλο εκείνης της έρευνας άγγιζε ορθότητα κοντά στο 72%.

Τόσο ένα νευρωνικό δίκτυο, όσο και η χρήση οπισθοδιάδοσης ερευνήθηκαν από τον Καην, με στόχο και πάλι την πρόβλεψη ενός αγώνα NFL. Ο Purucker, όμως, επέκτεινε εκείνη την έρευνα αγγίζοντας ορθότητα στο 75%, ξεπερνώντας κατά πολύ το 61% των ειδικών. Για την εύρεση αυτών των αποτελεσμάτων έγινε χρήση των βοξ σορες των αγώνων, όπως ακριβώς και στην εν λόγω εργασία.

Οι Bunker και Thabtah [38] κατασκεύασαν ένα μοντέλο μηχανικής μάθησης, το οποίο ήταν ικανό να προβλέπει το αποτέλεσμα ενός αγώνα, βασισμένο σε χακατηριστικά που περιέγραφα την εικόνα μιας ομάδας. Οι συγγραφείς πρότειναν ότι με τη χρήση ANN μοντέλων μπορούν να φτάσουν μεγαλύτερες τιμές ορθότητας. Αναδεικνύουν ακόμη ζωτικής σημασίας προκλήσεις για το πώς πρέπει κανείς να προ-επεξεργαστεί κατάλληλα τα δεδομένα του.

Οι Haghighat et al. [39] αναγνώρισαν ένα πλήθος μοντέλων μηχανικής μάθησης που είναι περισσότερα συχνά χρησιμοποιούμενα στον κλάδο της ανάλυσης αθλητικών γεγονότων. Οι συγγραφείς αναλύουν την προβλεπτική ακρίβεια κάθε μεθόδου σύμφωνα με την ήδη υπάρχουσα βιβλιογραφία και σαν προσθήκη χρησιμοποιούνται ποικίλα διαφορετικά σύνολα δεδομένων από διαφορετικές πηγές, όπως το NBA, το NFL και άλλες ιστοσελίδες όπως [40, 41, 42].

Ο Cao [35] προτείνει ένα εργαλείο βασισμένο στη μηχανική μάθηση για την πρόβλεψη αγώνων NBA. Χαρακτηριστικά σχετικά με το NBA , όπως ατομικά στατιστικά παιχτών, στατιστικά αντιπάλων και αρχική πεντάδα παιχτών συλλέγονται από την επίσημη ιστοσελίδα του NBA και αποθηκεύονται κατάλληλα σε μια βάση δεδομένων, ώστε στη συνέχεια να δοθούν ως είσοδος σε μοντέλα μηχανικής μάθησης. Οι συγγραφείς εφάρμοσαν διαφορετικές τεχνικές μηχανικής μάθησης, συμπεριλαμβανομένων των Support Vector Machines (SVM) [43], της Λογιστικής Παλινδρόμησης [44], του Naïve Bayes [45] και των ANN [46] στο σύνολο δεδομένων που είχαν συλλέξει, για να παράξουν μοντέλα πρόβλεψης αποτελεσμάτων αγώνων στο χώρο της ανάλυσης αθλητικών γεγονότων. Στα ευρύματά τους αναδεικνύουν τη Λογιστική Παλινδρόμηση ως την τεχνική με τη βέλτιστη τιμή ορθότητας. Στη δική μας εργασία δεν λαμβάνουμε υπ όψιν ατομικά χαρακτηριστικά, παρά μόνο τεαμ βοξ σορες, τόσο για τη γηπεδούχο, όσο και για τη φιλοξενούμενη ομάδα κάθε αγώνα.

Οι Miljkovic et. Al [47] ερεύνησαν μια τεχνολογία μηχανικής μάθησης που θα μπορούσε να προβλέψει την έκβαση ενός αγώνα NBA. Για το σκοπό αυτό χρησιμοποίησαν μοντέλα όπως οι αλγόριθμοι KNN, SVM, τα Δέντρα αποφάσεων, Multivariate γραμμική παλινδρόμηση και Naïve Bayes [45, 48, 49, 50, 51]. Δεδομένα με 141 μεταβλητές σχετικά με το NBA για τις σεζόν 2009-2010 συλλέχθηκαν με στόχο την αξιολόγηση δύο μεθόδων μηχανικής μάθησης. Παραδείγματα τέτοιων μεταβλητών ήταν οι μέχρι τώρα νίκες μιας ομάδας, το πλήθος ηττών, το σερί νικών και οι νίκες σε σύγκριση με άλλες ομάδες. Τελικά κατέληξαν στο να προβλέπουν το αποτέλεσμα ενός αγώνα NBA με ποσοστό ορθότητας στο 67%.

Kopf [52] επισημαίνει ότι κάθε ομάδα στο NBA συνεργάζεται με τμήματα ανάλυσης δε-

δομένων, τα οποία συνεργάζονται με τους προπονητές για να βελτιστοποιήσουν τις ατομικές επιδόσεις των παιχτών. Στις αρχές του 2009, οι περισσότερες λίγκες χρησιμοποιούσαν βίντεο για να καταγράψουν τις κινήσεις της μπάλας και των παιχτών, δίνοντας έτσι ποικιλόμορφα δεδομένα, κατάλληλα ώστε να ανακαλύψουν το ποιοί παίκτες συνέβαλαν περισσότερο στις νίκες της ομάδας.

Ο Leider [53] ερεύνησε χαρακτηριστικά που μπορούσαν να επηρεάσουν σημαντικά την έκβαση ενός αγώνα NBA. Οι συγγραφείς εφάρμοσαν τεχνικές μηχανικής μάθησης, όπως η Λογιστική Παλινδρόμηση, η Γραμμική Παλινδρόμηση και τα ΑΝΝ. Συλλέχθηκαν δεδομένα NBA για τη σεζόν 2014 τόσο για την εκπαίδευση των δεδομένων, όσο και για το τεστάρισμα. Τα αποτελέσματα ανέδειξαν ως βέλτιστη τεχνική τη Λογιστική Παλινδρόμηση με ορθότητα να αγγίζει το 70%.

Οι Cheng et al. [54] ανέπτυξαν ένα μοντέλο πρόβλεψης αγώνων NBA βασισμένο στην αρχή της μέγιστης Εντροπίας. Οι συγγραφείς συνέλεξαν δεδομένα από τις σεζόν 2007-2008 μέχρι και 2014-2015 και κατασκεύασαν μοντέλα με ορθότητα να αγγίζει και το 74.4%.

Αυτές είναι μερικές από τις βασικότερες έρευνες στη σχετική βιβλιογραφία που ερευνήσαμε και αποτέλεσαν πηγή έμπνευσης για την εν λόγω διπλωματική εργασία. Παρατίθενται μερικές ακόμη βιβλιογραφικές αναφορές που αποτέλεσαν πηγή έμπνευσης για εμάς και τη μεθοδολογία που ακολουθήσαμε για να προσεγγίσουμε τη βέλτιστη λύση στο πρόβλημα της ορθότερης πρόβλεψης ενός αγώνα NBA [55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76].

Μέρος II

Πρακτικό Μέρος

Κεφάλαιο 4

Σύνολο Δεδομένων

Στο κεφάλαιο αυτό παρουσιάζονται το αρχικό σύνολο δεδομένων που χρησιμοποιήθηκε στην εν λόγω εργασία, καθώς και μετέπειτα εκδοχές του, που εξυπηρετούν την εύρεση βέλτιστων αποτελεσμάτων, όσον αφορά την ορθότητα των μοντέλων Μηχανικής Μάθησης που χρησιμοποιήθηκαν. Παρουσιάζονται ακόμη διαδικασίες προ-επεξεργασίας των δεδομένων, καθώς και η λογική στην κατασκευή των διαφορετικών εκδοχών του αρχικού συνόλου δεδομένων. Τα αποτελέσματα των μοντέλων θα παρουσιαστούν στην επόμενη ενότητα.

4.1 Σύνολο Δεδομένων - Χαρακτηριστικά

Στην ενότητα αυτή παρουσιάζεται το σύνολο δεδομένων που χρησιμοποιήθηκε στην εν λόγω εργασία και τα χαρακτηριστικά του.

Ο κόσμος της καλαθοσφαίρισης περιβάλλεται από αυτόν του στοιχήματος. Συνεπώς, η πληροφορία είναι πλούσια και εύκολα προσβάσιμη σε όλους. Επομένως, ήταν εύκολο να βρεθούν πολλά και χρήσιμα στατιστικά και πληροφορίες για τους αγώνες του NBA που μελετά η παρούσα εργασία. Η ιστοσελίδα (https://www.kaggle.com/datasets/) παρέχει πληροφορίες για όλους τους αγώνες των μεγάλων πρωταθλημάτων τα τελευταία χρόνια, με όλα τα απαραίτητα στατιστικά ενός αγώνα NBA και όχι μόνο, αλλά και για κάθε είδους σύνολο δεδομένων, σε μορφή .csv αρχείου.

Το σύνολο δεδομένων που χρησιμοποιήθηκε ονομάζεται “NBA Enhanced Box Score and Standings (2012 - 2018)” και ανακτήθηκε από το Kaggle (https://www.kaggle.com/datasets/pablote/nba-enhanced-stats), στο οποίο μπορεί κανείς να αντλήσει περισσότερες πληροφορίες για το σύνολο δεδομένων μας. Περιλαμβάνει όλα τα box scores για κάθε ομάδα του NBA για τις σεζόν 2012-2018 (6 σεζόν). Δεν επιλέχθηκε πιο πρόσφατο σύνολο δεδομένων, γιατί δεν βρέθηκε κάποιο άλλο με όλη την πληροφία που περιλαμβάνει το δικό μας, αλλά και επειδή η σεζόν 2019-2020 δεν είναι αντιπροσωπευτική, καθώς, για την αποφυγή της διασποράς του covid-19, πραγματοποιήθηκε σε ειδικές εγκαταστάσεις, καταργώντας πλέον την έννοια της έδρας. Συνεπώς, οι σεζόν 2012-2018 είναι πιο αντιπροσωπευτικές και μας καλύπτουν για τον σκοπό της ανάλυσής μας.

Να σημειωθεί ότι το σύνολο δεδομένων μας είναι ελαφρώς ανισσόροπο, δηλαδή, κατά 58% εκατό κερδίζει τον αγώνα η γηπεδούχος ομάδα, ενώ κατα 42% η φιλοξενούμενη, κατά μέσο όρο και στις 6 διαθέσιμες σεζόν. Συνεπώς, δεν χρειάζεται η εφαρμογή ούτε υποδειγματοληψίας, αλλά ούτε και υπερδειγματοληψίας. Η μετρική στην οποία θα πραγματοποιήσουμε

την αναζήτηση του βέλτιστου μοντέλου μηχανικής μάθησης στο πρόβλημά μας θα είναι η ορθότητα (accuracy).

Στην παρακάτω λίστα παρουσιάζονται αναλυτικά τα χαρακτηριστικά που επιλέξαμε από το αρχικό dataset. Το σύνολο δεδομένων μας αποτελείται από γραμμές και στήλες. Κάθε γραμμή αφορά έναν αγώνα και τα χαρακτηριστικά με τη λέξη "team" αφορούν την ομάδα που αναγράφεται ως 1η στον συγκεκριμένο αγώνα, ενώ εκείνα με τη λέξη "oppt" αφορούν την ομάδα που αναγράφεται ως 2η στον συγκεκριμένο αγώνα. Το χαρακτηριστικό "teamLoc" ή "opptLoc" είναι εκείνο που καθορίζει αν μια ομάδα είναι γηπεδούχος ή φιλοξενούμενη, ανάλογα με το αν αναγράφεται σε αυτό η τιμή "Home" ή "Away". Περισσότερες πληροφορίες μπορεί να βρεις κανείς και στα metadata του συνόλου δεδομένων στο Kaggle:

Το αρχικό σύνολο δεδομένων μας περιλαμβάνει 119 χαρακτηριστικά. Αναλυτικότερα:

1.'gmDate' - Η ημερομηνία του αγώνα

2.'gmTime' - Η ώρα έναρξης του αγώνα

3.'seasTyp'- Ο τύπος της σεζόν, δηλαδή Regular Season και όχι Play offs

4.'offLNm' - Επίθετο διαιτητή αγώνα

5.'offFNm' - Όνομα διατητή αγώνα

6.'teamAbbr' - Όνομα 1ης ομάδας

7.'teamConf' - Γεωγραφικό Διαμέρισμα 1ης ομάδας (Δύση ή Ανατολή)

8.'teamDiv' - Υποκατηγορία Γεωγραφιού Διαμερίσματος 1ης ομάδας (π.χ. Νοτιανατολικά, Κεντρικά κλπ.)

9.'teamLoc' - 1η ομάδα γηπεδούχος ή φιλοξενούμενη

10.'teamRslt' - αποτέλεσμα αγώνα πρώτης ομάδας (νίκησε ή έχασε)

11.'teamMin' - Συσσωρευμένα λεπτά παίκτη ανά ομάδα

12.'teamDayOff' - Πλήθος ημερών ξεκούρασης μέχρι τον επόμενο αγώνα

13.'teamPTS' - Πόντοι που πέτυχε η 1η ομάδα

14.'teamAST' - Ασσίστ που πέτυχε η 1η ομάδα

15.'teamTO' - Λάθη που πραγματοποίησε η 1η ομάδα

16.'teamSTL' - Κλεψίματα που πραγματοποίησε η 1η ομάδα

17.'teamBLK' - Μπλοκς που πραγματοποίησε η 1η ομάδα

18.'teamPF' - Σύνολο ατομικών φάουλ 1ης ομάδας

19.'teamFGA' - Σουτ εντός παιδιάς που έγιναν από την 1η ομάδα

20.'teamFGM' - Εύστοχα Σουτ εντός παιδιάς της 1ης ομάδας

21.'teamFG%' - Ποσοστό εύστοχων Σουτ εντός παιδιάς της 1ης ομάδας

22.'team2PA' - Σουτ 2 πόντων που έγιναν από την 1η ομάδα

23.'team2PM' - Εύστοχα Σουτ 2 πόντων της 1ης ομάδας

24.'team2P%' - Ποσοστό εύστοχων Σουτ 2 πόντων της 1ης ομάδας

25.'team3PA' - Σουτ 3 πόντων που έγιναν από την 1η ομάδα

26.'team3PM' - Εύστοχα Σουτ 3 πόντων της 1ης ομάδας

27.'team3P%' - Ποσοστό εύστοχων Σουτ 3 πόντων της 1ης ομάδας

28.'teamFTA' - Βολές που εκτελέστηκαν από την 1η ομάδα

29.'teamFTM' - Εύστοχες Βολές από την 1η ομάδα

30.'teamFT%' - Ποσοστό εύστοχων βολών της 1ης ομάδας

31.'teamORB' - Επιθετικά ριμπάουντ της 1ης ομάδας

32.'teamDRB' - Αμυντικά ριμπάουντ της 1ης ομάδας

33.'teamTRB' - Συνολικά ριμπάουντ της 1ης ομάδας

34.'teamPTS1' - Συνολικοί πόντοι στην 1η περίοδο της 1ης ομάδας

35.'teamPTS2' - Συνολικοί πόντοι στην 2η περίοδο της 1ης ομάδας

36.'teamPTS3' - Συνολικοί πόντοι στην 3η περίοδο της 1ης ομάδας

37.'teamPTS4' - Συνολικοί πόντοι στην 4η περίοδο της 1ης ομάδας

38.'teamPTS5' - Συνολικοί πόντοι στην 1η παράταση της 1ης ομάδας

39.'teamPTS6' - Συνολικοί πόντοι στην 2η παράταση της 1ης ομάδας

40.'teamPTS7' - Συνολικοί πόντοι στην 3η παράταση της 1ης ομάδας

41.'teamPTS8' - Συνολικοί πόντοι στην 4η παράταση της 1ης ομάδας

42.'teamTREB%' - Ποσοστό συνολικών ριμπάουντ 1ης ομάδας (Για παράδειγμα αν στον αγώνα έγιναν 50 ριμπάουντ και τα 25 ήταν της 1ης ομάδας, τότε το ποσοστό ισούται με 50, ομοίως...)

43.'teamASST%' - Ποσοστό συνολικών ασσίστ 1ης ομάδας

44.'teamTS%' - Πραγματικό ποσοστό σουτ που πραγματοποιήθηκαν από την 1η ομάδα

45.'teamEFG%' - Ποσοστό επιτυχίας σουτ εντός πεδιάς για την 1η ομάδα

46.'teamOREB%' - Ποσοστό επιθετικών ριμπάουντ 1ης ομάδας

47.'teamDREB%' - Ποσοστό αμυντικών ριμπάουντ 1ης ομάδας

48.'teamTO%' - Ποσοστό Λαθών που πραγματοποίησε η 1η ομάδα

49.'teamSTL%' - Ποσοστό Κλεψιμάτων που πραγματοποίησε η 1η ομάδα

50.'teamBLK%' - Ποσοστό Μπλοκς που πραγματοποίησε η 1η ομάδα

51.'teamBLKR' - Ρυθμός/Συχνότητα των Μπλοκς που πραγματοποίησε η 1η ομάδα

52.'teamPPS' - Πόντοι ανά σουτ της 1ης ομάδας

53.'teamFIC' - Φόρμουλα που περιέχει όλα τα box scores της 1ης ομάδας σε ένα στατιστικό

54.'teamFIC40' - Φόρμουλα που περιέχει όλα τα box scores της 1ης ομάδας σε ένα στατιστικό ανά 40 λεπτά

55.'teamOrtg' - Επιθετική Βαθμολογία (διαθέσιμη από τη σεζόν 1977-78 στο NBA). Είναι οι πόντοι που πετυχαίνει η 1η ομάδα ανά 100 κατοχές.

56.'teamDrtg' - Αμυντική Βαθμολογία (διαθέσιμη από τη σεζόν 1977-78 στο NBA). Είναι οι πόντοι που δέχεται η 1η ομάδα ανά 100 κατοχές.

57.'teamEDiff' - Η διαφορά των στατιστικών 55 και 56 για την 1η ομάδα

58.teamPlay%' - Ποσοστό παιξίματος για την 1η ομάδα

59.'teamAR' - Ρυθμός/Συχνότητα των Ασσίστ που πραγματοποίησε η 1η ομάδα

60.'teamAST/TO' - Ασσίστ ως προς τον ρυθμό λαθών που πραγματοποίησε η 1η ομάδα

61.'teamSTL/TO' - Κλεψίματα ως προς τον ρυθμό λαθών που πραγματοποίησε η 1η

Τα χαρακτηριστικά 62 έως και 117 είναι πανομοιότυπα με τα προηγούμενα με μόνη διαφορά, ότι αναφέρονται στην 2η ομάδα της εκάστοτε γραμμής δεδομένων.

62.'opptAbbr'

63.'opptConf'

64.'opptDiv'

65.'opptLoc'

66.'opptRslt'

67.'opptMin'
68.'opptDayOff'
69.'opptPTS'
70.'opptAST'
71.'opptTO'
72.'opptSTL'
73.'opptBLK'
74.'opptPF'
75.'opptFGA'
76.'opptFGM'
77.'opptFG%'
78.'oppt2PA'
79.'oppt2PM'
80.'oppt2P%'
81.'oppt3PA'
82.'oppt3PM'
83.'oppt3P%'
84.'opptFTA'
85.'opptFTM'
86.'opptFT%'
87.'opptORB'
88.'opptDRB'
89.'opptTRB'
90.'opptPTS1'
91.'opptPTS2'
92.'opptPTS3'
93.'opptPTS4'
94.'opptPTS5'
95.'opptPTS6'
96.'opptPTS7'
97.'opptPTS8'
98.'opptTREB%'
99.'opptASST%'
100.'opptTS%'
101.'opptEFG%'
102.'opptOREB%'
103.'opptDREB%'
104.'opptTO%'
105.'opptSTL%'
106.'opptBLK%'
107.'opptBLKR'
108.'opptPPS'
109.'opptFIC'

110.'opptFIC40'
111.'opptOrtg'
112.'opptDrtg'
113.'opptEDiff'
114.'opptPlay%'
115.'opptAR'
116.'opptAST/TO'
117.'opptSTL/TO'
118.'poss' - Συνολικός αριθμός κατοχών και για τις δύο ομάδες
119.'pace' - Ρυθμός του παιχνιδιού αναλογικά με τη διάρκεια του αγώνα

Εάν υποθέσουμε ότι θα χρησιμοποιήσουμε αυτά τα χαρακτηριστικά για να προβλέψουμε έναν αγώνα NBA, τότε θα αναμένουμε ο αλγόριθμος μηχανικής μάθησης να μας προβλέπει το τελικό αποτέλεσμα με 100% accuracy-ορθότητα. Αυτό συμβαίνει, καθώς για τον αγώνα που επιθυμούμε να προβλέψουμε, έχουμε ήδη στα δεδομένα πληροφορίες που προδίδουν το τελικό αποτέλεσμα, όπως για παράδειγμα οι πόντοι ή οι assists, τα rebounds κλπ. Συνεπώς πρέπει να ακολουθήσουμε μια άλλη στρατηγική, κατα την οποία θα μεταφέρουμε στον αγώνα που θέλουμε να προβλέψουμε, την απαραίτητη πληροφορία από τους προηγούμενους αγώνες για κάθε ομάδα και για κάθε σεζόν. Πρωτού, όμως, φτάσουμε εκεί, θα πρέπει να επεργαστούμε κατάλληλα τα δεδομένα μας, να πραγματοποιήσουμε δηλαδή προ-επεξεργασία δεδομένων.

4.2 Προ-επεργασία δεδομένων και μηχανική χαρακτηριστικών (Data Prepossessing and Feature Engineering)

4.2.1 Τι είναι Προ-επεργασία δεδομένων και η μηχανική χαρακτηριστικών

Η προεπεξεργασία δεδομένων, ένα συστατικό της προετοιμασίας δεδομένων και περιγράφει κάθε τύπο επεξεργασίας που εκτελείται σε ακατέργαστα δεδομένα για την προετοιμασία τους για μια άλλη διαδικασία επεξεργασίας δεδομένων. Αποτελούσε παραδοσιακά ένα σημαντικό προκαταρκτικό βήμα για τη διαδικασία εξόρυξης δεδομένων. Πιο πρόσφατα, οι τεχνικές προεπεξεργασίας δεδομένων έχουν προσαρμοστεί για την εκπαίδευση μοντέλων μηχανικής εκμάθησης και μοντέλων τεχνητής νοημοσύνης και για τη διεξαγωγή συμπερασμάτων σε σχέση με αυτά.

Η προεπεξεργασία δεδομένων μετατρέπει τα δεδομένα σε μια μορφή που υφίσταται πιο εύκολη και αποτελεσματική επεξεργασία στην εξόρυξη δεδομένων, τη μηχανική μάθηση και άλλες εργασίες της επιστήμης των δεδομένων. Οι τεχνικές χρησιμοποιούνται γενικά στα πρώτα στάδια της μηχανικής μάθησης και της ανάπτυξης της τεχνητής νοημοσύνης για την εξασφάλιση ορθότερων αποτελεσμάτων.

Υπάρχουν πολλά διαφορετικά εργαλεία και μέθοδοι που χρησιμοποιούνται για την προεπεξεργασία δεδομένων, συμπεριλαμβανομένων των εξής:

- Δειγματοληψία, η οποία επιλέγει ένα αντιπροσωπευτικό υποσύνολο από έναν μεγάλο πληθυσμό δεδομένων.

- Μετασχηματισμός, ο οποίος χειρίζεται τα ακατέργαστα δεδομένα για να παράγει μια ενιαία είσοδο.
- Απομάκρυνση θορύβου, που αφαιρεί το θόρυβο από τα δεδομένα.
- Καταλογισμός (imputation), ο οποίος συνθέτει στατιστικά σχετικά δεδομένα για τιμές που λείπουν.
- Κανονικοποίηση, η οποία οργανώνει δεδομένα για πιο αποτελεσματική πρόσβαση και
- Εξαγωγή χαρακτηριστικών, η οποία εξάγει ένα σχετικό υποσύνολο χαρακτηριστικών που είναι σημαντικό σε ένα συγκεκριμένο πλαίσιο.

Αυτά τα εργαλεία και μέθοδοι μπορούν να χρησιμοποιηθούν σε μια ποικιλία πηγών δεδομένων, συμπεριλαμβανομένων των δεδομένων που είναι αποθηκευμένα σε αρχεία ή βάσεις δεδομένων και δεδομένων ροής.

Γιατί είναι σημαντική η προεπεξεργασία δεδομένων

Σχεδόν κάθε τύπος ανάλυσης δεδομένων, επιστήμης δεδομένων ή ανάπτυξης τεχνητής νοημοσύνης απαιτεί κάποιο είδος προεπεξεργασίας δεδομένων για την παροχή αξιόπιστων, ορθών και ισχυρών αποτελεσμάτων για εταιρικές εφαρμογές.

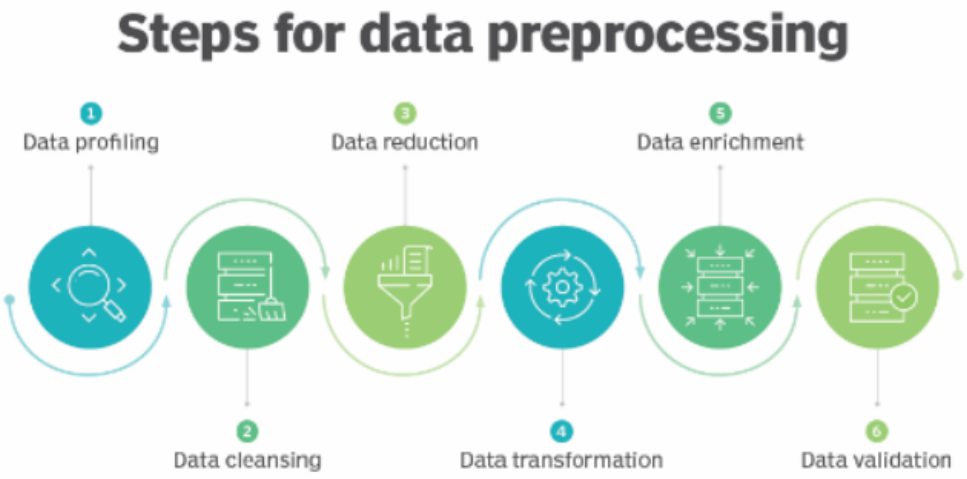
Τα δεδομένα του πραγματικού κόσμου είναι ακατάστατα και συχνά δημιουργούνται, επεξεργάζονται και αποθηκεύονται από διάφορους ανθρώπους, επιχειρηματικές διαδικασίες και εφαρμογές. Ως αποτέλεσμα, από ένα σύνολο δεδομένων μπορεί να λείπουν μεμονωμένα πεδία, να περιέχονται σφάλματα μη αυτόματης εισαγωγής ή να υπάρχουν διπλότυπα δεδομένα ή διαφορετικά ονόματα για να περιγραφεί το ίδιο πράγμα. Οι άνθρωποι μπορούν συχνά να εντοπίσουν και να διορθώσουν αυτά τα προβλήματα στα δεδομένα που χρησιμοποιούν στην επιχειρηματική τους δραστηριότητα, αλλά τα δεδομένα που χρησιμοποιούνται για την εκπαίδευση αλγορίθμων μηχανικής μάθησης ή βαθιάς μάθησης πρέπει να προεπεξεργάζονται αυτόματα.

Οι αλγόριθμοι μηχανικής μάθησης και βαθιάς μάθησης λειτουργούν καλύτερα όταν τα δεδομένα παρουσιάζονται σε μορφή που τονίζει τις σχετικές πτυχές που απαιτούνται για την επίλυση ενός προβλήματος. Οι πρακτικές μηχανικής χαρακτηριστικών που περιλαμβάνουν διαμάχη δεδομένων, μετασχηματισμό δεδομένων, μείωση δεδομένων, επιλογή χαρακτηριστικών και κλιμάκωση χαρακτηριστικών βοηθούν στην αναδιάρθρωση των πρωτογενών δεδομένων σε μορφή κατάλληλη για συγκεκριμένους τύπους αλγορίθμων. Αυτό μπορεί να μειώσει σημαντικά την επεξεργαστική ισχύ και τον χρόνο που απαιτείται για την εκπαίδευση ενός νέου αλγόριθμου μηχανικής μάθησης ή τεχνητής νοημοσύνης ή για την εκτέλεση συμπερασμάτων εναντίον του.

Μια αρχή που πρέπει να τηρείται κατά την προεπεξεργασία δεδομένων: η πιθανότητα επανακωδικοποίησης μεροληψίας στο σύνολο δεδομένων. Ο εντοπισμός και η διόρθωση της προκατάληψης είναι ζωτικής σημασίας για εφαρμογές που βοηθούν στη λήψη αποφάσεων που επηρεάζουν τους ανθρώπους, όπως οι εγκρίσεις δανείων. Παρόλο που οι επιστήμονες δεδομένων μπορεί σκόπιμα να αγνοούν μεταβλητές όπως το φύλο, η φυλή ή η θρησκεία,

αυτά τα χαρακτηριστικά μπορεί να συσχετίζονται με άλλες μεταβλητές όπως οι ταχυδρομικοί κώδικες ή τα σχολεία που φοιτούσαν, δημιουργώντας μεροληπτικά αποτελέσματα.

Τα περισσότερα σύγχρονα πακέτα και υπηρεσίες επιστήμης δεδομένων περιλαμβάνουν πλέον διάφορες βιβλιοθήκες προεπεξεργασίας που βοηθούν στην αυτοματοποίηση πολλών από αυτές τις εργασίες. Τα βασικά βήματα για την προ-επεξεργασία δεδομένων είναι με χρονολογική σειρά η δημιουργία προφίλ στα δεδομένα, ο καθαρισμός τους, η μείωσή τους, ο μετασχηματισμός τους, ο εμπλουτισμός τους και τέλος η επικύρωσή τους. Η σειρά αυτή αποτυπώνεται στην παρακάτω εικόνα :



Σχήμα 4.1: *Τα βήματα της προ-επεξεργασίας δεδομένων*

Αρχικά, οι διαστάσεις των δεδομένων μας, αποτελλούνται από 44284 αγώνες και 119 χαρακτηριστικά. Κατασκευάζουμε ένα νέο χαρακτηριστικό "seasonID", με πιθανές τιμές ¨2012-2013¨,¨2013-2014¨,¨2014-2015¨,¨2015-2016¨,¨2016-2017¨ και ¨2017-2018¨. Για την κατασκευή αυτού του χαρακτηριστικού ανακτήσαμε από το διαδίκτυο τις ημερομηνίες έναρξης και λήξης της εκάστοτε σεζόν και ¨τεμαχίσαμε¨ αναλόγως τα δεδομένα μας, σύμφωνα με το χαρακτηριστικό “gmDate”.

Παρατηρήσαμε, ακόμη, ότι ανά 3 εγγραφές δεδομένων, μεταφερόταν η ίδια ακριβώς πληροφορία, με μόνη διαφορά τα ονόματα και τα επίθετα των διαιτητών κάθε αγώνα. Με λίγα λόγια, κάθε αγώνας είχε 3 αντίγραφα του εαυτού του με διαφορετικούς διαιτητές. Αυτό συμβαίνει, επειδή στην καλαθοσφαίριση γενικότερα και ειδικά στο NBA, υπάρχουν 3 διαιτητές που διευθύνουν τον αγώνα και τα δεδομένα μας ακολουθούσαν αυτή τη μορφή. Αρχικά υπήρξε η ιδέα να αποθηκεύσουμε την τριάδα των διαιτητών σε ένα χαρακτηριστικό και να προβλέψουμε αν μπορεί να επηρεάσει το αποτέλεσμα ενός αγώνα NBA. Τελικά, όμως, επειδή οι διατητές δεν έμεναν καθολικά στην εξαετία και τα δεδομένα μας δεν φάνηκε να επαρκούν για το σκοπό αυτό, αφαιρέσαμε τους διαιτητές, υποτριπλασσιάζοντας έτσι τον όγκο των δεδομένων μας, μένοντας με 14758 αγώνες.

Παρατηρήθηκε ακόμη μια συμμετρία μεταξύ των δεδομένων ανά δύο εγγραφές. Η συμμετρία αυτή αποκάλυψε ότι κάθε εγγραφή περιελάμβανε τόσο την πληροφορία της γηπεδούχου ομάδας, όσο και τις φιλοξενούμενης. Τι πρόβλημα όμως δημιουργεί αυτό το γεγονός. Ας αναλογιστούμε ένα αγώνα Χ, μια εγγραφή στα δεδομένα μας, με τα εξής χαρακτηριστικά :

- Ημερομηνία : Z
- teamAbbr (πρώτη ομάδα στα δεδομένα μας) : DAL
- opptAbbr (δεύτερη ομάδα στα δεδομένα μας) : MIL
- teamLoc (τοποθεσία πρώτης ομάδας στα δεδομένα μας) : εντός έδρας
- Σκορ : 95-90

Και έναν αγώνα Y, μια εγγραφή στα δεδομένα μας, με τα εξής χαρακτηριστικά :

- Ημερομηνία : Z
- teamAbbr (πρώτη ομάδα στα δεδομένα μας) : MIL
- opptAbbr (δεύτερη ομάδα στα δεδομένα μας) : DAL
- teamLoc (τοποθεσία πρώτης ομάδας στα δεδομένα μας) : εκτός έδρας
- Σκορ : 90-95

Αντιλαμβανόμαστε ότι οι δύο αυτές εγγραφές αναφέρονται στον ίδιο αγώνα από άλλη οπτική, συνεπώς η μία εκ των δύο πρέπει να αφαιρεθεί. Πλέον τα δεδομένα μας υποδιπλασιάζονται, κρατώντας τις εγγραφές εκείνες στις οποίες η μεταβλητή “team” αφορά τη γηπεδούχο ομάδα και καταλήγουμε με 7379 αγώνες. Από αυτό το σημείο και έπειτα δεν μιλάμε πλεόν για 1η και 2η ομάδα σε κάθε εγγραφή των δεδομένων, αλλά πλεόν μπορούμε να χρησιμοποιούμε την ορολογία ¨γηπεδούχος¨ ή ¨φιλοξενούμενη¨. Αυτό είναι το πραγματικό πλήθος αγώνων των δεδομένων μας, με κατανομή ανά σεζόν 1230 αγώνες, εκτός από το 2012-2013 που έγιναν 1229 αγώνες.

Τα δεδμένα μας αφορούν μόνο αγώνες κανονικής διάρκειας (regular season) και όχι αγώνες play-off και αφορούν κάθε μία από τις 30 ομάδες που ανήκουν στον θεσμό του NBA (βλέπε επόμενο σχήμα).

Στα δεδομένα μας δεν υπήρχε καμία απουσιάζουσα τιμή.

Μια ακόμη αλλαγή που πραγματοποιήθηκε κατά τη διαδικασία της προ-επεξεργασίας των δεδομένων είναι η αλλαγή του τύπου των στηλών "teamRslt" και "opptRslt" από αλφαριθμητικά σε δυαδικά (0 ή 1), για διευκόλυνση κατά την επεξεργασία τους από τα μοντέλα μηχανικής μάθησης.

Εφόσον εξηγήσαμε την αναγκαιότητα κατασκευής χαρακτηριστικών που δεν μεταφέρουν στον αγώνα που θέλουμε να προβλέψουμε, την πληροφορία των χαρακτηριστικών του ίδιου αγώνα, προχωράμε στην κατασκευή νέων συνόλων δεδομένων που θα ενισχύσουν το επιθυμητό accuracy. Με το σύνολο δεδομένων το οποίο θα καταφέρει να αποδόσει το μέγιστο στις default παραμέτρους των μοντέλων μας, με εκείνο θα συνεχίσουμε και την ανάλυσή μας.

Codes:	ATL	Atlanta Hawks
	BKN	Brooklyn Nets
	BOS	Boston Celtics
	CHA	Charlotte Hornets
	CHI	Chicago Bulls
	CLE	Cleveland Cavaliers
	DAL	Dallas Mavericks
	DEN	Denver Nuggets
	DET	Detroit Pistons
	GS	Golden State Warriors
	HOU	Houston Rockets
	IND	Indiana Pacers
	LAC	Los Angeles Clippers
	LAL	Los Angeles Lakers
	MEM	Memphis Grizzlies
	MIA	Miami Heat
	MIL	Milwaukee Bucks
	MIN	Minnesota Timberwolves
	NO	New Orleans Pelicans
	NY	New York Knicks
	OKC	Oklahoma City Thunder
	ORL	Orlando Magic
	PHI	Philadelphia 76ers
	PHO	Phoenix Suns
	POR	Portland Trail Blazers
	SA	San Antonio Spurs
	SAC	Sacramento Kings
	TOR	Toronto Raptors
	UTA	Utah Jazz
	WAS	Washington Wizards

Σχήμα 4.2: *Οι 30 ομάδες του NBA*

4.2.2 Λογική κατασκευής επιπρόσθετων συνόλων δεδομένων και ποια είναι αυτά

Λογική Dummy συνόλου δεδομένων

Το σύνολο δεδομένων "Dummy" περιλαμβάνει τη λογική του να μεταφέρουμε στον τωρινό αγώνα την πληροφορία όλων των προηγούμενων αγώνων ανά ομάδα και ανά σεζόν. Για κάθε νέα σεζόν μηδενίζουμε τις τιμές των χαρακτηριστικών της στην πρώτη αγωνιστική. Για τη μεταφορά αυτής της πληροφορίας χρησιμοποιούμε τους αθροιστικούς μέσους όρους για κάθε υπάρχον χαρακτηριστικό, πλην των μη αριθμητικών ή των ordinal χαρακτηριστικών.

Πλεον αυξάνουμε το σύνολο δεδομένων μας στα 207 χαρακτηριστικά.

Εισάγαμε τις έννοιες των μεταβλητών “teamHome” και “opptAway” που προβάλλουν τη συμπεριφορά μιας ομάδας, αποκλειστικά όταν εκείνη είναι γηπεδούχος ή φιλοξενούμενη και όχι μόνο τη συνολική της εικόνα.

Πλέον κάθε χαρακτηριστικό έχει την κατάληξη “_cum_mean” υποδηλώνοντας την εκδοχή του με τον αντίστοιχο αθροιστικό μέσο όρο μέχρι τον τωρινό εκάστοτε αγώνα.

Για να κατανοήσουμε καλύτερα αυτά τα νέα χαρακτηριστικά, θα μπορούσαμε να φαντα-

στούμε το εξής παράδειγμα: Έστω το υποθετικό σενάριο όπου μια ομάδα DAL όπου στις πέντε πρώτες αγωνιστικές της έχει τα εξής rebounds

- 1η αγωνιστική - 30
- 2η αγωνιστική - 10
- 3η αγωνιστική - 30
- 4η αγωνιστική - 10
- 5η αγωνιστική - 30

Άρα έχει 22 rebounds αθροιστικό μέσο όρο στον 6ο αγώνα (από τους προηγούμενους 5). Έστω ότι έχουμε τη γνώση ότι μια ομάδα με περισσότερα από 20 rebounds κατά 80% κερδίζει τον αγώνα.

Αν τώρα υποθέσουμε ότι οι περιττοί αγώνες αφορούν τα rebounds της ίδιας ομάδας όταν εκείνη είναι γηπεδούχος, ενώ οι ζυγοί τα rebounds της ίδιας ομάδας όταν εκείνη είναι φιλοξενούμενη, τότε συμπεραίνουμε ότι ο αθροιστικός μέσος όρος της ομάδας στον επόμενο αγώνα της ως γηπεδούχος ομάδα, θα είναι 30 rebounds και θα υποδηλώνει ότι κατά μεγάλη πιθανότητα η ομάδα αυτή θα κερδίσει και αντίστοιχα όταν είναι φιλοξενούμενη θα είναι 10 rebounds και θα υποδηλώνει ότι πιθανότατα θα ηττηθεί.

Συνεπώς αποδείξαμε την αναγκαιότητα διάκρισης των χαρακτηριστικών σε “teamHome” και “opptAway”, καθώς αρχικά θα λέγαμε ότι η “DAL” θα κέρδιζε πιθανότατα τον επόμενο αγώνα, ενώ πλέον θα πρέπει να αναρρωτηθούμε πρώτα αν αυτή η ομάδα θα είναι γηπεδούχος ή φιλοξενούμενη και να ανατρέξουμε στη συμπεριφορά της ανάλογα με το που παίζει, για να αποφασίσουμε για την έκβαση του αγώνα.

Λογική LastN συνόλου δεδομένων

Το σύνολο δεδομένων "Last_N" βασίζεται στη λογική της αποτύπωσης της φόρμας μιας ομάδας. Το N στα πειράματά μας αποτελεί μια μεταβλητή με εύρος τιμών να ισούται με 3,5,7,9 και 10. Κάθε νέο χαρακτηριστικό πλέον, μη αλφαριθμητικό, θα έχει και την αντίστοιχη έκδοση “Last_N”, υποδηλώνοντας τη φόρμα της εκάστοτε ομάδας στους τελευταίους N αγώνες.

Για παράδειγμα ας σκεφτούμε τους πόντους που σκοράρει μια ομάδα στους τελευταίους 20 αγώνες. Έστω ότι ανά αγώνα βάζει 100 πόντους και έστω ότι μια ομάδα που σκοράρει 100 πόντους κερδίζει με μεγάλη πιθανότητα τον επόμενο αγώνα. Ο αριθμός 100 όμως, προέρχεται από τον αθροιστικό μέσο όρο των τελευταίων 20 αγώνων, πράγμα που μπορεί να ευνοεί ή και να αδικεί την εικόνα μιας ομάδας.

Για παράδειγμα αν μια ομάδα στους πρώτους 10 αγώνες της έβαζε 50 πόντους και στους τελευταίους 10 έβαζε 150, τότε δείχνει να έχει καλή φόρμα και με μεγαλύτερη πιθανότητα σε σχέση με πριν μπορούμε να αποφασίσουμε για την εικόνα της στον επόμενο αγώνα (την αδικούμε, είναι καλύτερη από 100 πόντους μέσο όρο, είναι στους 150 σύμφωνα με τη φόρμα της).

Από την άλλη, αν βρισκόμαστε στο αντίστροφο σενάριο, δηλαδή όταν στους 10 πιο παρελθοντικούς αγώνες της ίδιας ομάδας, εκείνη σκόραρε 150 πόντους και στους 10 πιο πρόσφατους σκόραρε 50, τότε μάλλον θα χάσει τον επόμενο αγώνα της, γιατί βρίσκεται σε πολλή

κακή φόρμα και τελικά οι 100 πόντοι μέσος όρος ευνοούσαν την πραγματικά κακή εικόνα της.

Λογική DropN συνόλου δεδομένων

Εδώ μπαίνει η έννοια του χρόνου και πόσο αυτός μπορεί να επηρεάσει τα δεδομένα μας. Αρχικά ας σκεφτούμε την πληροφορία που έχει ένα σύνολο δεδομένων με "Last_10" χαρακτηριστικά. Εφόσον κάθε νέα σεζόν κοιτάμε τη φόρμα της ομάδας στους 10 προηγούμενους αγώνες, όταν θα είμαστε στον 5ο για παράδειγμα αγώνα, θα ήταν λάθος να γεμίσουμε την αντίστοιχη “Last_10” κολώνα, αφού δεν έχουμε αρκετούς αγώνες για να ξέρουμε τη φόρμα της. Συνεπώς μηδενίζουμε τις 10 πρώτες αγωνιστικές. Για να μην εκπαιδεύσουμε τα δεδομένα μας σε μηδενική πληροφορία, τα αφαιρούμε κατασκευάζοντας το “Drop_10” σύνολο δεδομένων.

Αν τώρα θέλουμε να μεταφέρουμε την ανάλυσή μας ανάλογα με τον χρόνο, τότε έχουμε τη δυνατότητα να αφαιρούμε από κάθε σύνολο δεδομένων αγώνες, είτε από την αρχή, είτε από το τέλος του. Γιατί όμως να θέλουμε να πετάξουμε δεδομένα.

Στην καλαθοσφαίριση, στο NBA πιο συγκεκριμένα, αλλά και στα περισσότερα αθλήματα, κατά την έναρξη μιας σεζόν, οι ομάδες δεν έχουν βρει τα πατήματά τους, δηλαδή η αρχική τους εικόνα δεν ανταποκρίνεται πάντοτε στην τελική. Εξάλλου αν συνέβαινε αυτό, τότε μόνο έχοντας δεδομένα απο την έναρξη ενός πρωταθληματος, θα μπορούσαμε να προβλέψουμε και τη λήξη του, πράγμα που δεν ισχύει απόλυτα.

Αυτό πρακτικά σημαίνει ότι οι αρχικοί αγώνες φαίνεται να είναι πιο απρόβλεπτοι, γιατί για παράδειγμα, η ομάδα δεν έχει ολοκληρώσει όλες τις φετινές της μεταγραφές.

Θα αποφασίσουμε αν με την αφαίρεση δεδομένων είτε από την αρχή, είτε από το τέλος, βελτιώνουμε ή χειροτερεύουμε τα αποτελέσματα, για να δούμε εαν θα κρατήσουμε αυτού του είδους τα σύνολα δεδομένων.

Λογική Plus27 συνόλου δεδομένων

Πέραν της κατασκευής νέων χαρακτηριστικών από τα ήδη υπάρχοντα (_cum_mean και last_N), επιλέγουμε να κατασκευάσουμε και 27 επιπλέον δικά μας χαρακτηριστικά, που θεωρούμε ότι θα βελτιώσουν την απόδοση των αλγορίθμων της μηχανικής μάθησης. Έτσι κατασκευάζουμε το “_plus27” σύνολο δεδομένων.

Τα χαρακτηριστικά αυτά είναι τα εξής:

1. "teamAvgPtDiff" = Μέσος όρος διαφοράς πόντων της γηπεδούχου ομάδας μέχρι και πριν τον τωρινό αγώνα, ανά σεζόν

2. "opptAvgPtDiff" = Μέσος όρος διαφοράς πόντων της φιλοξενούμενης ομάδας μέχρι και πριν τον τωρινό αγώνα, ανά σεζόν

3. "teamHomeAvgPtDiff" = Μέσος όρος διαφοράς πόντων της γηπεδούχου ομάδας όταν παίζει ως γηπεδούχος μέχρι και πριν τον τωρινό αγώνα, ανά σεζόν

4. "opptAwayAvgPtDiff" = Μέσος όρος διαφοράς πόντων της φιλοξενούμενης ομάδας όταν παίζει ως φιλοξενούμενη μέχρι και πριν τον τωρινό αγώνα, ανά σεζόν

5. "teamLastNWin%" = Ποσοστό νικών της γηπεδούχου ομάδας στους τελευταίους N αγώνες, μέχρι και πριν τον τωρινό αγώνα, ανά σεζόν

6. "teamHomeLastNWin%" = Ποσοστό νικών της γηπεδούχου ομάδας όταν παίζει ως γηπεδούχος στους τελευταίους N αγώνες, μέχρι και πριν τον τωρινό αγώνα, ανά σεζόν

7. "opptLastNWin%" = Ποσοστό νικών της φιλοξενούμενης ομάδας στους τελευταίους N αγώνες, μέχρι και πριν τον τωρινό αγώνα, ανά σεζόν

8. "opptAwayLastNWin%" = Ποσοστό νικών της φιλοξενούμενης ομάδας όταν παίζει ως φιλοξενούμενη στους τελευταίους N αγώνες, μέχρι και πριν τον τωρινό αγώνα, ανά σεζόν

9. "teamHomeAvgPtDiffLastN" = Μέσος όρος διαφοράς πόντων της γηπεδούχου ομάδας όταν παίζει ως γηπεδούχος στους τελευταίους N αγώνες, μέχρι και πριν τον τωρινό αγώνα, ανά σεζόν

10. "teamAvgPtDiffLastN" = Μέσος όρος διαφοράς πόντων της γηπεδούχου ομάδας στους τελευταίους N αγώνες, μέχρι και πριν τον τωρινό αγώνα, ανά σεζόν

11. "opptAwayAvgPtDiffLastN" = Μέσος όρος διαφοράς πόντων της φιλοξενούμενης ομάδας όταν παίζει ως φιλοξενούμενη στους τελευταίους N αγώνες, μέχρι και πριν τον τωρινό αγώνα, ανά σεζόν

12. "opptAvgPtDiffLastN" = Μέσος όρος διαφοράς πόντων της φιλοξενούμενης ομάδας στους τελευταίους N αγώνες, μέχρι και πριν τον τωρινό αγώνα, ανά σεζόν

13. "teamB2BWins" = Απανωτές νίκες της γηπεδούχου ομάδας μέχρι και πριν τον τωρινό αγώνα, ανά σεζόν

14. "teamHomeB2BWins" = Απανωτές νίκες της γηπεδούχου ομάδας όταν παίζει ως γηπεδούχος, μέχρι και πριν τον τωρινό αγώνα, ανά σεζόν

15. "opptAwayB2BWins" = Απανωτές νίκες της φιλοξενούμενης ομάδας όταν παίζει ως φιλοξενούμενη, μέχρι και πριν τον τωρινό αγώνα, ανά σεζόν

16. "altitude" = Υψόμετρο Γηπέδου

17. "teamHomeWin%" = Ποσοστό νικών της γηπεδούχου ομάδας όταν παίζει ως γηπεδούχος, μέχρι και πριν τον τωρινό αγώνα, ανά σεζόν

18. "opptAwayWin%" = Ποσοστό νικών της φιλοξενούμενης ομάδας όταν παίζει ως φιλοξενούμενη, μέχρι και πριν τον τωρινό αγώνα, ανά σεζόν

19. "teamB2BGames" = Απανωτά παιχνίδια γηπεδούχου ομάδας (δυαδική μεταβλητή, μετιμές 1 εφόσον η ομάδα έπαιζε και την προηγούμενη μέρα, αλλιώς 0)

20. "opptB2BGames" = Απανωτά παιχνίδια φιλοξενούμενης ομάδας (δυαδική μεταβλητή, μετιμές 1 εφόσον η ομάδα έπαιζε και την προηγούμενη μέρα, αλλιώς 0)

21. "teamWin%" = Ποσοστό νικών της γηπεδούχου ομάδας, μέχρι και πριν τον τωρινό αγώνα, ανά σεζόν

22. "opptWin%" = Ποσοστό νικών της φιλοξενούμενης ομάδας, μέχρι και πριν τον τωρινό αγώνα, ανά σεζόν

23. "teamRestingDays" = Μέρες ξεκούρασης της γηπεδούχου ομάδας, μέχρι και πριν τον τωρινό αγώνα, ανά σεζόν

24. "opptB2BWins" = Απανωτές νίκες της φιλοξενούμενης ομάδας μέχρι και πριν τον τωρινό αγώνα, ανά σεζόν

25. "opptRestingDays" = Μέρες ξεκούρασης της φιλοξενούμενης ομάδας, μέχρι και πριν τον τωρινό αγώνα, ανά σεζόν

26. "teamPriorGames" = Αριθμός αγώνων που έχει συμμετάσχει η γηπεδούχος ομάδα μέχρι και πριν τον τωρινό αγώνα, ανά σεζόν

27. "opptPriorGames" = Αριθμός αγώνων που έχει συμμετάσχει η φιλοξενούμενη ομάδα μέχρι και πριν τον τωρινό αγώνα, ανά σεζόν

Η διαχείρηση των συνόλων δεδομένων και των χαρακτηριστικών, καθώς και όλη η προγραμματιστική υλοποίηση της εργασίας έγινε με τη χρήση της γλώσσας προγραμματισμού Python, χρησιμοποιώντας τις κατάλληλες βιβλιοθήκες. Για την κατασκευή του χαρακτηριστικού “Altitude”, ανασύραμε την πληροφορία για το υψόμετρο του εκάστοτε γηπέδου ξεχωριστά από το διαδίκτυο. Στο σημείο αυτό ολοκληρώνεται η ενότητα της περιγραφής του συνόλου δεδομένων και των διαδικασιών προεπεξεργασίας τους και μηχανικής των χαρακτηριστικών και μεταβαίνουμε στην επόμενη ενότητα που αφορά τα πειραματικά αποτελέσματα, καθώς και τη μεθοδολογία που ακολουθήθηκε για την εύρεση του βέλτιστου μοντέλου μηχανικής μάθησης. Να σημειωθεί ακόμη ότι με την εισαγωγή αυτών των χαρακτηριστικών, αλλά και τη χρήση του Dummy και Last_7 συνόλου δεδομένων ταυτόχρονα, καταλήγουμε σε σύνολο δεδομένων με 431 χαρακτηριστικά.

Κεφάλαιο 5

Πειραματικά Αποτελέσματα και Μεθολογία εύρεσής βέλτιστου αποτελέσματος

Σε αυτό το κεφάλαιο επιθυμούμε να βρούμε εκείνο τον αλγόριθμο μηχανικής μάθησης, ο οποίος στις βέλτιστες υπερπαραμέτρους του, επιτυγχάνει τη μεγαλύτερη τιμή της μετρικής accuracy, όσον αφορά την ορθότητα του στην εύρεση του αποτελέσματος ενός αγώνα NBA. Στην πλειοψηφία της, η σχετική βιβλιογραφία επιτυγχάνει 65% ορθότητα στην πρόβλεψη του αποτελέσματος ενός αγώνα NBA και θεωρούμε επιτυχημένη την παρούσα εργασία, σε περίπτωση αναπαραγωγής αυτών των αριθμών, δεδομένου ότι δοκιμάζουμε τη δική μας στρατηγική σε διαφορετικό σύνολο δεδομένων ή και την υπέρβασή τους.

5.1 Μεθοδολογία Εύρεσης βέλτιστων πειραματικών αποτελεσμάτων

Η στρατηγική που ακολουθούμε για να καταλήξουμε στο βέλτιστο μοντέλο μηχανικής μάθησης είναι η εξής:

1. Δοκιμάζουμε στο "Dummy" σύνολο δεδομένων, με τη χρήση του εργαλείου pycaret, τη βέλτιστη εκδοχή της συνάρτησης setup(), εκείνης της εκδοχής που καταλήγει σε μεγαλύτερες τιμές ορθότητας. Η setup() συνάρτηση ορίζει τις παραμέτρους εκείνες με τις οποίες θα εκπαιδευτούν όλοι οι ταξινομητές μας. Για περισσότερες πληροφορίες σχετικά με την τεκμηρίωση αυτής της βιβλιοθήκης της python (pycaret), μπορεί κανείς να ανατρέξει στον εξής σύνδεσμο (https://pycaret.readthedocs.io/en/stable/) και συγκεκριμένα να επιλέξει την καρτέλα της δυαδικής ταξινόμησης. Για πληροφορίες σχετικές με τον τρόπο που γίνονται train-test-validation split τα δεδομένα μας αλλα και για τη μέθοδο cross validation που ακολουθήθηκε μπορεί κανείς να ανατρέξει στον σύνδεσμο (https://www.datasource.ai/uploads/624e8836466a40923b64b901b5050c0f.html). Η βιβλιοθήκη αυτή είναι κατάλληλη για προβλήματα ταξινόμησης, παλινδόμησης, εντοπισμού ανωμαλιών, συσταδοποίησης, εύρεσης κανόνων συσχέτησης κ.ά., καλώντας άλλες βιβλιοθήκες για την εκτέλεση των μοντέλων μηχανικής μάθησης. Είναι πρακτική για τον προγραμματιστή, καθώς χωρίς την απαίτηση συγγραφής κώδικα, καλεί τα μοντέλα που επιθυμεί εκείνος και συνοψίζει τα αποτελέσματά τους, πραγματοποιώντας ακόμα και βελτιστοποίηση υπερπαραμέτρων. Το μόνο που χρειάζεται κανείς για να μπορέσει να εκμεταλλευτεί σωστά αυτό το εργαλείο είναι να διαβάσει πολύ καλά την τεκμηρίωσή του, ώστε να γνωρίζει επ ακριβώς τις αλλαγές που

επιφέρουν οι μετατροπές των παραμέτρων που εκείνο χρησιμοποιεί. Στην παρούσα διπλωματική εργασία οι αλγόριθμοι Μηχανικής Μάθησης που θα εφαρμοστούν είναι οι Logistic Regression, K Neighbors Classifier, Naive Bayes, Decision Tree Classifier, SVM - Linear Kernel, SVM - Radial Kernel, Gaussian Process Classifier, MLP Classifier, Ridge Classifier, Random Forest Classifier, Quadratic Discriminant Analysis, Ada Boost Classifier, Gradient Boosting Classifier, Linear Discriminant Analysis, Extra Trees Classifier, Extreme Gradient Boosting, Light Gradient Boosting Machine και ο CatBoost Classifier. Το βέλτιστο μοντέλο καθορίζεται από το βέλτιστο accuracy στο σύνολο επικύρωσης, για τον top 1 αλγόριθμο, στις default παραμέτρους του.

2. Μόλις ανακαλύψουμε τη βέλτιστη setup() συνάρτηση, τότε αναζητούμε το βέλτιστο accuracy, επανεξετάζοντας τα αποτελέσματα, κατόπιν χρήσης κάθε πιθανού dataset που είναι παράγωγο του αρχικού.

3. Μόλις βρεθεί και το dataset που μας δίνει το βέλτιστο accuracy, τότε εφαρμόζουμε hyperparameter tuning στο σύνολο επικύρωσης και ελέγχουμε το performance των αλγορίθμων μας στο test set. Εάν αυτό προσεγγίζει το 65% ή το ξεπερνάει (παγκόσμια βιβλιογραφία σε αντίστοιχο πρόβλημα), τότε έχουμε πετύχει τον σκοπό μας.

4. Τέλος, εφαρμόζουμε δοκιμές με custom ANN και deep learning models για να ερευνήσουμε αν καταφέρνουμε να ξεπεράσουμε την επίδοση του καλύτερου μοντέλου μηχανικής μάθησης.

5.2 Πειραματικά Αποτελέσματα

5.2.1 Εύρεση βέλτιστων παραμέτρων της setup() συνάρτησης του pycaret

Για την εύρεση των βέλτιστων παραμέτρων της setup() συνάρτησης του pycaret, θα δοκιμάσουμε το “Dummy” σύνολο δεδομένων και θα κρατάμε κάθε φορά ως βέλτιστη, τη δοκιμή εκείνη που θα καταλήξει σε υψηλότερη τιμή ορθότητας για τις default παραμέτρους στο σύνολο επικύρωσης των μοντέλων μηχανικής μάθησης που χρησιμοποιούμε.

1.1. Αρχικά ορίζουμε μόνο την παράμετρο "feature_selection" να είναι Αληθής. Όταν οριστεί σε True, επιλέγεται ένα υποσύνολο χαρακτηριστικών χρησιμοποιώντας έναν συνδυασμό διαφόρων τεχνικών σημαντικότητας μετάθεσης (permutation importance), όπως Τυχαίο Δάσος, Adaboost και Γραμμική συσχέτιση με μεταβλητή στόχο. Το μέγεθος του υποσυνόλου εξαρτάται από την παράμετρο feature_selection_threshold της οποίας η default τιμή είναι στο 0.8. Στο σημείο αυτό ελέγχουμε μόνο τα αποτελέσματα στις default παραμέτρους των μοντέλων και δεν πραγματοποιούμε κάποια βελτιστοποίηση υπερπαραμέτρων. Θα δείξουμε στα αποτελέσματα μας μόνο τους τελικούς πίνακες με τα μοντέλα και τις μετρικές. Σε επόμενο στάδιο που θα εφαρμόσουμε βελτιστοποίηση των υπερπαπαρμέτρων, θα δείξουμε και τις υπερπαράμετρους εκείνες του βέλτιστου μοντέλου και περισσότερες λεπτομέρειες για αυτό.

Τα αποτελέσματα αυτής της 1ης δοκιμής έχουν ως εξής:

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
lr	Logistic Regression	0.6468	0.6834	0.7828	0.6715	0.7227	0.2438	0.2497	0.7560
gbc	Gradient Boosting Classifier	0.6468	0.6822	0.7721	0.6747	0.7200	0.2473	0.2518	1.8960
et	Extra Trees Classifier	0.6425	0.6776	0.7714	0.6708	0.7174	0.2371	0.2417	0.1520
lightgbm	Light Gradient Boosting Machine	0.6425	0.6656	0.7551	0.6757	0.7130	0.2427	0.2455	0.3380
ridge	Ridge Classifier	0.6421	0.0000	0.7683	0.6711	0.7163	0.2373	0.2417	0.0210
lda	Linear Discriminant Analysis	0.6409	0.6683	0.7593	0.6725	0.7131	0.2375	0.2411	0.1670
rf	Random Forest Classifier	0.6404	0.6744	0.7672	0.6698	0.7150	0.2338	0.2382	0.2900
ada	Ada Boost Classifier	0.6315	0.6540	0.7509	0.6658	0.7056	0.2175	0.2206	0.3880
nb	Naive Bayes	0.6044	0.6623	0.5678	0.7056	0.6245	0.2161	0.2241	0.0200
dummy	Dummy Classifier	0.5885	0.5000	1.0000	0.5885	0.7409	0.0000	0.0000	0.0150
svm	SVM - Linear Kernel	0.5828	0.0000	0.7055	0.6694	0.6167	0.1118	0.1392	0.0690
knn	K Neighbors Classifier	0.5823	0.5757	0.6997	0.6308	0.6633	0.1171	0.1185	0.1950
dt	Decision Tree Classifier	0.5807	0.5667	0.6460	0.6433	0.6443	0.1335	0.1337	0.1060
qda	Quadratic Discriminant Analysis	0.5316	0.5116	0.6242	0.5937	0.5858	0.0238	0.0234	0.0420

Πίνακας 5.1: *Αποτελέσματα 1ης δοκιμής, μόνο χρήση παραμέτρου feature_selection*

Παρατηρούμε σε φθίνουσα σειρά τα μοντέλα εκείνα με την υψηλότερη τιμή ορθότητας, συνοδευόμενα και από τις υπόλοιπες μετρικές τους.

Σε αυτή την 1η δοκιμή έχουμε το εξής top 5 μοντέλων μηχανικής μάθησης, ως προς την ορθότητα :

1. Logistic Regression με ορθότητα ίση με 64,68%
2. Gradient Boosting Classifier με ορθότητα ίση με 64,68%
3. Extra Trees Classifier με ορθότητα ίση με 64,25%
4. Light Gradient Boosting Machine με ορθότητα ίση με 64,25%
5. Ridge Classifier με ορθότητα ίση με 64,21%

Να σημειωθεί ότι οι διαφορές μεταξύ των top 5 αλγορίθμων ως προς την τιμή της μετρικής της ορθότητας είναι πολύ μικρές, συνήθως μικρότερες της τάξης του 2%, συνεπώς στόχος της εργασίας δεν είναι τόσο η εξήγηση του γιατί ένα μοντέλο είναι καλύτερο από ένα άλλο, αλλά η εύρεση του καλύτερου. Εξ άλλου, οποιαδήποτε εικασία σε τόσο μικρές διαφορές θα μπορούσε να μην είναι απόλυτα σωστή και δύσκολα αποδεικνύεται. Επιλέγουμε συνεπώς τον δρόμου του "trial and error".

Κρατάμε συνεπώς ότι από την 1η δοκιμή μας έχουμε ως βέλτιστη ορθότητα το 64,68%. Επειδή τα χαρακτηριστικά μας είναι πολλά σε πλήθος, αποφασίζουμε ως σημείο εκκίνησης να υιοθετήσουμε αυτή την αλλαγή, ανεξαρτήτως εάν μειώνεται ή όχι η ορθότητα κατά λίγο, καθώς με την επιλογή χαρακτηριστικών, κερδίζουμε σημαντικά σε υπολογιστική ισχύ. Ας προσπαθήσουμε να ξεπεράσουμε αυτή την ορθότητα.

1.2. Στη συνέχεια ορίζουμε την παράμετρο Normalization να ισούται με Αληθής με την default normalize_method, τον "zscore normalizer". Όταν οριστεί σε True, μετατρέπει τα

αριθμητικά χαρακτηριστικά κλιμακώνοντάς τα σε ένα δεδομένο εύρος. Ο τύπος κλιμάκωσης ορίζεται από την παράμετρο normalize_method.

Τα αποτελέσματα αυτής της 2ης δοκιμής έχουν ως εξής:

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
gbc	Gradient Boosting Classifier	0.6500	0.6812	0.7752	0.6770	0.7226	0.2542	0.2590	1.9370
et	Extra Trees Classifier	0.6455	0.6808	0.7742	0.6729	0.7198	0.2436	0.2484	0.1520
ridge	Ridge Classifier	0.6449	0.0000	0.7724	0.6726	0.7189	0.2429	0.2477	0.0180
lr	Logistic Regression	0.6431	0.6744	0.7672	0.6724	0.7165	0.2402	0.2445	0.3170
lda	Linear Discriminant Analysis	0.6411	0.6682	0.7593	0.6727	0.7131	0.2380	0.2417	0.1750
lightgbm	Light Gradient Boosting Machine	0.6380	0.6728	0.7555	0.6713	0.7105	0.2316	0.2351	0.3430
rf	Random Forest Classifier	0.6321	0.6727	0.7517	0.6661	0.7061	0.2189	0.2221	0.2960
ada	Ada Boost Classifier	0.6311	0.6537	0.7572	0.6636	0.7072	0.2144	0.2180	0.3950
knn	K Neighbors Classifier	0.6154	0.6241	0.7371	0.6536	0.6927	0.1835	0.1861	0.2160
nb	Naive Bayes	0.6060	0.6567	0.5681	0.7084	0.6265	0.2193	0.2276	0.0200
svm	SVM - Linear Kernel	0.5972	0.0000	0.6945	0.6491	0.6682	0.1543	0.1563	0.0420
dummy	Dummy Classifier	0.5885	0.5000	1.0000	0.5885	0.7409	0.0000	0.0000	0.0150
dt	Decision Tree Classifier	0.5772	0.5644	0.6370	0.6419	0.6391	0.1287	0.1288	0.1050
qda	Quadratic Discriminant Analysis	0.4978	0.4966	0.5025	0.5845	0.5258	-0.0069	-0.0082	0.0420

Πίνακας 5.2: *Αποτελέσματα 2ης δοκιμής, χρήση παραμέτρων feature_selection και Normalization*

Σε αυτή την 2η δοκιμή έχουμε το εξής top 5 μοντέλων μηχανικής μάθησης, ως προς την ορθότητα :

1. Gradient Boosting Classifier με ορθότητα ίση με 65%
2. Extra Trees Classifier με ορθότητα ίση με 64,55%
3. Ridge Classifier με ορθότητα ίση με 64,49%
4. Logistic Regression με ορθότητα ίση με 64,31%
5. Linear Discriminant Analysis με ορθότητα ίση με 64,11%

Κρατάμε συνεπώς ότι από την 2η δοκιμή μας έχουμε ως βέλτιστη ορθότητα το 65%. Παρατηρούμε βελτίωση σε σχέση με την προηγούμενη βέλτιστη τιμή της ορθότητας. Συνεπώς υιοθετούμε αυτή την αλλαγή και δοκιμάζουμε την επόμενη. Ας προσπαθήσουμε και πάλι να την ξεπεράσουμε.

1.3. Στη συνέχεια ορίζουμε την παράμετρο Transformation να ισούται με Αληθής. Όταν οριστεί σε True, εφαρμόζει τον μετασχηματισμό ισχύος για να κάνει τα δεδομένα πιο όμοια με Gaussian. Ο τύπος μετασχηματισμού ορίζεται από την παράμετρο transformation_method, στην περίπωσή μας είναι η yeo-johnson.

Τα αποτελέσματα αυτής της 3ης δοκιμής έχουν ως εξής:

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
et	Extra Trees Classifier	0.6453	0.6777	0.7735	0.6727	0.7195	0.2435	0.2483	0.1440
ridge	Ridge Classifier	0.6443	0.0000	0.7693	0.6730	0.7179	0.2424	0.2466	0.0190
lda	Linear Discriminant Analysis	0.6441	0.6695	0.7641	0.6744	0.7164	0.2436	0.2473	0.1740
gbc	Gradient Boosting Classifier	0.6439	0.6812	0.7703	0.6722	0.7178	0.2411	0.2457	1.9440
lr	Logistic Regression	0.6372	0.6691	0.7610	0.6684	0.7115	0.2280	0.2318	0.2660
rf	Random Forest Classifier	0.6362	0.6727	0.7575	0.6682	0.7098	0.2269	0.2307	0.2980
ada	Ada Boost Classifier	0.6317	0.6563	0.7568	0.6643	0.7075	0.2159	0.2194	0.3980
lightgbm	Light Gradient Boosting Machine	0.6307	0.6653	0.7516	0.6646	0.7052	0.2154	0.2188	0.3230
nb	Naive Bayes	0.6294	0.6689	0.6308	0.7080	0.6668	0.2524	0.2548	0.0200
knn	K Neighbors Classifier	0.6172	0.6350	0.7343	0.6559	0.6928	0.1891	0.1915	0.2430
svm	SVM - Linear Kernel	0.5942	0.0000	0.6709	0.6501	0.6595	0.1568	0.1577	0.0530
dummy	Dummy Classifier	0.5885	0.5000	1.0000	0.5885	0.7409	0.0000	0.0000	0.0140
dt	Decision Tree Classifier	0.5770	0.5636	0.6394	0.6409	0.6401	0.1272	0.1273	0.1070
qda	Quadratic Discriminant Analysis	0.5126	0.4943	0.5978	0.5852	0.5854	-0.0122	-0.0131	0.0400

Πίνακας 5.3: *Αποτελέσματα 3ης δοκιμής, χρήση παραμέτρων feature_selection, Normalization και Transformation*

Σε αυτή την 3η δοκιμή έχουμε το εξής top 5 μοντέλων μηχανικής μάθησης, ως προς την ορθότητα :

1. Extra Trees Classifier με ορθότητα ίση με 64,53%
2. Ridge Classifier με ορθότητα ίση με 64,43%
3. Linear Discriminant Analysis με ορθότητα ίση με 64,41%
4. Gradient Boosting Classifier με ορθότητα ίση με 64,39%
5. Logistic Regression με ορθότητα ίση με 63,72%

Κρατάμε συνεπώς ότι από την 3η δοκιμή μας έχουμε ως βέλτιστη ορθότητα το 64,53%. Δεν παρατηρούμε βελτίωση σε σχέση με την προηγούμενη βέλτιστη τιμή της ορθότητας. Συνεπώς δεν υιοθετούμε αυτή την αλλαγή και δοκιμάζουμε την επόμενη. Ας προσπαθήσουμε και πάλι να την ξεπεράσουμε.

1.4. Στη συνέχεια ορίζουμε την παράμετρο ignore_low_variance να ισούται με Αληθής. Όταν οριστεί σε True, όλα τα κατηγορικά χαρακτηριστικά με μηδαμινές αποκλίσεις αφαιρούνται από τα δεδομένα. Η διακύμανση υπολογίζεται χρησιμοποιώντας την αναλογία μοναδικών τιμών προς τον αριθμό των δειγμάτων και την αναλογία της πιο κοινής τιμής προς τη συχνότητα της δεύτερης πιο κοινής τιμής.

Τα αποτελέσματα αυτής της 4ης δοκιμής έχουν ως εξής:

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
et	Extra Trees Classifier	0.6490	0.6777	0.7839	0.6734	0.7243	0.2488	0.2545	0.1470
ridge	Ridge Classifier	0.6451	0.0000	0.7735	0.6726	0.7194	0.2430	0.2478	0.0210
gbc	Gradient Boosting Classifier	0.6449	0.6806	0.7697	0.6735	0.7182	0.2438	0.2483	1.9140
rf	Random Forest Classifier	0.6421	0.6761	0.7669	0.6714	0.7158	0.2379	0.2421	0.2950
lr	Logistic Regression	0.6413	0.6772	0.7645	0.6714	0.7148	0.2367	0.2407	0.2960
lda	Linear Discriminant Analysis	0.6404	0.6711	0.7575	0.6727	0.7124	0.2371	0.2405	0.1610
lightgbm	Light Gradient Boosting Machine	0.6396	0.6714	0.7582	0.6717	0.7121	0.2348	0.2384	0.3190
ada	Ada Boost Classifier	0.6315	0.6542	0.7509	0.6658	0.7056	0.2175	0.2206	0.3980
knn	K Neighbors Classifier	0.6129	0.6261	0.7246	0.6547	0.6877	0.1820	0.1839	0.2230
nb	Naive Bayes	0.6119	0.6620	0.5861	0.7078	0.6371	0.2270	0.2340	0.0190
svm	SVM - Linear Kernel	0.6007	0.0000	0.6737	0.6582	0.6640	0.1705	0.1723	0.0370
dummy	Dummy Classifier	0.5885	0.5000	1.0000	0.5885	0.7409	0.0000	0.0000	0.0150
dt	Decision Tree Classifier	0.5789	0.5658	0.6398	0.6431	0.6412	0.1314	0.1315	0.1020
qda	Quadratic Discriminant Analysis	0.5359	0.5196	0.6113	0.6044	0.6007	0.0398	0.0417	0.0410

Πίνακας 5.4: *Αποτελέσματα 4ης δοκιμής, χρήση παραμέτρων feature_selection, Normalization και ignore_low_variance*

Σε αυτή την 4η δοκιμή έχουμε το εξής top 5 μοντέλων μηχανικής μάθησης, ως προς την ορθότητα :

1. Extra Trees Classifier με ορθότητα ίση με 64,90%
2. Ridge Classifier με ορθότητα ίση με 64,51%
3. Gradient Boosting Classifier με ορθότητα ίση με 64,49%
4. Random Forest Classifier με ορθότητα ίση με 64,21%
5. Logistic Regression με ορθότητα ίση με 64,13%

Κρατάμε συνεπώς ότι από την 4η δοκιμή μας έχουμε ως βέλτιστη ορθότητα το 64,90%. Δεν παρατηρούμε βελτίωση σε σχέση με την προηγούμενη βέλτιστη τιμή της ορθότητας. Συνεπώς δεν υιοθετούμε αυτή την αλλαγή και δοκιμάζουμε την επόμενη. Ας προσπαθήσουμε και πάλι να την ξεπεράσουμε.

1.5. Στη συνέχεια ορίζουμε την παράμετρο remove_multicolinearity να ισούται με Αληθής. Όταν οριστεί σε True, τα χαρακτηριστικά με τις διασυσχετίσεις υψηλότερες από το καθορισμένο όριο καταργούνται (πολυσυγγραμικά). Όταν δύο χαρακτηριστικά συσχετίζονται σε μεγάλο βαθμό μεταξύ τους, αφαιρείται το χαρακτηριστικό που είναι λιγότερο συσχετισμένο με τη μεταβλητή στόχο. Λαμβάνει υπόψη μόνο αριθμητικά χαρακτηριστικά.

Τα αποτελέσματα αυτής της 5ης δοκιμής έχουν ως εξής:

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
gbc	Gradient Boosting Classifier	0.6525	0.6803	0.7783	0.6785	0.7249	0.2590	0.2638	1.2720
et	Extra Trees Classifier	0.6466	0.6826	0.7849	0.6707	0.7232	0.2425	0.2485	0.1390
lr	Logistic Regression	0.6451	0.6748	0.7728	0.6727	0.7191	0.2433	0.2480	0.1810
rf	Random Forest Classifier	0.6437	0.6809	0.7672	0.6730	0.7169	0.2416	0.2458	0.2220
ridge	Ridge Classifier	0.6421	0.0000	0.7711	0.6702	0.7169	0.2365	0.2413	0.0180
lda	Linear Discriminant Analysis	0.6400	0.6719	0.7686	0.6689	0.7152	0.2323	0.2368	0.1100
lightgbm	Light Gradient Boosting Machine	0.6364	0.6686	0.7457	0.6722	0.7069	0.2312	0.2336	0.1840
ada	Ada Boost Classifier	0.6359	0.6579	0.7599	0.6677	0.7107	0.2252	0.2289	0.2690
knn	K Neighbors Classifier	0.6145	0.6275	0.7343	0.6534	0.6913	0.1826	0.1852	0.1720
nb	Naive Bayes	0.6095	0.6549	0.5809	0.7055	0.6358	0.2227	0.2287	0.0180
svm	SVM - Linear Kernel	0.6056	0.0000	0.6883	0.6581	0.6691	0.1779	0.1801	0.0360
dummy	Dummy Classifier	0.5885	0.5000	1.0000	0.5885	0.7409	0.0000	0.0000	0.0160
dt	Decision Tree Classifier	0.5817	0.5697	0.6377	0.6466	0.6421	0.1390	0.1391	0.0790
qda	Quadratic Discriminant Analysis	0.5210	0.5018	0.6107	0.5894	0.5938	0.0038	0.0038	0.0320

Πίνακας 5.5: *Αποτελέσματα 5ης δοκιμής, χρήση παραμέτρων feature_selection, Normalization και remove_multicolinearity*

Σε αυτή την 5η δοκιμή έχουμε το εξής top 5 μοντέλων μηχανικής μάθησης, ως προς την ορθότητα :

1. Gradient Boosting Classifier με ορθότητα ίση με 65,25%
2. Extra Trees Classifier με ορθότητα ίση με 64,66%
3. Logistic Regression με ορθότητα ίση με 64,51%
4. Random Forest Classifier με ορθότητα ίση με 64,37%
5. Ridge Classifier με ορθότητα ίση με 64,21%

Κρατάμε συνεπώς ότι από την 5η δοκιμή μας έχουμε ως βέλτιστη ορθότητα το 65,25%. Παρατηρούμε βελτίωση σε σχέση με την προηγούμενη βέλτιστη τιμή της ορθότητας. Συνεπώς υιοθετούμε αυτή την αλλαγή και δοκιμάζουμε την επόμενη. Ας προσπαθήσουμε και πάλι να την ξεπεράσουμε.

1.6. Στη συνέχεια ορίζουμε την παράμετρο data_split_stratify να ισούται με Αληθής. Ελέγχει τη διαστρωμάτωση κατά τη διάρκεια του "train_test_split". Όταν οριστεί σε True, θα στρωματοποιηθεί κατά στήλη-στόχο. Για στρωματοποίηση σε οποιεσδήποτε άλλες στήλες, πρέπει ο χρήστης να περάσει μια λίστα με τα ονόματα των στηλών. Αγνοείται όταν το data_split_shuffle είναι False.

Τα αποτελέσματα αυτής της 6ης δοκιμής έχουν ως εξής:

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
rf	Random Forest Classifier	0.6555	0.6839	0.7811	0.6805	0.7272	0.2659	0.2708	0.2250
lr	Logistic Regression	0.6535	0.6826	0.7784	0.6792	0.7253	0.2619	0.2665	0.1680
lda	Linear Discriminant Analysis	0.6531	0.6800	0.7801	0.6783	0.7254	0.2603	0.2654	0.0970
ridge	Ridge Classifier	0.6523	0.0000	0.7804	0.6774	0.7251	0.2582	0.2633	0.0180
gbc	Gradient Boosting Classifier	0.6494	0.6882	0.7732	0.6768	0.7215	0.2538	0.2582	1.2270
et	Extra Trees Classifier	0.6484	0.6794	0.7763	0.6746	0.7216	0.2505	0.2555	0.1360
lightgbm	Light Gradient Boosting Machine	0.6425	0.6686	0.7524	0.6762	0.7120	0.2440	0.2468	0.1860
ada	Ada Boost Classifier	0.6400	0.6649	0.7579	0.6720	0.7121	0.2363	0.2397	0.2600
knn	K Neighbors Classifier	0.6180	0.6305	0.7177	0.6614	0.6881	0.1973	0.1987	0.1650
svm	SVM - Linear Kernel	0.6007	0.0000	0.6671	0.6604	0.6593	0.1737	0.1771	0.0350
nb	Naive Bayes	0.5891	0.6645	0.4884	0.7222	0.5807	0.2072	0.2228	0.0170
dummy	Dummy Classifier	0.5876	0.5000	1.0000	0.5876	0.7403	0.0000	0.0000	0.0150
dt	Decision Tree Classifier	0.5791	0.5680	0.6309	0.6452	0.6374	0.1355	0.1359	0.0720
qda	Quadratic Discriminant Analysis	0.5267	0.4856	0.7201	0.5781	0.6389	-0.0304	-0.0333	0.0310

Πίνακας 5.6: *Αποτελέσματα 6ης δοκιμής, χρήση παραμέτρων feature_selection, Normalization, remove_multicolinearity και data_split_stratify*

Σε αυτή την 6η δοκιμή έχουμε το εξής top 5 μοντέλων μηχανικής μάθησης, ως προς την ορθότητα :

1. Random Forest Classifier με ορθότητα ίση με 65,55%
2. Logistic Regression με ορθότητα ίση με 65,35%
3. Linear Discriminant Analysis με ορθότητα ίση με 65,31%
4. Ridge Classifier με ορθότητα ίση με 65,23%
5. Gradient Boosting Classifier με ορθότητα ίση με 64,94%

Κρατάμε συνεπώς ότι από την 6η δοκιμή μας έχουμε ως βέλτιστη ορθότητα το 65,55%. Παρατηρούμε βελτίωση σε σχέση με την προηγούμενη βέλτιστη τιμή της ορθότητας. Συνεπώς υιοθετούμε αυτή την αλλαγή και δοκιμάζουμε την επόμενη. Ας προσπαθήσουμε και πάλι να την ξεπεράσουμε.

1.7. Στη συνέχεια ορίζουμε την παράμετρο PCA να ισούται με Αληθής. Όταν οριστεί σε True, εφαρμόζεται μείωση διαστάσεων για την προβολή των δεδομένων σε χώρο χαμηλότερων διαστάσεων χρησιμοποιώντας τη μέθοδο που ορίζεται στην παράμετρο pca_method, στην περίπτωσή μας με γραμμική μέθοδο.

Τα αποτελέσματα αυτής της 7ης και τελευταίας δοκιμής έχουν ως εξής:

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
ridge	Ridge Classifier	0.6529	0.0000	0.7912	0.6746	0.7281	0.2563	0.2628	0.0160
lda	Linear Discriminant Analysis	0.6527	0.6857	0.7891	0.6750	0.7275	0.2565	0.2627	0.0350
lr	Logistic Regression	0.6515	0.6857	0.7874	0.6743	0.7263	0.2541	0.2602	0.0260
lightgbm	Light Gradient Boosting Machine	0.6492	0.6725	0.7738	0.6761	0.7215	0.2532	0.2578	0.1700
rf	Random Forest Classifier	0.6470	0.6794	0.7884	0.6698	0.7241	0.2428	0.2493	0.2970
gbc	Gradient Boosting Classifier	0.6470	0.6808	0.7794	0.6725	0.7218	0.2458	0.2510	1.2540
et	Extra Trees Classifier	0.6425	0.6655	0.8456	0.6509	0.7354	0.2119	0.2301	0.1200
ada	Ada Boost Classifier	0.6250	0.6495	0.7360	0.6632	0.6975	0.2074	0.2096	0.2730
nb	Naive Bayes	0.6209	0.6399	0.6549	0.6858	0.6691	0.2254	0.2264	0.0160
knn	K Neighbors Classifier	0.6186	0.6310	0.7194	0.6615	0.6889	0.1981	0.1997	0.0970
qda	Quadratic Discriminant Analysis	0.6160	0.6458	0.6414	0.6850	0.6621	0.2184	0.2192	0.0200
svm	SVM - Linear Kernel	0.6019	0.0000	0.6778	0.6573	0.6660	0.1724	0.1735	0.0240
dummy	Dummy Classifier	0.5876	0.5000	1.0000	0.5876	0.7403	0.0000	0.0000	0.0140
dt	Decision Tree Classifier	0.5671	0.5540	0.6281	0.6322	0.6299	0.1082	0.1082	0.0710

Πίνακας 5.7: *Αποτελέσματα 7ης δοκιμής, χρήση παραμέτρων feature_selection, Normalization, remove_multicolinearity, data_split_stratify και PCA*

1. Ridge Classifier με ορθότητα ίση με 65,29%
2. Linear Discriminant Analysis με ορθότητα ίση με 65,27%
3. Logistic Regression με ορθότητα ίση με 65,15%
4. Light Gradient Boosting Machine με ορθότητα ίση με 64,92%
5. Random Forest Classifier με ορθότητα ίση με 64,70%

Κρατάμε συνεπώς ότι από την 7η δοκιμή μας έχουμε ως βέλτιστη ορθότητα το 65,29%. Παρατηρούμε μείωση σε σχέση με την προηγούμενη βέλτιστη τιμή της ορθότητας. Εδώ η μείωση είναι τόσο μικρή που πρακτικά μας συμφέρει να επιλέξουμε τη δοκιμή με το PCA, καθώς έχουμε αισθητό κέρδος σε υπολογιστική ισχύ, λόγω της μείωσης της διαστατικότητας. Από την άλλη όμως, μας ενδιαφέρει να εντοπίσουμε τα 15 εκείνα χαρακτηριστικά που διαδραματίζουν σημαντικότερο ρόλο στην έκβαση ενός αγώνα NBA. Συνεπώς δεν υιοθετούμε αυτή την αλλαγή, καθώς πηγαίνοντας σε νέο χώρο διαστάσεων, θα είναι δύσκολο να επιστρέψουμε και να να βρούμε αυτά τα 10 καλύτερα αρχικά χαρακτηριστικά.

1.8. Αλλαγή στην παράμετρο normalize_method από zscore σε min-max scaler.

Τα αποτελέσματα αυτής της 8ης και τελευταίας δοκιμής έχουν ως εξής:

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
lr	Logistic Regression	0.6578	0.6932	0.7908	0.6796	0.7308	0.2682	0.2746	0.0250
lda	Linear Discriminant Analysis	0.6576	0.6929	0.7908	0.6794	0.7307	0.2677	0.2741	0.0330
ridge	Ridge Classifier	0.6572	0.0000	0.7922	0.6786	0.7308	0.2663	0.2729	0.0180
svm	SVM - Linear Kernel	0.6529	0.0000	0.7614	0.6851	0.7198	0.2658	0.2709	0.0270
rf	Random Forest Classifier	0.6437	0.6691	0.8106	0.6608	0.7277	0.2272	0.2377	0.3040
qda	Quadratic Discriminant Analysis	0.6405	0.6759	0.6653	0.7064	0.6844	0.2671	0.2685	0.0220
gbc	Gradient Boosting Classifier	0.6368	0.6655	0.7891	0.6600	0.7185	0.2177	0.2250	1.1540
et	Extra Trees Classifier	0.6345	0.6473	0.8047	0.6538	0.7212	0.2067	0.2166	0.1160
lightgbm	Light Gradient Boosting Machine	0.6303	0.6524	0.7666	0.6596	0.7088	0.2098	0.2149	0.1560
nb	Naive Bayes	0.6274	0.6516	0.7131	0.6726	0.6916	0.2213	0.2228	0.0190
ada	Ada Boost Classifier	0.6209	0.6444	0.7326	0.6600	0.6941	0.1989	0.2011	0.2530
knn	K Neighbors Classifier	0.5917	0.5957	0.7197	0.6348	0.6743	0.1327	0.1348	0.0990
dummy	Dummy Classifier	0.5876	0.5000	1.0000	0.5876	0.7403	0.0000	0.0000	0.0180
dt	Decision Tree Classifier	0.5463	0.5316	0.6153	0.6137	0.6144	0.0632	0.0633	0.0700

Πίνακας 5.8: *Αποτελέσματα 8ης δοκιμής, χρήση παραμέτρων feature_selection, Normalization with min-max scaler, remove_multicolinearity και data_split_stratify*

1. Logistic Regression με ορθότητα ίση με 65,78%
2. Linear Discriminant Analysis με ορθότητα ίση με 65,76%
3. Ridge Classifier με ορθότητα ίση με 65,72%
4. SVM - Linear Kernel με ορθότητα ίση με 65,29%
5. Random Forest Classifier με ορθότητα ίση με 64,37%

Κρατάμε συνεπώς ότι από την 8η δοκιμή μας έχουμε ως βέλτιστη ορθότητα το 65,78%. Παρατηρούμε αύξηση σε σχέση με την προηγούμενη βέλτιστη τιμή της ορθότητας. Συνεπώς υιοθετούμε αυτή την αλλαγή. Αυτή είναι και η τελευταία δοκιμή μας και πετύχαμε πλέον τη βέλτιστη ορθότητα για το πιο απλό σύνολο δεδομένων μας, το "Dummy". Ας δοκιμάσουμε τώρα πώς για τις συγκεκριμένες παραμέτρους της συνάρτησης setup(), ανταπεξέρχονται και τα υπόλοιπα σύνολα δεδομένων μας, πάνω στη λογική που είχαμε θέσει. Θα κρατήσουμε εκείνο το σύνολο δεδομένων που θα μας δώσει και πάλι τη βέλτιστη ορθότητα ως προς τις default παραμέτρους των μοντέλων μας στο σύνολο επικύρωσης.

Συνοψίζοντας θα λέγαμε ότι η βέλτιστη setup() συνάρτηση έχει τις εξείς παραμέτρους :

feature_selection, Normalization with min-max scaler, remove_multicolinearity και data_split_stratify

Ο διαχωρισμός σε train-test είναι by default από το pycaret 70-30 και εφαρμόζεται 10-fold cross validation.

Συνολικά έχουμε ότι :

Δοκιμές μέχρι την έυρεση της βέλτιστης setup() συνάρτησης.

	Παράμετρος	Accuracy	Αποδοχή	Βέλτιστος Αλγόριθμος
Δοκιμή 1	feature_selection	64,68%	ΑΡΧΗ	**Logistic Regression**
Δοκιμή 2	zscore normalization	65%	ΝΑΙ	**Gradient Boosting Classifier**
Δοκιμή 3	transformation	64,53%	ΟΧΙ	**Extra Trees Classifier**
Δοκιμή 4	ignore_low_variance	64,90%	ΟΧΙ	**Extra Trees Classifier**
Δοκιμή 5	remove_multicolinearity	65,25%	ΝΑΙ	**Gradient Boosting Classifier**
Δοκιμή 6	data_split_stratify	65,55%	ΝΑΙ	**Random Forest Classifier**
Δοκιμή 7	PCA	65,29%	ΟΧΙ	**Ridge Classifier**
Δοκιμή 8	min-max normalization	65,78%	ΑΝΤΙΚΑΤΑΣΤΑΣΗ	**Logistic Regression**

Πίνακας 5.9: *Δοκιμές μέχρι την έυρεση της βέλτιστης setup() συνάρτησης*

5.2.2 Εύρεση βέλτιστου συνόλου δεδομένων.

2.1. Δοκιμή με σύνολο δεδομένων "Last_3"

Υπενθυμίζουμε ότι αυτό το σύνολο δεδομένων περιλαμβάνει όσα χαρακτηριστικά είχε και το "Dummy" σύνολο δεδομένων, καθώς και νέα χαρακτηριστικά που αφορούν τη φόρμα των ομάδων στους τελευταίους 3 αγώνες ή αντίστοιχα N αγώνες για τις διαφορετικές τιμές του N (3,5,7,9,10). Για το λόγο αυτό οι πρώτοι 3 αγώνες σε αυτή την περίπτωση φέρουν μηδενική πληροφορία για τα Last_N χαρακτηριστικά.

Τα αποτελέσματα αυτής της δοκιμής έχουν ως εξής:

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
lr	Logistic Regression	0.6523	0.6864	0.7766	0.6784	0.7240	0.2596	0.2644	0.3760
ridge	Ridge Classifier	0.6515	0.0000	0.7777	0.6772	0.7238	0.2574	0.2624	0.0180
lda	Linear Discriminant Analysis	0.6512	0.6846	0.7735	0.6783	0.7226	0.2583	0.2627	0.0650
catboost	CatBoost Classifier	0.6480	0.6799	0.8186	0.6620	0.7319	0.2354	0.2476	8.0050
svm	SVM - Linear Kernel	0.6407	0.0000	0.7416	0.6774	0.7063	0.2439	0.2483	0.0320
rf	Random Forest Classifier	0.6347	0.6565	0.8619	0.6407	0.7349	0.1865	0.2097	0.3690
qda	Quadratic Discriminant Analysis	0.6337	0.6688	0.6698	0.6962	0.6819	0.2501	0.2512	0.0250
gbc	Gradient Boosting Classifier	0.6337	0.6625	0.7988	0.6541	0.7190	0.2071	0.2168	1.9780
lightgbm	Light Gradient Boosting Machine	0.6294	0.6539	0.7738	0.6567	0.7103	0.2054	0.2113	0.3760
xgboost	Extreme Gradient Boosting	0.6223	0.6423	0.7576	0.6544	0.7020	0.1936	0.1977	1.7730
nb	Naive Bayes	0.6211	0.6459	0.7194	0.6641	0.6903	0.2039	0.2054	0.0170
et	Extra Trees Classifier	0.6188	0.6397	0.8911	0.6230	0.7332	0.1340	0.1635	0.1310
ada	Ada Boost Classifier	0.6146	0.6428	0.7405	0.6511	0.6927	0.1811	0.1844	0.4130
knn	K Neighbors Classifier	0.5836	0.5900	0.7069	0.6299	0.6658	0.1177	0.1194	0.2720
dt	Decision Tree Classifier	0.5542	0.5408	0.6174	0.6214	0.6191	0.0816	0.0817	0.0930

Πίνακας 5.10: *Αποτελέσματα δοκιμής Last_3 συνόλου δεδομένων, χρήση παραμέτρων feature_selection, Normalization with min-max scaler, remove_multicolinearity και data_split_stratify*

1. Logistic Regression με ορθότητα ίση με 65,23%
2. Ridge Classifier με ορθότητα ίση με 65,15%
3. Linear Discriminant Analysis με ορθότητα ίση με 65,12%
4. CatBoost Classifier με ορθότητα ίση με 64,80%
5. SVM - Linear Kernel με ορθότητα ίση με 64,07%

Κρατάμε συνεπώς ότι από τη δοκιμή μας έχουμε ως βέλτιστη ορθότητα το 65,23%. Παρατηρούμε μείωση σε σχέση με την προηγούμενη βέλτιστη τιμή της ορθότητας. Συνεπώς δεν υιοθετούμε αυτή την αλλαγή και δοκιμάζουμε την επόμενη. Ας προσπαθήσουμε και πάλι να την ξεπεράσουμε.

2.2. Δοκιμή με σύνολο δεδομένων "Last_5"

Τα αποτελέσματα αυτής της δοκιμής έχουν ως εξής:

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
lr	Logistic Regression	0.6563	0.6919	0.7856	0.6797	0.7285	0.2666	0.2727	0.3670
ridge	Ridge Classifier	0.6541	0.0000	0.7856	0.6776	0.7272	0.2611	0.2673	0.0180
lda	Linear Discriminant Analysis	0.6541	0.6900	0.7811	0.6789	0.7260	0.2626	0.2683	0.0530
catboost	CatBoost Classifier	0.6515	0.6799	0.8096	0.6679	0.7318	0.2468	0.2567	7.1950
svm	SVM - Linear Kernel	0.6404	0.0000	0.7603	0.6718	0.7127	0.2365	0.2407	0.0310
lightgbm	Light Gradient Boosting Machine	0.6390	0.6650	0.7752	0.6657	0.7160	0.2281	0.2338	0.3500
rf	Random Forest Classifier	0.6376	0.6608	0.8328	0.6497	0.7297	0.2042	0.2197	0.3870
gbc	Gradient Boosting Classifier	0.6347	0.6669	0.7995	0.6552	0.7200	0.2091	0.2181	1.7780
qda	Quadratic Discriminant Analysis	0.6325	0.6683	0.6694	0.6946	0.6812	0.2474	0.2482	0.0250
nb	Naive Bayes	0.6284	0.6499	0.7284	0.6691	0.6968	0.2184	0.2207	0.0160
et	Extra Trees Classifier	0.6284	0.6517	0.8685	0.6344	0.7331	0.1677	0.1918	0.1460
ada	Ada Boost Classifier	0.6270	0.6443	0.7454	0.6623	0.7011	0.2093	0.2126	0.3760
xgboost	Extreme Gradient Boosting	0.6219	0.6420	0.7433	0.6580	0.6979	0.1975	0.2003	1.6600
knn	K Neighbors Classifier	0.5824	0.5837	0.7010	0.6302	0.6633	0.1170	0.1184	0.2540
dt	Decision Tree Classifier	0.5638	0.5500	0.6285	0.6288	0.6282	0.1002	0.1004	0.0990

Πίνακας 5.11: *Αποτελέσματα δοκιμής Last_5 συνόλου δεδομένων, χρήση παραμέτρων feature_selection, Normalization with min-max scaler, remove_multicolinearity και data_split_stratify*

1. Logistic Regression με ορθότητα ίση με 65,63%
2. Ridge Classifier με ορθότητα ίση με 65,41%
3. Linear Discriminant Analysis με ορθότητα ίση με 65,41Ξ%
4. CatBoost Classifier με ορθότητα ίση με 65,15%
5. SVM - Linear Kernel με ορθότητα ίση με 64,04%

Κρατάμε συνεπώς ότι από τη δοκιμή μας έχουμε ως βέλτιστη ορθότητα το 65,63%. Παρατηρούμε μείωση σε σχέση με την προηγούμενη βέλτιστη τιμή της ορθότητας. Συνεπώς δεν υιοθετούμε αυτή την αλλαγή και δοκιμάζουμε την επόμενη. Ας προσπαθήσουμε και πάλι να την ξεπεράσουμε.

2.3. Δοκιμή με σύνολο δεδομένων "Last_7"

Τα αποτελέσματα αυτής της δοκιμής έχουν ως εξής:

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
lr	Logistic Regression	0.6614	0.6957	0.7888	0.6838	0.7323	0.2778	0.2835	0.3660
lda	Linear Discriminant Analysis	0.6598	0.6938	0.7877	0.6825	0.7311	0.2742	0.2799	0.0500
ridge	Ridge Classifier	0.6596	0.0000	0.7898	0.6816	0.7316	0.2730	0.2791	0.0210
gbc	Gradient Boosting Classifier	0.6498	0.6815	0.8030	0.6684	0.7294	0.2448	0.2534	1.6500
catboost	CatBoost Classifier	0.6449	0.6846	0.7940	0.6662	0.7243	0.2360	0.2436	7.0720
svm	SVM - Linear Kernel	0.6435	0.0000	0.7444	0.6801	0.7100	0.2491	0.2519	0.0320
lightgbm	Light Gradient Boosting Machine	0.6392	0.6734	0.7693	0.6673	0.7145	0.2308	0.2357	0.3800
qda	Quadratic Discriminant Analysis	0.6364	0.6742	0.6691	0.6993	0.6834	0.2566	0.2574	0.0280
rf	Random Forest Classifier	0.6351	0.6673	0.8269	0.6489	0.7269	0.2004	0.2148	0.3560
ada	Ada Boost Classifier	0.6292	0.6507	0.7433	0.6654	0.7019	0.2153	0.2180	0.3530
et	Extra Trees Classifier	0.6254	0.6471	0.8543	0.6347	0.7281	0.1655	0.1864	0.1300
nb	Naive Bayes	0.6227	0.6466	0.7332	0.6613	0.6950	0.2033	0.2059	0.0190
xgboost	Extreme Gradient Boosting	0.6196	0.6431	0.7381	0.6572	0.6950	0.1940	0.1967	1.4770
knn	K Neighbors Classifier	0.5870	0.5902	0.7097	0.6326	0.6687	0.1251	0.1268	0.2500
dt	Decision Tree Classifier	0.5563	0.5441	0.6132	0.6246	0.6186	0.0880	0.0882	0.0940

Πίνακας 5.12: *Αποτελέσματα δοκιμής Last_7 συνόλου δεδομένων, χρήση παραμέτρων feature_selection, Normalization with min-max scaler, remove_multicolinearity και data_split_stratify*

1. Logistic Regression με ορθότητα ίση με 66,14%
2. Linear Discriminant Analysis με ορθότητα ίση με 65,98%
3. Ridge Classifier με ορθότητα ίση με 65,96%
4. Gradient Boosting Classifier με ορθότητα ίση με 64,98%
5. CatBoost Classifier με ορθότητα ίση με 64,49%

Κρατάμε συνεπώς ότι από τη δοκιμή μας έχουμε ως βέλτιστη ορθότητα το 66,14%. Παρατηρούμε αύξηση σε σχέση με την προηγούμενη βέλτιστη τιμή της ορθότητας. Συνεπώς υιοθετούμε αυτή την αλλαγή και δοκιμάζουμε την επόμενη. Αυτή είναι και η καλύτερη επιλογή της λογικής των Last_N συνόλων δεδομένων. Συνεπώς όταν έχουμε την πληροφορία για τη φόρμα των ομάδων στους τελευταίους 7 αγώνες, πετυχαίνουμε οριακά καλύτερη ορθότητα. Ας προσπαθήσουμε και πάλι να την ξεπεράσουμε.

2.4. Δοκιμές με σύνολα δεδομένων "Drop_N_Last_7"

Σε προηγούμενο κεφάλαιο εξηγήσαμε τη λογική των Drop_N συνόλων δεδομένων. Ας δούμε αν όντως η λογική μας επαληθεύεται. Σε αυτό το παράδειγμα αποφασίζουμε να αφαιρέσουμε 10 αγωνιστικές από τα δεδομένα μας από το Last_7 σύνολο δεδομένων. Η επιλογή δεν ήταν και τόσο τυχαία. Οι πρώτες 7 αφορούν τους πρώτους 7 αγώνες που έχουν μηδενικές τιμές στα χαρακτηριστικά της φόρμας μιας ομάδας και ακόμη 3 αγωνιστικές. Ομοίως δοκιμές έγιναν και για αφαίρεση 15 και 20 αγωνιστικών ή και για αφαίρεση αγωνιστικών και από το τέλος. Σε όλες τις περιπτώσεις τα αποτελέσματα ως προς την ορθότητα χειροτέρευαν, απορρίπτοντας την αρχική μας υπόθεση. Ο πιθανότερος λόγος που απορρίφθηκε αυτή μας η υπόθεση είναι ότι έχουμε ήδη περιορισμένο αριθμό δεδομένων και δεν υπάρχει η ¨πολυτέλεια¨ κατάργησης αγωνιστικών. Συνεπώς η λογική αυτή δεν υιοθετήθηκε και απορρίφθηκαν όλα τα σχετικά σύνολα δεδομένων, επιστρέφοντας στο "Last_7" ως μέχρι τώρα βέλτιστο σύνολο δεδομένων.

Ενδεικτικά τα αποτελέσματα από τα "Drop_10_Last_7" και "Drop_15_Last_7" σύνολα δεδομένων έχουν ως εξής :

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
lr	Logistic Regression	0.6563	0.6970	0.7607	0.6862	0.7213	0.2758	0.2787	0.3600
catboost	CatBoost Classifier	0.6532	0.6863	0.8167	0.6661	0.7337	0.2511	0.2624	8.5240
ridge	Ridge Classifier	0.6513	0.0000	0.7555	0.6827	0.7170	0.2657	0.2684	0.0190
lda	Linear Discriminant Analysis	0.6510	0.6931	0.7522	0.6836	0.7160	0.2660	0.2684	0.0570
gbc	Gradient Boosting Classifier	0.6480	0.6825	0.7979	0.6664	0.7261	0.2445	0.2529	1.5290
svm	SVM - Linear Kernel	0.6408	0.0000	0.7207	0.6843	0.7009	0.2516	0.2536	0.0310
nb	Naive Bayes	0.6400	0.6655	0.7433	0.6749	0.7073	0.2426	0.2447	0.0180
qda	Quadratic Discriminant Analysis	0.6395	0.6887	0.6811	0.6965	0.6884	0.2607	0.2611	0.0280
rf	Random Forest Classifier	0.6378	0.6646	0.8757	0.6391	0.7387	0.1927	0.2211	0.3080
lightgbm	Light Gradient Boosting Machine	0.6326	0.6670	0.7584	0.6627	0.7071	0.2201	0.2241	0.3810
et	Extra Trees Classifier	0.6290	0.6574	0.8869	0.6300	0.7366	0.1660	0.1976	0.1090
xgboost	Extreme Gradient Boosting	0.6216	0.6533	0.7367	0.6578	0.6946	0.2009	0.2037	1.3250
ada	Ada Boost Classifier	0.6196	0.6461	0.7334	0.6568	0.6925	0.1975	0.2005	0.3260
knn	K Neighbors Classifier	0.6025	0.6161	0.7047	0.6477	0.6747	0.1661	0.1673	0.2150
dt	Decision Tree Classifier	0.5681	0.5566	0.6241	0.6328	0.6283	0.1128	0.1128	0.0800

Πίνακας 5.13: *Αποτελέσματα δοκιμής "Drop_10_Last_7" συνόλου δεδομένων, χρήση παραμέτρων feature_selection, Normalization with min-max scaler, remove_multicolinearity και data_split_stratify*

και

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
gbc	Gradient Boosting Classifier	0.6514	0.6813	0.7945	0.6688	0.7259	0.2566	0.2648	1.2500
catboost	CatBoost Classifier	0.6498	0.6951	0.8008	0.6656	0.7267	0.2505	0.2592	8.7710
lr	Logistic Regression	0.6487	0.6928	0.7494	0.6799	0.7124	0.2635	0.2664	0.3570
ridge	Ridge Classifier	0.6464	0.0000	0.7459	0.6786	0.7103	0.2589	0.2615	0.0170
lda	Linear Discriminant Analysis	0.6450	0.6873	0.7419	0.6785	0.7083	0.2570	0.2593	0.0460
lightgbm	Light Gradient Boosting Machine	0.6437	0.6754	0.7713	0.6683	0.7159	0.2447	0.2493	0.3670
xgboost	Extreme Gradient Boosting	0.6377	0.6642	0.7552	0.6669	0.7080	0.2354	0.2390	1.0810
qda	Quadratic Discriminant Analysis	0.6353	0.6772	0.6893	0.6866	0.6875	0.2495	0.2500	0.0240
nb	Naive Bayes	0.6283	0.6632	0.7367	0.6626	0.6974	0.2189	0.2213	0.0170
rf	Random Forest Classifier	0.6276	0.6556	0.8551	0.6334	0.7275	0.1786	0.2006	0.2410
et	Extra Trees Classifier	0.6266	0.6575	0.8799	0.6279	0.7326	0.1675	0.1969	0.0960
ada	Ada Boost Classifier	0.6239	0.6571	0.7119	0.6656	0.6876	0.2164	0.2175	0.2730
svm	SVM - Linear Kernel	0.6195	0.0000	0.7136	0.6609	0.6846	0.2055	0.2081	0.0260
knn	K Neighbors Classifier	0.6148	0.6343	0.7038	0.6578	0.6796	0.1978	0.1991	0.2050
dt	Decision Tree Classifier	0.5551	0.5446	0.6085	0.6193	0.6135	0.0891	0.0893	0.0630

Πίνακας 5.14: *Αποτελέσματα δοκιμής "Drop_15_Last_7" συνόλου δεδομένων, χρήση παραμέτρων feature_selection, Normalization with min-max scaler, remove_multicolinearity και data_split_stratify*

2.5. Δοκιμές με σύνολο δεδομένων "plus_27"

Σε αυτό το σημείο δοκιμάζουμε αν τα νέα 27 χαρακτηριστικά που κατασκευάσαμε μπορούν να αυξήσουν περισσότερο την ορθότητα των μοντέλων μας. Τα αποτελέσματα έχουν ως εξής:

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
lr	Logistic Regression	0.6647	0.6976	0.7877	0.6875	0.7340	0.2860	0.2914	0.0330
ridge	Ridge Classifier	0.6610	0.0000	0.7863	0.6842	0.7315	0.2776	0.2830	0.0170
lda	Linear Discriminant Analysis	0.6594	0.6960	0.7815	0.6841	0.7293	0.2752	0.2802	0.0500
svm	SVM - Linear Kernel	0.6502	0.0000	0.7811	0.6752	0.7240	0.2532	0.2587	0.0310
catboost	CatBoost Classifier	0.6500	0.6852	0.8120	0.6662	0.7317	0.2422	0.2525	7.1620
gbc	Gradient Boosting Classifier	0.6490	0.6783	0.8127	0.6648	0.7312	0.2397	0.2504	1.7450
lightgbm	Light Gradient Boosting Machine	0.6427	0.6697	0.7780	0.6684	0.7188	0.2360	0.2417	0.2940
qda	Quadratic Discriminant Analysis	0.6380	0.6781	0.6753	0.6989	0.6863	0.2583	0.2592	0.0230
rf	Random Forest Classifier	0.6368	0.6637	0.8429	0.6468	0.7316	0.1985	0.2163	0.3700
et	Extra Trees Classifier	0.6327	0.6596	0.8775	0.6361	0.7374	0.1752	0.2029	0.1320
nb	Naive Bayes	0.6309	0.6480	0.7367	0.6690	0.7007	0.2215	0.2241	0.0170
ada	Ada Boost Classifier	0.6282	0.6452	0.7409	0.6649	0.7005	0.2137	0.2166	0.3670
xgboost	Extreme Gradient Boosting	0.6272	0.6452	0.7541	0.6601	0.7037	0.2066	0.2105	1.4840
knn	K Neighbors Classifier	0.6027	0.6130	0.7152	0.6468	0.6789	0.1611	0.1629	0.1190
dt	Decision Tree Classifier	0.5585	0.5453	0.6205	0.6247	0.6224	0.0908	0.0909	0.0900

Πίνακας 5.15: *Αποτελέσματα δοκιμής "plus_27" συνόλου δεδομένων, χρήση παραμέτρων feature_selection, Normalization with min-max scaler, remove_multicolinearity και data_split_stratify*

1. Logistic Regression με ορθότητα ίση με 66,47%
2. Ridge Classifier με ορθότητα ίση με 66,10%
3. Linear Discriminant Analysis με ορθότητα ίση με 65,94%
4. SVM - Linear Kernel με ορθότητα ίση με 65,02%

5. CatBoost Classifier με ορθότητα ίση με 65,00%

Συνολικά έχουμε ότι :

Δοκιμές μέχρι την έυρεση του βέλτιστου συνόλου δεδομένων.

	Σύνολο Δεδομένων	Accuracy	Αποδοχή	Βέλτιστος Αλγόριθμος
Δοκιμή 1	Last_3	65,23 %	OXI	**Logistic Regression**
Δοκιμή 2	Last 5	65,63%	OXI	**Logistic Regression**
Δοκιμή 3	Last 7	66,14%	NAI	**Logistic Regression**
Δοκιμή 4	Drop_10_Last_7	65,63%	OXI	**Logistic Regression**
Δοκιμή 5	Drop_15_Last_7	65,14%	OXI	**Gradient Boosting Classifier**
Δοκιμή 6	Plus_27	66,47%	NAI	**Logistic Regression**

Πίνακας 5.16: *Δοκιμές μέχρι την έυρεση του βέλτιστου συνόλου δεδομένων*

Άρα στο σύνολο δεδομένων plus_27 έχουμε τη μέχρι στιγμής βέλτιστη τιμή ορθότητας.

3. Στο σημείο αυτό υπήρξε ένα πολύ σημαντικό εύρημα, καθοριστικό για τη συνέχεια της εργασίας. Ενώ εφαρμόστηκε βελτιστοποίηση υπερπαραμέτρων στο μοντέλο με την καλύτερη μέχρις στιγμής ορθότητα στις default παραμέτρους του, στο σύνολο επικύρωσης, παρατηρήθηκε ότι αγγίζαμε ορθότητα μέχρι και 67%, πράγμα που μας έκανε ιδιαίτερα χαρούμενους. Όταν όμως χρησιμοποιούσαμε ένα κομμάτι των δεδομένων ως ¨κρυφά¨ (test_set), δηλαδή τελείως καινούργια και προσπαθήσαμε να βρούμε την ορθότητα του μοντέλου μας σε αυτά, τότε παρατηρήσαμε ότι δεν τα πήγαινε εξίσου καλά. Με άλλα λόγια, όταν το μοντέλο προέβλεπε με το validation set τα πήγαινε πολύ ικανοποιητικά ως προς την ορθότητα των προβλέψεών του, αλλά όχι εξίσου καλά στις προβλέψεις του στο test set. Ο λόγος ήταν απλός. Το μοντέλο μας υπερπροσαρμοζόταν στα δεδομένα εκπαίδευσης και δεν γενίκευε εξίσου καλά στα test data. Αυτό συνέβαινε γιατί ανακατεύαμε τα δεδομένα μας, πράγμα που θεωρείται λανθασμένο, καθώς το μοντέλο εκπαιδευόταν σε αγώνες π.χ. της 1ης σεζόν που έγιναν χρονικά αργότερα από τους αγώνες που υπήρχαν στο test set και έγιναν χρονικά νωρίτερα.

Για να αποφευχθεί αυτή η προκατάληψη και να καταλήξουμε στο πραγματικά βέλτιστο για εμάς μοντέλο, το οποίο θα γενικεύει εξίοσου καλά, ορίζουμε τη μεταβλητή data_split_shuffle ίση με False και επιλέγουμε να εφαρμόσουμε διασταυρούμενη επικύρωση με 5-Fold Cross Validation. Στα δεδομένα εκπαίδευσης έχουμε ένα σύνολο δεδομένων με τη σεζόν 2017-2018 ¨κρυφή¨ και τις υπόλοιπες τις δίνουμε στο μοντέλο μας για train και validation. Έτσι κάθε φορά σπάμε τα δεδομένα μας σε 4-1-1 (train-validation-test split) δοκιμάζοντας όλους τους πιθανούς συνδυασμούς με τη σεζόν 2017-2018 να είναι ¨κρυφή¨. Στη συνέχεια επαναλαμβάνουμε την ίδια διαδικασία, κρατώντας κάθε φορά διαφορετική σεζόν για test set και προφανώς αντίστοιχες τετράδες για σύνολα εκπαίδευσης. Αν κατά μέσο όρο στα 6 πιθανά πειράματα (1 για κάθε διαφορετική test set σεζόν) προβλέπουμε το test set κατά 65% και πάνω, τότε έχουμε πετύχει το σκοπό αυτής της εργασίας.

Όπως προαναφέρθηκε, πραγματοποιήθηκαν 6 διαφορετικά πειράματα για να επαληθευτεί η τελική ορθότητα του μοντέλου μας.

- **1ο Πείραμα - Σεζόν 2012-2013, 2013-2014, 2014-2015, 2015-2016, 2016-2017 για σύνολο εκπαίδευσης-επικύρωσης και 2017-2018 για σύνολο ελέγχου.** Για αυτό το πείραμα εφαρμόστηκε 5 fold cross validation με κάθε μία από τις σεζόν του συνόλου εκπαίδευσης, κάθε φορά, να χρησιμοποιείται ως σύνολο επικύρωσσης. Κατά μέσο όρο και στις 5 φορές η υψηλότερη τιμή ορθότητας στο σύνολο επικύρωσης του μοντέλου μας επιτεύχθηκε με τον αλγόριθμο CatBoost Classifier της τάξεως του 66,29%. Στη συνέχεια, κατόπιν βελτιστοποίησης υπερπαραμέτρων, δοκιμάστηκε και η ορθότητα στο και στο σύνολο επικύρωσης, αλλά και σύνολο ελέγχου, η οποία έφτασε το 66,77% και 68% αντίστοιχα. Η διαδικασία αυτή επαναλήφθηκε και για τις υπόλοιπες σεζόν.

- **2ο Πείραμα - Σεζόν 2012-2013, 2013-2014, 2014-2015, 2015-2016, 2017-2018 για σύνολο εκπαίδευσης-επικύρωσης και 2016-2017 για σύνολο ελέγχου.** Κατά μέσο όρο και στις 5 φορές η υψηλότερη τιμή ορθότητας στο σύνολο επικύρωσης του μοντέλου μας επιτεύχθηκε με τον αλγόριθμο CatBoost Classifier της τάξεως του 65,54%. Στη συνέχεια, κατόπιν βελτιστοποίησης υπερπαραμέτρων, δοκιμάστηκε και η ορθότητα στο και στο σύνολο επικύρωσης, αλλά και σύνολο ελέγχου, η οποία έφτασε το 66,04% και 65,92% αντίστοιχα.

- **3ο Πείραμα - Σεζόν 2012-2013, 2013-2014, 2014-2015, 2017-2018, 2016-2017 για σύνολο εκπαίδευσης-επικύρωσης και 2015-2016 για σύνολο ελέγχου.** Κατά μέσο όρο και στις 5 φορές η υψηλότερη τιμή ορθότητας στο σύνολο επικύρωσης του μοντέλου μας επιτεύχθηκε με τον αλγόριθμο CatBoost Classifier της τάξεως του 65,83%. Στη συνέχεια, κατόπιν βελτιστοποίησης υπερπαραμέτρων, δοκιμάστηκε και η ορθότητα στο και στο σύνολο επικύρωσης, αλλά και σύνολο ελέγχου, η οποία έφτασε το 67,15% και 67,22% αντίστοιχα.

- **4ο Πείραμα - Σεζόν 2012-2013, 2013-2014, 2017-2018, 2015-2016, 2016-2017 για σύνολο εκπαίδευσης-επικύρωσης και 2014-2015 για σύνολο ελέγχου.** Κατά μέσο όρο και στις 5 φορές η υψηλότερη τιμή ορθότητας στο σύνολο επικύρωσης του μοντέλου μας επιτεύχθηκε με τον αλγόριθμο CatBoost Classifier της τάξεως του 66,61%. Στη συνέχεια, κατόπιν βελτιστοποίησης υπερπαραμέτρων, δοκιμάστηκε και η ορθότητα στο και στο σύνολο επικύρωσης, αλλά και σύνολο ελέγχου, η οποία έφτασε το 67,01% και 66,47% αντίστοιχα.

- **5ο Πείραμα - Σεζόν 2012-2013, 2017-2018, 2014-2015, 2015-2016, 2016-2017 για σύνολο εκπαίδευσης-επικύρωσης και 2013-2014 για σύνολο ελέγχου.** Κατά μέσο όρο και στις 5 φορές η υψηλότερη τιμή ορθότητας στο σύνολο επικύρωσης του μοντέλου μας επιτεύχθηκε με τον αλγόριθμο CatBoost Classifier της τάξεως του 64,92%. Στη συνέχεια, κατόπιν βελτιστοποίησης υπερπαραμέτρων, δοκιμάστηκε και η ορθότητα στο και στο σύνολο επικύρωσης, αλλά και σύνολο ελέγχου, η οποία έφτασε το 66,11% και 65,76% αντίστοιχα.

- **6ο Πείραμα - Σεζόν 2017-2018, 2013-2014, 2014-2015, 2015-2016, 2016-2017 για σύνολο εκπαίδευσης-επικύρωσης και 2012-2013 για σύνολο ελέγχου.** Κατά μέσο όρο και στις 5 φορές η υψηλότερη τιμή ορθότητας στο σύνολο επικύρωσης του μοντέλου μας επιτεύχθηκε με τον αλγόριθμο CatBoost Classifier της τάξεως του 66.35%. Στη συνέχεια, κατόπιν βελτιστοποίησης υπερπαραμέτρων, δοκιμάστηκε και η ορθότητα στο και στο σύνολο επικύρωσης, αλλά και σύνολο ελέγχου, η οποία έφτασε το 67,08% και 66,72% αντίστοιχα.

Τα αποτελέσματα παρατίθενται στον παρακάτω πίνακα :

Δοκιμές μέχρι την έυρεση του αλγορίθμου με τη βέλτιστη ακρίβεια .

	Σεζόν Συνόλου Ελέγχου	Ακρίβεια Συνόλου Επικύρωσης	Ακρίβεια Συνόλου Επικύρωσης κατόπιν βελτιστοποίησης υπερπαραμέτρων	Ακρίβεια Συνόλου Ελέγχου	Βέλτιστος Αλγόριθμος
Δοκιμή 1	2017-2018	66,29 %	66,77%	68%	**CatBoost Classifier**
Δοκιμή 2	2016-2017	65,54%	66,04%	65,92%	**CatBoost Classifier**
Δοκιμή 3	2015-2016	66,83%	67,15%	67,22%	**CatBoost Classifier**
Δοκιμή 4	2014-2015	66,61%	67,01%	66,47%	**CatBoost Classifier**
Δοκιμή 5	2013-2014	64,92%	66,11%	65,76%	**CatBoost Classifier**
Δοκιμή 6	2012-2013	66,35%	67,08%	66,72%	**CatBoost Classifier**
Μ/Ο	-	65,92%	66,69%	66,68%	**CatBoost Classifier**

Πίνακας 5.17: *Δοκιμές μέχρι την έυρεση του αλγορίθμου με τη βέλτιστη ορθότητα*

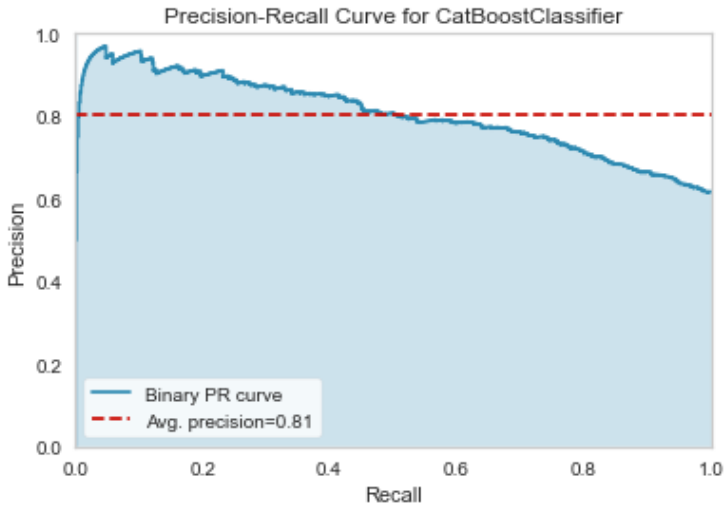
Μπορούμε πλέον με ασφάλεια να πούμε ότι κατά μέσο όρο ο αλγόριθμος CatBoost Classifier στις default υπερπαραμέτρους του επιτυγχάνει ορθότητα 65,92% και στις βελτιστοποιημένες υπερπαραμέτρους του επιτυγχάνει ορθότητα 66,69% στο σύνολο επικύρωσης και 66,68% στο σύνολο ελέγχου. Η αλλαγή που εφαρμόσαμε, με το να μην ανακατεύουμε τα δεδομένα μας και να δίνουμε ανά σεζόν τα σύνολα δεδομένων, είχαν μεγάλο αντίκτυπο στο να καταφέρουμε το μοντέλο μας να γενικεύει σε άγνωστα για αυτό δεδομένα. Η παρούσα εργασία ξεπέρασε το 65% ορθότητας της πλεοψηφίας της παγκόσμιας βιβλιογραφίας και πέτυχε τον αρχικό της σκοπό.

Οι βέλτιστες υπερπαράμετροι του μοντέλου και του εργαλείου pycaret, παρατίθενται στον παρακάτω πίνακα :

```
{'nan_mode': 'Min',
 'eval_metric': 'Logloss',
 'iterations': 230,
 'sampling_frequency': 'PerTree',
 'leaf_estimation_method': 'Newton',
 'grow_policy': 'SymmetricTree',
 'penalties_coefficient': 1,
 'boosting_type': 'Plain',
 'model_shrink_mode': 'Constant',
 'feature_border_type': 'GreedyLogSum',
 'bayesian_matrix_reg': 0.10000000149011612,
 'force_unit_auto_pair_weights': False,
 'l2_leaf_reg': 100,
 'random_strength': 0.5,
 'rsm': 1,
 'boost_from_average': False,
 'model_size_reg': 0.5,
 'pool_metainfo_options': {'tags': {}},
 'subsample': 0.800000011920929,
 'use_best_model': False,
 'class_names': [0, 1],
 'random_seed': 123,
 'depth': 2,
 'posterior_sampling': False,
 'border_count': 254,
 'classes_count': 0,
 'auto_class_weights': 'None',
 'sparse_features_conflict_fraction': 0,
 'leaf_estimation_backtracking': 'AnyImprovement',
 'best_model_min_trees': 1,
 'model_shrink_rate': 0,
 'min_data_in_leaf': 1,
 'loss_function': 'Logloss',
 'learning_rate': 0.029999999329447743,
 'score_function': 'Cosine',
 'task_type': 'CPU',
 'leaf_estimation_iterations': 10,
 'bootstrap_type': 'MVS',
 'max_leaves': 4}
```

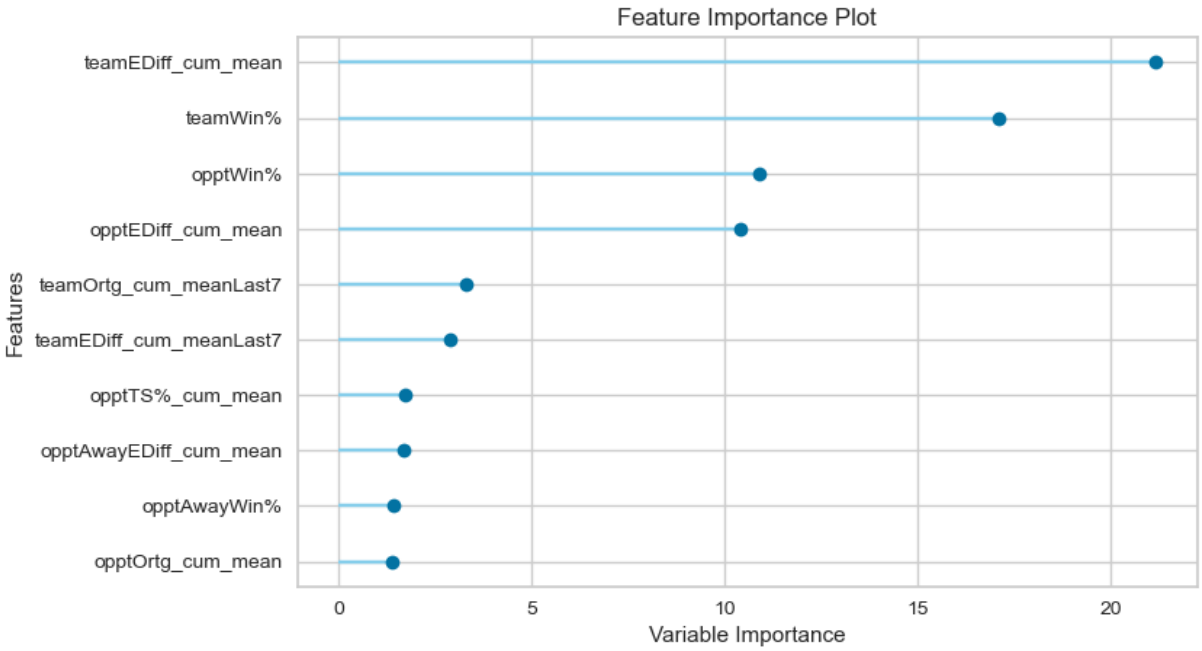

Πίνακας 5.18: *Hyperparameter Tuning*

Παρατίθενται ακόμη διαγράμματα που βοηθούν στην ερμηνεία των αποτελεσμάτων :



Πίνακας 5.19: *Precision-Recall Graph*

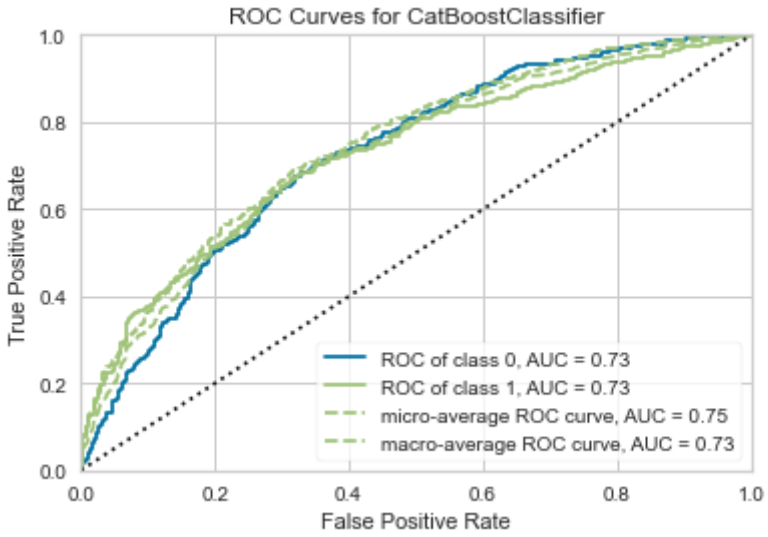
Η καμπύλη ακριβείας-ανάκλησης δείχνει την αντιστάθμιση μεταξύ ακρίβειας και ανάκλησης για διαφορετικό threshold. Μια υψηλή περιοχή κάτω από την καμπύλη αντιπροσωπεύει τόσο υψηλή ανάκληση όσο και υψηλή ακρίβεια, όπου η υψηλή ακρίβεια σχετίζεται με χαμηλό ποσοστό ψευδώς θετικών και η υψηλή ανάκληση σχετίζεται με χαμηλό ψευδώς αρνητικό ποσοστό. Αυτό ακριβώς παρατηρούμε και στην περίπτωση του μοντέλου μας.



Πίνακας 5.20: *Σημαντικότητα Χαρακτηριστικών*

Εδώ παρατηρούμε τα 10 χαρακτηριστικά μιας ομάδας που διαδραμματίζουν το σημαντικότερο ρόλο στην έκβαση ενός αποτελέσματος αγώνα NBA, σύμφωνα πάντα με το βέλτιστο μοντέλο μας. Με μεγάλη μας χαρά παρατηρούμε ότι αρκετά από αυτά είναι κατακσευασμένα από εμάς και δεν υπήρχαν στον αρχικό σύνολο δεδομένων. Συνεπώς οι υποθέσεις μας ήταν λογικές και απέδωσαν.

Παρατίθεται ακόμη, η AUC καμπύλη.



Σχήμα 5.1: *Area Under Curve*

Παρατηρούμε από την παραπάνω γραφική παράσταση ότι η μετρική AUC τόσο για την κατηγορία της νίκης της γηπεδούχου ομάδας, όσο και της νίκης της φιλοξενούμενης είναι στο 73%, πράγμα που σημαίνει την επαρκή διαφορροποίηση μεταξύ των δύο κατηγοριών.

Τέλος παρατίθενται παραδείγματα αγώνων με τα αντίστοιχα στατιστικά τους :

	gmDate	teamAbbr	teamConf	teamDiv	teamRslt	teamDayOff	opptAbbr	opptConf	opptDiv	opptDayOff	seasonID	oppt2P%_cum_mean	opptAway2P%
136	2012-11-17	PHO	West	Pacific	0	1	MIA	East	Southeast	2	2012-2013	0.51346	
142	2012-11-18	DET	East	Central	1	2	BOS	East	Atlantic	1	2012-2013	0.50172	
150	2012-11-19	DAL	West	Southwest	0	2	GS	West	Pacific	1	2012-2013	0.48786	
151	2012-11-19	UTA	West	Northwest	1	2	HOU	West	Southwest	1	2012-2013	0.48624	
152	2012-11-20	PHI	East	Atlantic	1	2	TOR	East	Atlantic	2	2012-2013	0.43629	

Πίνακας 5.21: *Παραδείγματα Πρόβλεψης αποτελεσμάτων 1/2*

in%_Last7	opptWin%_Last7	opptAwayWin%_Last7	teamHomeAvgPtDiff_Last7	opptAwayAvgPtDiff_Last7	teamAvgPtDiff_Last7	opptAvgPtDiff_Last7	Label	Score
0.0	0.714286	0.0	0.0	0.0	-5.285714	6.285714	0	0.6737
0.0	0.714286	0.0	0.0	0.0	-3.857143	4.000000	0	0.6941
0.0	0.428571	0.0	0.0	0.0	-3.285714	-2.857143	1	0.6218
0.0	0.285714	0.0	0.0	0.0	0.714286	-1.714286	1	0.6200
0.0	0.285714	0.0	0.0	0.0	2.428571	-6.714286	1	0.7064

Πίνακας 5.22: *Παραδείγματα Πρόβλεψης αποτελεσμάτων 2/2*

Ο δεύτερος πίνακας αποτελεί συνέχεια του πρώτου και βρίσκεται στα δεξιά του. Σε αυτά τα παραδείγματα, επιλέξαμε τη σεζόν 2012-2013 ως σύνολο ελέγχου και δοκιμάσαμε την ορθότητα του μοντέλου μας σε σχέση με την προβλεπόμενη κατηγορία. Η στήλη του ποσοστού ορθότητας ανά αγώνα είναι η "Score" και εκείνη της κατηγορίας είναι η "Label". Το 0 αφορά τη νίκη της φιλοξενούμενης ομάδας και το 1 της γηπεδούχου. Έστω λοιπόν ότι ζούμε στο παρελθόν και πιο συγκεκριμένα στις 16/11/2012. Στη γραμμή 136 για παράδειγμα αναφέρουμε για τον αγώνα που θα πραγματοποιηθεί την επόμενη μέρα, μεταξύ των Phoenix Suns και των Miami Heat και το μοντέλο μας υποστηρίζει ότι οι Phoenix θα χάσουν τον αγώνα κατά 67,37% πιθανότητα. Πράγμα που όντως επαληθεύτεται αν δει κανείς τη στήλη "teamRslt".

Για την ακρίβεια το τελικό σκορ ήταν το εξής :

Σχήμα 5.2: *Αποτέλεσμα αγώνα μεταξύ Phoenix Suns και Miami Heat*

Ο αγώνας πραγματοποιήθηκε στο Φοίνιξ στην Αριζόνα. Η γηπεδούχος ομάδα εμφανίζεται 2η στους αγώνες του NBA.

5.3 Μοντέλα Βαθιάς Μηχανικής Μάθησης και custom ANN

Στην ενότητα αυτή προσπαθούμε να δοκιμάσουμε ένα τεχνητό νευρωνικό δίκτυο ANN αλλά και αλγορίθμους βαθιάς μηχναικής μάθησης όπως τα LSTM και TCN για την εύρεση ακόμη καλύτερης ορθότητας σε σχέση με τα μοντέλα μηχανικής μάθησης. Και πάλι με τη λογική "trial and error" προσπαθούμε να εντοπίσουμε τη μέγιστη τιμή της ορθότητας του βέλτιστου μοντέλου. Για το ANN χρησιμοποιούμε τη βιβλιοθήκη της python "tensorflow", ενώ για τη βαθιά μηχανική μάθηση τη βιβλιοθήκη "keras". Θα προσπαθήσουμε να ξεκινήσουμε από ένα ποσοστό ορθότητας και σταδιακά θα το αυξάνουμε. Κύριος λόγος πειραματισμού με την τεχνολογία της βαθιάς μηχανικής μάθησης, ακόμη και σε περίπτωη που δεν επιτύχουμε τα επιθυμητά αποτελέσματα, είναι η προσπάθεια εύρεσης λύσης που θα αποτελέσει τη βάση για μελλονική έρευνα και ανάπτυξη της παρούσας εργασίας. Ας δούμε τι καταφέραμε :

5.4 Αποτελέσματα Τεχνητού Νευρωνικού Δικτύου ANN

Αρχικά προεπεξεργαστήκαμε κατάλληλα το σύνολο δεδομένων μας (last_7_plus_27) ώστε όλα τα κατηγορικά χαρακτηριστικά να μετατραπούν σε μορφή one hot encoding, ώστε να δοθούν ως είσοδο στα μοντέλα μας. Έγιναν πάρα πολλές δοκιμές για την εύρεση του βέλτιστου μοντέλου και με τη χρήση άπληστης αναζήτησης. Συνεπώς παρατίθενται η καλύτερη εκδοχή του μοντέλου μας με τις αντίστοιχες τιμές ορθότητας.

Αποτελέσματα ANN

Για το ANN με τα παρακάτω χαρακτηριστικά :

- learning rate = 0.01
- epochs = 300
- batch size = 32
- optimizer = Adam
- layers = 14

Και πιο συγκεκριμένα παρατηρούμε και τις συναρτήσεις ενεργοποίησης και το dropout, με Binary Cross-entropy loss function:

```
model = Sequential()
model.add(Dropout(0.2, input_shape=(487,)))
model.add(Dense(400, activation='relu', input_dim = 487))
model.add(Dropout(0.2, input_shape=(400,)))
model.add(Dense(370, activation='relu'))
model.add(Dropout(0.2, input_shape=(370,)))
model.add(Dense(340, activation='relu'))
model.add(Dropout(0.2, input_shape=(340,)))
model.add(Dense(300, activation='relu'))
model.add(Dropout(0.2, input_shape=(300,)))
model.add(Dense(270, activation='relu'))
model.add(Dropout(0.2, input_shape=(270,)))
model.add(Dense(240, activation='relu'))
model.add(Dropout(0.2, input_shape=(240,)))
model.add(Dense(200, activation='relu'))
model.add(Dropout(0.2, input_shape=(200,)))
model.add(Dense(165, activation='relu'))
model.add(Dropout(0.2, input_shape=(165,)))
model.add(Dense(120, activation='relu'))
model.add(Dropout(0.2, input_shape=(120,)))
model.add(Dense(90, activation='relu'))
model.add(Dropout(0.2, input_shape=(90,)))
model.add(Dense(65, activation='relu'))
model.add(Dropout(0.2, input_shape=(65,)))
model.add(Dense(30, activation='relu'))
model.add(Dropout(0.2, input_shape=(30,)))
model.add(Dense(15, activation='relu'))
model.add(Dense(1, activation='sigmoid'))

model.compile(loss='BinaryCrossentropy', optimizer=opt, metrics=['accuracy'])
```

Σχήμα 5.3: *Αρχιτεκτονική Artificial Neural Network*

Κρατώντας και πάλι 1 σεζόν για σύνολο δεδομένων επικύρωσης, μία για ελέγχου και 4 για εκπαίδευση σε όλους τους δυνατούς συνδυασμούς τους έχουμε:

Σεζόν ελέγχου seasonID_2012-2013, Σεζόν επικύρωσης seasonID_2013-2014
Ορθότητα στο σύνολο επικύρωσης: 62.94, Ορθότητα στο σύνολο ελέγχου: 65.95
Σεζόν ελέγχου seasonID_2012-2013, Σεζόν επικύρωσης seasonID_2014-2015
Ορθότητα στο σύνολο επικύρωσης: 68.56, Ορθότητα στο σύνολο ελέγχου: 67.54
Σεζόν ελέγχου seasonID_2012-2013, Σεζόν επικύρωσης seasonID_2015-2016
Ορθότητα στο σύνολο επικύρωσης: 59.37, Ορθότητα στο σύνολο ελέγχου: 61.38
Σεζόν ελέγχου seasonID_2012-2013, Σεζόν επικύρωσης seasonID_2016-2017
Ορθότητα στο σύνολο επικύρωσης: 58.38, Ορθότητα στο σύνολο ελέγχου: 61.38
Σεζόν ελέγχου seasonID_2012-2013, Σεζόν επικύρωσης seasonID_2017-2018
Ορθότητα στο σύνολο επικύρωσης: 64.00, Ορθότητα στο σύνολο ελέγχου: 65.58
Σεζόν ελέγχου seasonID_2013-2014, Σεζόν επικύρωσης seasonID_2012-2013
Ορθότητα στο σύνολο επικύρωσης: 61.38, Ορθότητα στο σύνολο ελέγχου: 57.08
Σεζόν ελέγχου seasonID_2013-2014, Σεζόν επικύρωσης seasonID_2014-2015
Ορθότητα στο σύνολο επικύρωσης: 64.55, Ορθότητα στο σύνολο ελέγχου: 59.50
Σεζόν ελέγχου seasonID_2013-2014, Σεζόν επικύρωσης seasonID_2015-2016
Ορθότητα στο σύνολο επικύρωσης: 66.17, Ορθότητα στο σύνολο ελέγχου: 63.04
Σεζόν ελέγχου seasonID_2013-2014, Σεζόν επικύρωσης seasonID_2016-2017
Ορθότητα στο σύνολο επικύρωσης: 63.50, Ορθότητα στο σύνολο ελέγχου: 64.06
Σεζόν ελέγχου seasonID_2013-2014, Σεζόν επικύρωσης seasonID_2017-2018
Ορθότητα στο σύνολο επικύρωσης: 64.65, Ορθότητα στο σύνολο ελέγχου: 65.18
Σεζόν ελέγχου seasonID_2014-2015, Σεζόν επικύρωσης seasonID_2012-2013

Ορθότητα στο σύνολο επικύρωσης: 65.11, Ορθότητα στο σύνολο ελέγχου: 68.00
Σεζόν ελέγχου seasonID_2014-2015, Σεζόν επικύρωσης seasonID_2013-2014
Ορθότητα στο σύνολο επικύρωσης: 59.96, Ορθότητα στο σύνολο ελέγχου: 63.06
Σεζόν ελέγχου seasonID_2014-2015, Σεζόν επικύρωσης seasonID_2015-2016
Ορθότητα στο σύνολο επικύρωσης: 67.57, Ορθότητα στο σύνολο ελέγχου: 68.75
Σεζόν ελέγχου seasonID_2014-2015, Σεζόν επικύρωσης seasonID_2016-2017
Ορθότητα στο σύνολο επικύρωσης: 58.38. Ορθότητα στο σύνολο ελέγχου: 57.56
Σεζόν ελέγχου seasonID_2014-2015, Σεζόν επικύρωσης seasonID_2017-2018
Ορθότητα στο σύνολο επικύρωσης: 63.44, Ορθότητα στο σύνολο ελέγχου: 69.96
Σεζόν ελέγχου seasonID_2015-2016, Σεζόν επικύρωσης seasonID_2012-2013
Ορθότητα στο σύνολο επικύρωσης: 61.38, Ορθότητα στο σύνολο ελέγχου: 59.37
Σεζόν ελέγχου seasonID_2015-2016, Σεζόν επικύρωσης seasonID_2013-2014
Ορθότητα στο σύνολο επικύρωσης: 57.08, Ορθότητα στο σύνολο ελέγχου: 59.37
Σεζόν ελέγχου seasonID_2015-2016, Σεζόν επικύρωσης seasonID_2014-2015
Ορθότητα στο σύνολο επικύρωσης: 68.75, Ορθότητα στο σύνολο ελέγχου: 68.03
Σεζόν ελέγχου seasonID_2015-2016, Σεζόν επικύρωσης seasonID_2016-2017
Ορθότητα στο σύνολο επικύρωσης: 58.38, Ορθότητα στο σύνολο ελέγχου: 59.37
Σεζόν ελέγχου seasonID_2015-2016, Σεζόν επικύρωσης seasonID_2017-2018
Ορθότητα στο σύνολο επικύρωσης: 65.86, Ορθότητα στο σύνολο ελέγχου: 68.78
Σεζόν ελέγχου seasonID_2016-2017, Σεζόν επικύρωσης seasonID_2012-2013
Ορθότητα στο σύνολο επικύρωσης: 66.51, Ορθότητα στο σύνολο ελέγχου: 62.20
Σεζόν ελέγχου seasonID_2016-2017, Σεζόν επικύρωσης seasonID_2013-2014
Ορθότητα στο σύνολο επικύρωσης: 57.08, Ορθότητα στο σύνολο ελέγχου: 58.38
Σεζόν ελέγχου seasonID_2016-2017, Σεζόν επικύρωσης seasonID_2014-2015
Ορθότητα στο σύνολο επικύρωσης: 67.54, Ορθότητα στο σύνολο ελέγχου: 63.31
Σεζόν ελέγχου seasonID_2016-2017, Σεζόν επικύρωσης seasonID_2015-2016
Ορθότητα στο σύνολο επικύρωσης: 67.66, Ορθότητα στο σύνολο ελέγχου: 63.87
Σεζόν ελέγχου seasonID_2016-2017, Σεζόν επικύρωσης seasonID_2017-2018
Ορθότητα στο σύνολο επικύρωσης: 63.91, Ορθότητα στο σύνολο ελέγχου: 63.04
Σεζόν ελέγχου seasonID_2017-2018, Σεζόν επικύρωσης seasonID_2012-2013
Ορθότητα στο σύνολο επικύρωσης: 68.66, Ορθότητα στο σύνολο ελέγχου: 65.02
Σεζόν ελέγχου seasonID_2017-2018, Σεζόν επικύρωσης seasonID_2013-2014
Ορθότητα στο σύνολο επικύρωσης: 66.20, Ορθότητα στο σύνολο ελέγχου: 64.84
Σεζόν ελέγχου seasonID_2017-2018, Σεζόν επικύρωσης seasonID_2014-2015
Ορθότητα στο σύνολο επικύρωσης: 57.56, Ορθότητα στο σύνολο ελέγχου: 58.33
Σεζόν ελέγχου seasonID_2017-2018, Σεζόν επικύρωσης seasonID_2015-2016
Ορθότητα στο σύνολο επικύρωσης: 64.77, Ορθότητα στο σύνολο ελέγχου: 62.88
Σεζόν ελέγχου seasonID_2017-2018, Σεζόν επικύρωσης seasonID_2016-2017
Ορθότητα στο σύνολο επικύρωσης: 58.38, Ορθότητα στο σύνολο ελέγχου: 58.33

Συνολικά, κατά μέσο όρο καταφέρνουμε να προβλέψουμε με 59% το αποτέλεσμα ενός αφώνα NBA με τη χρήση ενός ANN στο σύνολο επικύρωσης και 58,83% στο σύνολο ελέγχου. Υπήρχαν σεζόν που το μοντέλο τα πηγαίνει καλύτερα και σεζόν που δεν τα πάει και τόσο καλά. Τα νούμερα που πετυχαίνουμε δεν είναι και τόσο καλά, αυτό ίσως να οφείλεται και

στο μικρό σχετικά πλήθος δεδομένων. Ας δούμε τώρα και τα αποτελέσματα των LSTM και TCN.

5.5 Αποτελέσματα LSTM

Ομοίως και εδώ τα αποτελέσματα δεν ξεπερνούν το 60%. Η αρχιτεκτονική του βέλτιστου μοντέλου LSTM είναι η εξής :

```
model = Sequential()
model.add(LSTM(50, activation='tanh', recurrent_dropout=0.2, return_sequences=True))
model.add(Dropout(0.2))
model.add(LSTM(50, activation='relu', recurrent_dropout=0.2, return_sequences=True))
model.add(Dropout(0.2))
model.add(Dense(1, activation='sigmoid'))

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

Πίνακας 5.23: *Αρχιτεκτονική LSTM*

Με βέλτιστη ορθότητα στο σύνολο ελέγχου το 58.88%. Και εδώ μια λογική εξήγηση του γιατί δεν πετυχαίνουμε τα ιδανικά αποτελέσματα είναι τα πιθανώς λίγα σε πλήθος δεδομένα. Ακόμη αξίζει να επισημανθεί ότι δεν αντιμετωπίσαμε τα δεδομένα ως πολλές παράλληλες ταυτόχρονες χρονοσειρές, αλλά ως μία. Δηλαδή την ίδια ημέρα και ώρα γίνονται πολλοί αγώνες ταυτόχρονα. Εμείς δόσαμε τυχαία σειρά στη σειρά εκκίνησης τους, ενώ θα έπρεπε π.χ. κάθε αγωνιστική να είναι και μια χρονοσειρά για κάθε ομάδα. Αφήνουμε την ιδέα αυτή για μελλοντική έρευνα και ανάπτυξη.

5.6 Αποτελέσματα TCN

Ομοίως και εδώ τα αποτελέσματα δεν ξεπερνούν το 60%. Η αρχιτεκτονική του βέλτιστου μοντέλου TCN είναι η εξής :

```
from tcn import TCN
from keras.models import Input, Model

batch_size, timesteps, input_dim = None, 487, 1

x_train_new = x_train.to_numpy().reshape(x_train.shape[0], x_train.shape[1], 1)
x_test_new = x_test.to_numpy().reshape(x_test.shape[0], x_test.shape[1], 1)
x_val_new = x_val.to_numpy().reshape(x_val.shape[0], x_val.shape[1], 1)

i = Input(batch_shape=(batch_size, timesteps, input_dim))

o = TCN(return_sequences=False)(i)  # The TCN layers are here.
o = Dense(1)(o)

m = Model(inputs=[i], outputs=[o])
m.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])

history = m.fit(x_train_new, y_train, epochs=10, validation_data=(x_val_new, y_val))
```

Πίνακας 5.24: *Αρχιτεκτονική TCN*

Με βέλτιστη ορθότητα στο σύνολο ελέγχου το 58.84%. Επικαλούμαστε τους ίδιους λόγους μη βελτιστοποιημένων πειράματων με τη χρήση LSTM.

Μέρος III

Επίλογος

Κεφάλαιο 6

Συμπεράσματα

Στο κεφάλαιο αυτό συνοψίζονται τα συμπεράσματα της εν λόγω εργασίας. Καταγράφονται ποιες μεθοδολογίες απέδωσαν ως προς την εύρεση της βέλτιστης ορθότητας των μοντέλων μας και ποιες απορρίφθηκαν. Τέλος, περιγράφονται τα 10 σημαντικότερα χαρακτηριστικά ενός αγώνα NBA και πόσα από αυτά κατασκευάστηκαν εξ ολοκλήρου από εμάς, από το αρχικό σύνολο δεδομένων, αλλά και η ολική συνεισφορά της εργασίας.

6.1 Συμπεράσματα

Κατόπιν κατάλληλης προ-επεξεργασίας του αρχικού συνόλου δεδομένων μας, από 44284 αγώνες NBA για τις σεζόν 2012-2018, καταλήξαμε σε 7379 αγώνες, 1230 ανά σεζόν, με μια σεζόν να περιλαμβάνει 1229. Ως αρχικό σημείο της ανάλυσής μας χρησιμοποιήθηκε το σύνολο δεδομένων Dummy, για λόγος που εξηγήθηκαν στο Κεφάλαιο 4, το οποίο σύνολο δεδομένων περιλαμβάνει κάθε box score και για τις 30 ομάδες του NBA, για τη Regular season, με τους αντίστοιχους αθροιστικούς μέσους όρους, που μεταφέρουν στον εκάστοτε τωρινό αγώνα, όλη την απαραίτητη παρελθοντική πληροφορία.

Μετά από αναζήτηση της βέλτιστης τιμής ορθότητας του βέλτιστου μοντέλου μηχανικής μάθησης στα δεδομένα επικύρωσης, καταλήξαμε στο συμπέρασμα ότι η Λογιστική Παλινδρόμηση ξεπερνά κάθε προσσδοκία, αγγίζοντας ορθότητα στο 65,78%.

Στη συνέχεια δοκιμάσαμε να βρούμε το σύνολο δεοδμένων εκείνο, το οποίο αποτελεί παράγωγο του αρχικού και βελτιώνει ακόμη περισσότερο την ορθότητα. Αποδείχθηκε ότι η λογική αφαίρεσης αγωνιστικών, είτε από την αρχή, είτε απότ το τέλος του συνόλου δεδομένων, με στόχο την εύρεση του κατά πόσο επηρεάζουν αρνητικά οι πρώτοι ή οι τελευταίοι αγώνες κάθες σεζόν την πρόβλεψή μας, δεν καταλήξαμε σε ευεργετικά αποτελέσματα και συνεπώς απορρίψαμε αυτή την υπόθεση.

Η λογική που στηριζόταν στη φόρμα μιας ομάδας, τόσο εντός , όσο και εκτός έδρας, αποδεικνύεται να ξεπερνά τη μέχρι τώρα βέλτιστη απόδοση και πιο συγκεκριμένα, η πληροφορία της φόρμας μιας ομάδας στους 7 πιο πρόσφατους αγώνες της, αυξάνει την ορθότητά μας στο 66,14% στο σύνολο επικύρωσης.

Τέλος, με την εισαγωγή 27 νέων δικών μας χαρακτηριστικών, η συνολική ορθότητα στο σύνολο επικύρωσης αγγίζει το 66,47%, ξανά με τη χρήση της Λογιστικής Παλινδρόμησης.

Στο σημείο αυτό, ανακαλύψαμε ότι το μοντέλο μας δεν γενίκευε εξίσου καλά στα διαφορετικά σύνολα ελέγχου. Συνεπώς υπήρχε υπερπροσαρμογή του μοντέλου μας στα δεδομένα

του συνόλου εκπαίδευσης. Ο λόγος ήταν ότι ανακατεύαμε τα δεδομένα, με αποτέλεσμα το μοντέλα μας να εκπαιδεύεται π.χ. στην 20η αγωνιστική μιας σεζόν, ζητώντας του να προβλέψει τη 15η, η οποία και προφανώς προηγείται χρονικά και το αποτέλεσμα της θα ταιριάζει κατά πάσα πιθανότητα με εκείνο της 20ης, καθώς έχει κοινές πληροφορίες από το μέλλον, σε σχέση με το παρελθόν. Αντιλαμβανόμαστε ότι αυτό το γεγονός ελλοχεύει ενός είδους προκατάληψη και δικαιολογή αυτή την υπερπροσαρμογή.

Για να αντιμετωπιστεί αυτό το φαινόμενο, εφαρμόσαμε 5 fold cross validation, χωρίς ανακάτεμα των δεδομένων μας, διατηρώντας κάθε φορά μία από τις 6 σεζόν ως σύνολο ελέγχου. Έτσι στο μοντέλο μας δεν θα επιτραπεί η πρόβλεψη αγώνων μιας σεζόν που προηγούνται χρονικά σε σχέση με αγώνες τις ίδιας σεζόν, στους οποίους το μοντέλο μας έχει ήδη εκπαιδευτεί.

Μετά από πολλαπλά πειράματα με κάθε δυνατό συνδυασμό Train-Test-Validation split της μορφής 4-1-1 των δεδομένων μας ανά σεζόν, καταλήγουμε στο γεγονός ότι το μοντέλο μας επιτυγχάνει 66,69% ορθότητα στο σύνολο επικύρωσης και για τις 6 σεζόν κατά μέσο όρο, κατόπιν βελτιστοποίησης των υπεπαραμέτρων του και 66,68% στα σύνολα ελέγχου κατά μέσο όρο, αποδεικνύοντας, έτσι, μια ιδανική ικανότητα γενίκευσης, με τη χρήση του αλγορίθμου μηχανικής μάθησης "CatBoost Classifier". Ερμηνεύοντας ακόμη τις καμπύλες AUC και Precision-Recall, καταλήγουμε στο συμπέρασμα ότι το μοντέλο μας πραγματοποιεί ικανοποιητική διάκριση μεταξύ των δύο κατηγοριών δυαδικής ταξινόμησης.

Να σημειωθεί εδώ, όπως ήδη αναφέραμε, ότι οι διαφορές μεταξύ των top 5 αλγορίθμων ως προς την τιμή της μετρικής της ορθότητας είναι πολύ μικρές, συνήθως μικρότερες της τάξης του 2%, συνεπώς στόχος της εργασίας δεν ήταν τόσο η εξήγηση του γιατί ένα μοντέλο είναι καλύτερο από ένα άλλο, αλλά η εύρεση του καλύτερου. Εξ άλλου, οποιαδήποτε εικασία σε τόσο μικρές διαφορές θα μπορούσε να μην είναι απόλυτα σωστή και δύσκολα αποδεικνύεται. Επιλέξαμε συνεπώς τον δρόμου του "trial and error".

Ακόμη δοκιμάστηκε η χρήση της Βαθιάς Μηχανικής Μάθησης για τη βελτίωση της ορθότητας σε σχέση με τα μοντέλα Μηχανικής Μάθησης. Η δυσκολία του να αξιοποιήσουμε τα δεδομένα μας ως 30 παράλληλες χρονοσειρές (1 για κάθε ομάδα, για κάθε σεζόν, για όλες τις αγωνιστικές της), σε συνδυασμό με το μη επαρκές πλήθος δεδομένων, οδήγησαν στη μη επίτευξη υψηλότερης τιμής ορθότητας.

6.2 Συνεισφορά της εργασίας

Η εν λόγω εργασία κατάφερε να αναδείξει τον CatBoost Classifier, ως τον βέλτιστο αλγόριθμο μηχανικής μάθησης, στο πρόβλημα στη πρόβλεψης του αποτελέσματος ενός αγώνα NBA, με ποσοστό ορθότητας των προβλέψεών του σε κάθε νέο σύνολο ελέγχου, κατά μέσο όρο στο 66,68%, ξεπερνώντας έτσι αρκετές έρευνες της παγκόσμιας βιβλιογραφίας που πετύχαιναν ποσοστό πρόβλεψης αγώνα στο 65%. Η καινοτομία μας αφορούσε το γεγονός ότι μελετήθηκε ένα σύνολο δεδομένων που δεν είχε προηγουμένως μελετηθεί, χωρίς τη χρήση ατομικών χαρακτηριστικών των παιχτών.

Ανέδειξε ακόμη τα 10 χαρακτηριστικά, τα οποία διαδραματίζουν σημαντικότερο ρόλο στην έκβαση ενός αγώνα NBA. Αυτά, σε φθίνουσα σειρά σημαντικότητας, είναι :

- teamEDiff_cum_mean, δηλαδή ο αθροιστικός μέσος όρος της διαφοράς των πόντων που πετυχαίνει μείον αυτών που δέχεται η γηπεδούχος ομάδα ανά 100 κατοχές, ανά σεζόν, μέχρι τον τωρινό αγώνα
- teamWin%, δηλαδή το ποσοστό νικών της γηπεδούχου ομάδας ανά σεζόν, μέχρι τον τωρινό αγώνα
- opptWin%, δηλαδή το ποσοστό νικών της φιλοξενούμενης ομάδας ανά σεζόν, μέχρι τον τωρινό αγώνα
- opptEDiff_cum_mean, δηλαδή ο αθροιστικός μέσος όρος της διαφορά των πόντων που πετυχαίνει μείον αυτών που δέχεται η φιλοξενούμενη ομάδα ανά 100 κατοχές, ανά σεζόν, μέχρι τον τωρινό αγώνα
- teamOrtg_cum_meanLast7, δηλαδή ο αθροιστικός μέσος όρος των πόντων που πετυχαίνει η γηπεδούχος ομάδα ανά 100 κατοχές, στους τελευταίους 7 αγώνες, ανά σεζόν, μέχρι τον τωρινό αγώνα
- teamEDiff_cum_meanLast7, δηλαδή ο αθροιστικός μέσος όρος της διαφοράς των πόντων που πετυχαίνει μείον αυτών που δέχεται η γηπεδούχος ομάδα ανά 100 κατοχές, ανά σεζόν, μέχρι τον τωρινό αγώνα, στους τελευταίους 7 αγώνες
- opptTS%_cum_mean, δηλαδή ο αθροιστικός μέσος όρος των πραγματικών σουτ που πραγματοποιήθηκαν από τη γηπεδούχο ομάδα
- opptAwayEDiff_cum_mean, δηλαδή ο αθροιστικός μέσος όρος της διαφοράς των πόντων που πετυχαίνει μείον αυτών που δέχεται η φιλοξενούμενη ομάδα ως φιλοξενούμενη, ανά 100 κατοχές, ανά σεζόν, μέχρι τον τωρινό αγώνα
- opptAwayWin%, δηλαδή το ποσοστό νικών της φιλοξενούμενης ομάδας ως φιλοξενούμενη ανά σεζόν, μέχρι τον τωρινό αγώνα
- teamOrtg_cum_mean, δηλαδή ο αθροιστικός μέσος όρος των πόντων που πετυχαίνει η γηπεδούχος ομάδα ανά 100 κατοχές, ανά σεζόν, μέχρι τον τωρινό αγώνα

Με αυτά τα αποτελέσματα η εν λόγω εργασία δύναται δύναται να εξοπλίσει τη φαρέτρα όλων των εμπλεκομένων που απαρτίζουν το άθλημα της καλαθοσφαίρισης, ξεκινώντας από τους παίκτες και τους προπονητές, έως και τους χορηγούς, διευθυντές, αλλά και τους ίδιους τους φιλάθλους, καθώς ενισχύει την προβλεπτική τους ικανότητα, η οποία κάλλιστα θα μπορούσε να εφαρμοστεί και στον στοιχηματισμό. Το γεγονός αυτό αποτελεί, συνεπώς, πόλο έλξης του ενδιαφέροντος πάσης φύσεως εταιρείας στοιχηματισμού αθλητικών γεγονότων, αλλά και ερευνητών στην προσπάθεια επέκτασης αυτής της εργασίας και περεταίρω βελτιστοποίησης της τιμής της ορθότητας των μοντέλων που παρουσιάστηκαν.

Κεφάλαιο 7

Προκλήσεις και Μελλοντικές Επεκτάσεις

7.1 Προκλήσεις

Στην ενότητα αυτή παρουσιάζονται μερικές από τις σημαντικότερες προκλήσεις που κληθήκαμε να αντιμετωπίσουμε στην παρούσα εργασία.

Καταρχάς, πρόκληση αποτελεί από μόνο του το αντικείμενο που πραγματεύεται η εν λόγω εργασία, η πρόβλεψη δηλαδή, ενός τόσο απρόβλεπτου συμβάντος, όπως είναι η εύρεση του αποτελέσματος ενός αγώνα NBA.

Η πρώτη πρόκληση που αντιμετωπίσααμε, ήταν η εύρεση του συνόλου δεδομένων, στο οποίο θα εφαρμόζαμε την ανάλυσή μας. Η δυσκολία πρόσβασης σε κοστοβόρα, αλλά μεγαλύτερα σύνολα δεδομένων και η γενικότερη δυσκολία ανεύρεσής τους αποτέλεσαν έναν από τους παράγοντες επιλογή του συγκεκριμένου συνόλου δεδομένων. Άλλοι βασικοί λόγοι επιλογής του αναφέρονται αναλυτικότερα στο Κεφάλαιο 4.

Μια δεύτερη πρόκληση ήταν η κατάλληλη διαχείριση και προ-επεξεργασία των δεδομένων μας. Δεν βρέθηκε τρόπος να αξιοποιηθούν τα ονοματεπώνυμα των 3 διαιτητών του εκάστοτε αγώνα. Δυσκολία ακόμη δημιούργησε το γεγονός διαχείρησης μικρού σχετικά σε πλήθος αγώνων και παράλληλα περισσότερων από 400 χαρακτηριστικών, μέσω των μοντέλων μηχναικής μάθησης.

Μια ακόμη πρόκληση για εμάς αποτέλεσε η δυσκολία αύξησης της τιμής της ορθότητας των μοντέλων μηχανικής μάθησης, μετά από μεγάλο πλήθος προσπαθειών, για κάθε πιθανό συνδυασμό υπερπαραμέτρων, συνόλων δεδομένων και τεχνικών προ-επεξεργασίας δεδομένων, πράγμα που οδήγησε στην υιοθέτηση της μεθοδολογίας που ακολουθήσαμε στην εν λόγω εργασία. Πρόκληση αποτελεί από μόνο του το γεγονός οργάνωσης και δοκιμής τόσο πολλών πιθανών σεναρίων, ώστε με τη λογική "trial and error" να αποκλείονται σταδιακά τα μη ευεργετικά ως προς την αύξηση της μετρικής της ορθότητας, σενάρια.

Παρόλο που το εργαλείο "pycaret" βοήθησε σημαντικά στην αντιμετώπιση του προβλήματος των πολυπληθών σεναρίων, αποτέλεσε εν τέλη μια περεταίρω πρόκληση, το να μπορέσουμε να εξοικειωθούμε πλήρως με αυτό και τις διάφορες παραμέτρους του, για την εξασφάλιση, όσο το δυνατόν, πιο αξιόπιστων και αντικειμενικών αποτελεσμάτων.

Τέλος, η μεγαλύτερη πρόκληση για εμάς ήταν η εφαρμογή της Βαθιάς Μηχανικής Μάθησης, ενός πιο εξειδικευμένου και εξελιγμένου υποκλάδου της Μηχανικής Μάθησης, καθώς δεν καταφέραμε να αξιοποιήσουμε στο έπακρον τα δεδομένα μας και να τα διαχειριστούμε ως 30 παράλληλες χρονοσειρές (1 για κάθε ομάδα, για κάθε σεζόν, για όλες τις αγωνιστικές της)

ώστε να επιτύχουμε μεγαλύτερη ορθότητα σε σχέση με τα μοντέλα Μηχανικής Μάθησης. Αυτές αποτελούν μερικές από τις κυριότερες προκλήσεις που κληθήκαμε να αντιμετωπίσουμε στην παρούσα εργασία.

7.2 Μελλοντικές Επεκτάσεις

Τόσο η αντιμετώπιση των προκλήσεων, όσο και η επισήμανση νέων προσεγγίσεων και ιδεών αξιοποίησης των δεδομένων, με σημείο εκκίνησης την εν λόγω εργασία, δύναται να οδηγήσουν στην επέκτασή της, οδηγώντας στη μελλοντική έρευνα και ανάπτυξη του συγκεκριμένου ερευνητικού αντικειμένου.

Μερικές από τις βασικότερες ιδέες και μελλοντικές προτάσεις από τη δική μας πλευρά είναι :

- Χρήση πιο εμπλουτισμένου συνόλου δεδομένων, είτε από πλευράς χαρακτηριστικών, είτε συμπεριλαμβάνοντας σεζόν μετά από το 2017-2018 με ιδιαίτερη προσοχή στη σεζόν 2019-2020 η οποία ενδέχεται να επηρεάσει την ιδιότητα της έδρας στα αποτελέσματα.
- Χρήση μοντέλων transfer learning για επαλήθευση των προβλέψεων και σε αγώνες άλλων πρωταθλημάτων π.χ. Ευρωλίγκας ή δοκιμής της μεθοδολογίας μας στην εύρεση των αντίστοιχα καλύτερων 10 χαρακτηριστικών σε εκείνα τα πρωταθλήματα.
- Διαχείρηση των δεδομένων σε μορφή χρονοσειρών γενικότερα, ή ειδικότερα 30 παράλληλων χρονοσειρών (1 για κάθε ομάδα, για κάθε σεζόν, για όλες τις αγωνιστικές της).
- Δοκιμή περεταίρω αλγορίθμων Βαθιάς Μηχανικής Μάθησης, όπως τα RNN και GRU.
- Μια πιο ευφάνταστη ιδέα θα ήταν η μετατροπή των γραμμών των δεδομένων (του κάθε αγώνα δηλαδή) σε διανύσματα εικόνων και εφαρμογή CNN δικτύων για το κομμάτι της δυαδικής ταξινόμησης του αποτελέσματος του εκάστοτε αγώνα.
- Συγχώνευση διαφορετικών συνόλψν δεδομένων, όπως με τις μεταγραφές της κάθε ομάδας. Αν π.χ. ήταν καλές με βάση το ELO ή Ranking του κάθε παίκτη.
- Κατασκευή νέων χαρακτηριστικών στο σύνολο δεδομένων μας όπως :
  - Αν μια ομάδα ταξίδεψε από την Ανατολή στη Δύση, χαρακτηρισικό που μπορεί να συσχετιστεί με την κούραση της ομάδας.
  - Έλεγχος συνεχόμενων ηττών μιας ομάδας, σε περίπτωση που προσπαθεί να κάνει tanking. Ο όρος αυτός επινοήθηκε για να αποτυπώσει την εσκεμένη απώλεια βαθμών από μια ομάδα, ώστε να τερματίσει όσο το δυνατό χαμηλότερα, για να μπορέσει στην επόμενη χρονιά να διεκδικήσει καλύτερη σειρά επιλογής νεαρών ταλέντων και να "ξαναχτίσει" μια δυνατότερη ομάδα.
  - Έλεγχος συνεχόμενων ηττών μιας ομάδας στους τελευταίους N αγώνες.
  - Έλεγχος αν μια ομάδα με μια άλλη παίζουν μεταξύ τους back to back αγώνες.

- Αν οι διαιτητές δύναται να επηρεάσουν το αποτέλεσμα του αγώνα και με ποιό τρόπο.
- Αν οι προπονητές δύναται να επηρεάσουν το αποτέλεσμα του αγώνα και με ποιό τρόπο.
- Ατομικά Χαρακτηριστικά παιχτών.
- Απουσίες και τραυματισμοί παιχτών.
- Το ELO ή Ranking κάθε ομάδας πριν από τον τωρινό αγώνα.
- Χρήση API για εύρεση αντίστοιχων στοιχηματικών αποδόσεων της νίκης ή ήττας της γηπεδούχου ομάδος και πώς μπορούν οι αποδόσεις να επηρεάσουν το αποτέλεσμα. Επέκταση της ανάλυσής μας συμπεριλαμβάνοντας νέα ανάλυση που θα αφορά την εύρεστη της μέγιστης κερδοφοίας, με το οποίο δεν ασχοληθήκαμε στην εν λόγω εργασία.
- Ανάλυση στο αν το χρονικό παράδειγμα επηρεάζει το ποσοστό ορθότητας. Για παράδειγμα αν κάθε χρονιά το μήνα Ιανουάριο για κάποιους λόγους π.χ. μεταγραφές, η απόδοση του μοντέλου μας αυξάνεται ή μειώνεται, ώστε από πλευράς στοιχήματος να εφαρμόζουμε ή αντίστοιχα να αποκλείουμε τις προβλέψεις μας σε αυτά τα χρονικά παράθυρα.
- Πλήθος αγώνων που έπαιξε κάθε ομάδα στην ίδια βδομάδα, πριν τον τωρινό αγώνα.Ένα ακόμη χαρακτηριστικό που σχετίζεται είτε με την κούραση, είτε με την ετοιμότητα μιας ομάδας (αν είναι πολύ μεγάλο ή μικρό αντίστοιχα).

Αυτές είναι μόνο μερικές ιδέες και προτάσεις επέκτασης της εν λόγω εργασίας. Ελπίζουμε να αποτελέσει πηγή έμπνευσης για τους μεταγενέστερους μελετητές και ερευνητές και χρήσιμο εργαλείο για όλους τους συσχετιζόμενους με το χώρο της καλαθοσφάιρισης, όπως αποτέλεσαν και για εμάς οι πηγές που αναφέρουμε ως βιβλιογραφικές αναφορές τις εν λόγω εργασίας.

Βιβλιογραφία

[1] *Τί είναι η δυαδική ταξινόμηση*. https://en.wikipedia.org/wiki/Binary_classification. Ημερομηνία πρόσβασης: 10-10-2022.

[2] *Μετρική επίδοσης μοντέλων ταξινόμησης*. https://medium.com/analytics-vidhya/performance-metrics-classification-model-69a5546b118c. Ημερομηνία πρόσβασης: 10-10-2022.

[3] *Τί είναι ο αλγόριθμος KNN*. https://www.javatpoint.com/k-nearest-neighbor-algorithm-for-machine-learninghttps://www.ibm.com/topics/knn. Ημερομηνία πρόσβασης: 10-10-2022.

[4] *Τί είναι ο αλγόριθμος Naive Bayes*. https://towardsdatascience.com/naive-bayes-classifier-81d512f50a7c. Ημερομηνία πρόσβασης: 10-10-2022.

[5] *Τί είναι ο αλγόριθμος Decision Tree*. https://www.javatpoint.com/machine-learning-decision-tree-classification-algorithm. Ημερομηνία πρόσβασης: 10-10-2022.

[6] *Τί είναι ο αλγόριθμος Support Vector Machine με linear kernel*. https://www.javatpoint.com/machine-learning-support-vector-machine-algorithm. Ημερομηνία πρόσβασης: 10-10-2022.

[7] *Τί είναι ο αλγόριθμος Multilayer Perceptron Classifier*. https://en.wikipedia.org/wiki/Multilayer_perceptron. Ημερομηνία πρόσβασης: 10-10-2022.

[8] *Τί είναι ο αλγόριθμος Random Forest*. https://www.datacamp.com/tutorial/random-forests-classifier-python. Ημερομηνία πρόσβασης: 10-10-2022.

[9] *Τί είναι ο αλγόριθμος Ada Boost Classifier*. https://www.datacamp.com/tutorial/adaboost-classifier-python. Ημερομηνία πρόσβασης: 10-10-2022.

[10] *Η θεωρία πίσω από το Gradient Boosting Classifier*. https://stackabuse.com/gradient-boosting-classifiers-in-python-with-scikit-learn/. Ημερομηνία πρόσβασης: 10-10-2022.

[11] *Τί είναι ο αλγόριθμος Light Gradient Boosting Machine*. https://www.analyticsvidhya.com/blog/2017/06/which-algorithm-takes-the-crown-light-gbm-vs-xgboost/. Ημερομηνία πρόσβασης: 10-10-2022.

[12] *Τί είναι ένα LSTM*. https://intellipaat.com/blog/what-is-lstm/. Ημερομηνία πρόσβασης: 10-10-2022.

[13] *Τί είναι ένα TCN*. https://dida.do/blog/temporal-convolutional-networks-for-sequence-modeling. Ημερομηνία πρόσβασης: 10-10-2022.

[14] *Τί είναι η πρόβλεψη*. https://en.wikipedia.org/wiki/Prediction. Ημερομηνία πρόσβασης: 10-10-2022.

[15] *Delphi method*. https://en.wikipedia.org/wiki/Delphi$_m$*ethod.Hmeromhnaprsbashc* : 10 – 10 – 2022.

[16] *Kalman Filter*. https://en.wikipedia.org/wiki/Extended$_K$*alman$_f$ilter.Hmeromhnaprsbashc* : 10 – 10 – 2022.

[17] *jimmy schneider*. https://www.arcadia938.gr/index.php/diafora/san-simera/18666-san-shmera-jimmy-the-greek-snyder. Ημερομηνία πρόσβασης: 10-10-2022.

[18] *Jeff Sagarin*. https://en.wikipedia.org/wiki/Jeff$_S$*agarin.Hmeromhnaprsbashc* : 10 – 10 – 2022.

[19] *Advanced Football Analytics*. https://en.wikipedia.org/wiki/Advanced$_F$*ootball$_A$nalytics.Hmeromhnaprsb* 10 – 10 – 2022.

[20] *Burke, Brian (2008). "NFL Win Prediction Methodology"*. http://www.advancednflstats.com/2007/09/nfl-win-prediction-methodology.html. Ημερομηνία πρόσβασης: 10-10-2022.

[21] *Ken Pomeroy*. https://en.wikipedia.org/wiki/Ken$_P$*omeroy. Hmeromhnaprsbashc* : 10 – 10 – 2022.

[22] *Τί είναι η Μηχανική Μάθηση*. https://www.ibm.com/cloud/learn/machine-learning. Ημερομηνία πρόσβασης: 10-10-2022.

[23] *Tom Mitchell*. https://towardsai.net/p/machine-learning/what-is-machine-learning-ml-b58162f97ec7. Ημερομηνία πρόσβασης: 10-10-2022.

[24] *Τί είναι η Λογιστική Παλινδρόμηση*. https://www.ibm.com/topics/logistic-regression. Ημερομηνία πρόσβασης: 10-10-2022.

[25] *Hosmer Lemshow*. https://en.wikipedia.org/wiki/HosmerLemeshow$_t$*est.Hmeromhnaprsbashc* : 10 – 10 – 2022.

[26] *Τί είναι ο αλγόριθμος Support Vector Machine με radial kernel*. https://datascienceplus.com/radial-kernel-support-vector-classifier/. Ημερομηνία πρόσβασης: 10-10-2022.

[27] *Τί είναι ο αλγόριθμος Gaussian Process Classifier*. https://machinelearningmastery.com/gaussian-processes-for-classification-with-python/. Ημερομηνία πρόσβασης: 10-10-2022.

[28] *Τί είναι ο αλγόριθμος Ridge Classifier*. https://machinelearningmastery.com/ridge-regression-with-python/. Ημερομηνία πρόσβασης: 10-10-2022.

[29] *Τί είναι ο αλγόριθμος Quadratic Discriminant Analysis*. https://www.datascienceblog.net/post/machine-learning/linear-discriminant-analysis/. Ημερομηνία πρόσβασης: 10-10-2022.

[30] *Τί είναι ο αλγόριθμος Gradient Boosting Classifier*. https://blog.paperspace.com/gradient-boosting-for-classification/. Ημερομηνία πρόσβασης: 10-10-2022.

[31] *Τί είναι ο αλγόριθμος Linear Discriminant Analysis*. https://www.datascienceblog.net/post/machine-learning/linear-discriminant-analysis/. Ημερομηνία πρόσβασης: 10-10-2022.

[32] *Τί είναι ο αλγόριθμος Extra Trees Classifier*. https://machinelearningmastery.com/extra-trees-ensemble-with-python/. Ημερομηνία πρόσβασης: 10-10-2022.

[33] *Τί είναι ο αλγόριθμος Extreme Gradient Boosting*. https://towardsdatascience.com/xgboost-extreme-gradient-boosting-how-to-improve-on-regular-gradient-boosting-5c6acf66c70a. Ημερομηνία πρόσβασης: 10-10-2022.

[34] *Τί είναι ο αλγόριθμος CatBoost Classifier*. https://www.analyticsvidhya.com/blog/2017/08/catboost-automated-categorical-data/. Ημερομηνία πρόσβασης: 10-10-2022.

[35] *Sports data mining technology used in basketball outcome prediction*. https://arrow.tudublin.ie/cgi/viewcontent.cgi?article=1040&context=scschcomdis. Ημερομηνία πρόσβασης: 10-10-2022.

[36] Ge Cheng, Zhenyu Zhang, Moses Kyebambe και Nasser Kimbugwe. *Predicting the Outcome of NBA Playoffs Based on the Maximum Entropy Principle*. *Entropy*, 18(12):450, 2016.

[37] *Neural network quarterbacking*. https://arxiv.org/pdf/1601.00574.pdf. Ημερομηνία πρόσβασης: 10-10-2022.

[38] Rory P. Bunker και Fadi Thabtah. *A machine learning framework for sport result prediction*. *Applied Computing and Informatics*, 15(1):27–33, 2019.

[39] ACSIJ J. και Hamid Rastegari. *A Review of Data Mining Techniques for Result Prediction in Sports*. 2, 2013.

[40] Alan McCabe και Jarrod Trevathan. *Artificial Intelligence in Sports Prediction*. *Fifth International Conference on Information Technology: New Generations (itng 2008)*, 2008.

[41] Krzysztof Trawinski. *A fuzzy classification system for prediction of the results of the basketball games*. *International Conference on Fuzzy Systems*, 2010.

[42] Eftim Zdravevski και Andrea Kulakov. *System for Prediction of the Winner in a Sports Game*. *ICT Innovations 2009*, σελίδα 55–63, 2010.

[43] *A Tutorial on Support Vector Machines for Pattern Recognition*. https://www.di.ens.fr/~mallat/papiers/svmtutorial.pdf. Ημερομηνία πρόσβασης: 10-10-2022.

[44] Lemeshow S Hosmer D. *Applied logistic regression*. Wiley, New York, 2000.

[45] *An Analysis on Bayesian Classifiers*. https://aaai.org/Papers/AAAI/1992/AAAI92-035.pdf. Ημερομηνία πρόσβασης: 10-10-2022.

[46] Schalkoff RJ. *Artificial neural networks*. International ed. McGraw-Hill, New York, 1997.

[47] Dragan Miljkovic, Ljubisa Gajic, Aleksandar Kovacevic και Zora Konjovic. *The use of data mining for basketball matches outcomes prediction*. *IEEE 8th International Symposium on Intelligent Systems and Informatics*, 2010.

[48] James M. Keller, Michael R. Gray και James A. Givens. *A fuzzy K-nearest neighbor algorithm*. *IEEE Transactions on Systems, Man, and Cybernetics*, ΣΜ"-15(4):580–585, 1985.

[49] *Naive (Bayes) at forty: the independence assumption in information retrieval. In: European conference on machine learning*. https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.11.8264&rep=rep1&type=pdf. Ημερομηνία πρόσβασης: 10-10-2022.

[50] *The support vector machine under test*. https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.467.8587&rep=rep1&type=pdf. Ημερομηνία πρόσβασης: 10-10-2022.

[51] J. R. Quinlan. *Induction of decision trees*. *Machine Learning*, 1(1):81–106, 1986.

[52] *Data analytics have made the NBA unrecognizable*. https://qz.com/1104922/data-analytics-have-revolutionized-the-nba/.

[53] Nachi Lieder. *Can Machine-Learning Methods Predict the Outcome of an NBA Game?* *SSRN Electronic Journal*, 2018.

[54] Ge Cheng, Zhenyu Zhang, Moses Kyebambe και Nasser Kimbugwe. *Predicting the Outcome of NBA Playoffs Based on the Maximum Entropy Principle*. *Entropy*, 18(12):450, 2016.

[55] Fadi Thabtah, Li Zhang και Neda Abdelhamid. *NBA Game Result Prediction Using Feature Analysis and Machine Learning*. *Annals of Data Science*, 6(1):103–116, 2019.

[56] Rory P. Bunker και Fadi Thabtah. *A machine learning framework for sport result prediction*. *Applied Computing and Informatics*, 15(1):27–33, 2019.

[57] Ge Cheng, Zhenyu Zhang, Moses Kyebambe και Nasser Kimbugwe. *Predicting the Outcome of NBA Playoffs Based on the Maximum Entropy Principle*. *Entropy*, 18(12):450, 2016.

[58] Bernard Loeffelholz, Earl Bednar και Kenneth W Bauer. *Predicting NBA Games Using Neural Networks*. *Journal of Quantitative Analysis in Sports*, 5(1), 2009.

[59] ACSIJ J. και Hamid Rastegari. *A Review of Data Mining Techniques for Result Prediction in Sports*. 2, 2013.

[60] Nachi Lieder. *Can Machine-Learning Methods Predict the Outcome of an NBA Game?* *SSRN Electronic Journal*, 2018.

[61] Alan McCabe και Jarrod Trevathan. *Artificial Intelligence in Sports Prediction*. *Fifth International Conference on Information Technology: New Generations (itng 2008)*, 2008.

[62] Dragan Miljkovic, Ljubisa Gajic, Aleksandar Kovacevic και Zora Konjovic. *The use of data mining for basketball matches outcomes prediction*. *IEEE 8th International Symposium on Intelligent Systems and Informatics*, 2010.

[63] Krzysztof Trawinski. *A fuzzy classification system for prediction of the results of the basketball games*. *International Conference on Fuzzy Systems*, 2010.

[64] Sushma Jain και Harmandeep Kaur. *Machine learning approaches to predict basketball game outcome*. *2017 3rd International Conference on Advances in Computing,Communication amp; Automation (ICACCA) (Fall)*, 2017.

[65] Eftim Zdravevski και Andrea Kulakov. *System for Prediction of the Winner in a Sports Game*. *ICT Innovations 2009*, σελίδα 55–63, 2010.

[66] Ilker Ali Ozkan. *A Novel Basketball Result Prediction Model Using a Concurrent Neuro-Fuzzy System*. *Applied Artificial Intelligence*, 34(13):1038–1054, 2020.

[67] Faisal Asghar, Mubeen Asif, Muhammad Athar Nadeem, Muhammad Nawaz και Muhammad Idrees. *A NOVEL APPROACH TO RANKING NATIONAL BASKETBALL ASSOCIATION PLAYERS*. 2018.

[68] Luca Grassetti, Ruggero Bellio, Luca Di Gaspero, Giovanni Fonseca και Paolo Vidoni. *An extended regularized adjusted plus-minus analysis for lineup management in basketball using play-by-play data*. *IMA Journal of Management Mathematics*, 32, 2020.

[69] M Utku Özmen. *Marginal contribution of game statistics to probability of winning at different levels of competition in basketball: Evidence from the Euroleague*. *International Journal of Sports Science amp; Coaching*, 11(1):98–107, 2016.

[70] L PageGarritt, W FellinghamGilbert και Reese C. Shane. *Using Box-Scores to Determine a Position's Contribution to Winning Basketball Games*. *Journal of Quantitative Analysis in Sports*, 3:1–18, 2007.

[71] Francisco J. R. Ruiz και Fernando Perez-Cruz. *A generative model for predicting outcomes in college basketball*. *Journal of Quantitative Analysis in Sports*, 11(1), 2015.

[72] Ping Feng Pai, Lan Hung ChangLiao και Kuo Ping Lin. *Analyzing basketball games by a support vector machines with decision tree model*. *Neural Computing and Applications*, 28, 2017.

[73] Mason Chen και Charles Chen. *Data Mining Computing of Predicting NBA 2019-2020 Regular Season MVP Winner*. *2020 International Conference on Advances in Computing and Communication Engineering (ICACCE)*, σελίδες 1–5, 2020.

[74] *nba-prediction*. https://github.com/topics/nba-prediction?o=descs=updated.

[75] *Descriptive and Predictive Analysis of Euroleague Basketball Games and the Wisdom of Basketball Crowds*. https://paperswithcode.com/paper/descriptive-and-predictive-analysis-of. Ημερομηνία πρόσβασης: 10-10-2022.

[76] *4 Insights in 4 Minutes about Kobe Bryant's Career*. https://www.datarobot.com/blog/four-insights-in-four-minutes-about-kobe-bryants-career/. Ημερομηνία πρόσβασης: 10-10-2022.