ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ Μ/Υ
ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ
ΣΧΟΛΗ ΝΑΥΤΙΛΙΑΣ ΚΑΙ ΒΙΟΜΗΧΑΝΙΑΣ
ΤΜΗΜΑΤΟΣ ΒΙΟΜΗΧΑΝΙΚΗΣ ΔΙΟΙΚΗΣΗΣ & ΤΕΧΝΟΛΟΓΙΑΣ
ΔΙΑΠΑΝΕΠΙΣΤΗΜΙΑΚΟ ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ
«ΤΕΧΝΟ-ΟΙΚΟΝΟΜΙΚΑ ΣΥΣΤΗΜΑΤΑ»

ΔΙΕΠΙΣΤΗΜΟΝΙΚΟ – ΔΙΑΠΑΝΕΠΙΣΤΗΜΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ
«ΤΕΧΝΟ-ΟΙΚΟΝΟΜΙΚΑ ΣΥΣΤΗΜΑΤΑ»

ΕΦΑΡΜΟΓΗ ΜΟΝΤΕΛΩΝ ΜΗΧΑΝΙΚΗΣ ΜΑΘΗΣΗΣ
για την πρόβλεψη και αξιολόγηση εγκυρότητας πιστωτικών συναλλαγών σε κινητές συσκευές

ΜΕΤΑΠΤΥΧΙΑΚΗ ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Αντώνη Γιαννάκη Νικόλαος

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

Αθήνα, Φεβρουάριος 2025

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ Μ/Υ
ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ
ΣΧΟΛΗ ΝΑΥΤΙΛΙΑΣ ΚΑΙ ΒΙΟΜΗΧΑΝΙΑΣ
ΤΜΗΜΑΤΟΣ ΒΙΟΜΗΧΑΝΙΚΗΣ ΔΙΟΙΚΗΣΗΣ & ΤΕΧΝΟΛΟΓΙΑΣ
ΔΙΑΠΑΝΕΠΙΣΤΗΜΙΑΚΟ ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ
«ΤΕΧΝΟ-ΟΙΚΟΝΟΜΙΚΑ ΣΥΣΤΗΜΑΤΑ»

ΔΙΕΠΙΣΤΗΜΟΝΙΚΟ – ΔΙΑΠΑΝΕΠΙΣΤΗΜΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ
«ΤΕΧΝΟ-ΟΙΚΟΝΟΜΙΚΑ ΣΥΣΤΗΜΑΤΑ»

ΕΦΑΡΜΟΓΗ ΜΟΝΤΕΛΩΝ ΜΗΧΑΝΙΚΗΣ ΜΑΘΗΣΗΣ
για την πρόβλεψη και αξιολόγηση εγκυρότητας πιστωτικών συναλλαγών σε κινητές συσκευές

ΜΕΤΑΠΤΥΧΙΑΚΗ ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Αντώνη Γιαννάκη Νικόλαος

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

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 21η Φεβρουάριου 2025.

Δημήτριος Ασκούνης Δ.Ε.Π	Ιωάννης Ψαρράς Δ.Ε.Π	Ευάγγελος Μαρινάκης Δ.Ε.Π
Καθηγητής Σχολής ΗΜΜΥ,	Καθηγητής Σχολής ΗΜΜΥ,	Καθηγητής Σχολής ΗΜΜΥ,
ΕΜΠ	ΕΜΠ	ΕΜΠ

Αθήνα, Φεβρουάριος 2025

Αντώνη Γιαννάκη Νικόλαος
Διπλωματούχος/Κάτοχος Μεταπτυχιακού Προγράμματος
Τεχνοοικονομικά Συστήματα της Σχολής Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών Ε.Μ.Π

.....

Νικόλαος Γ. Αντώνη

Αφιερώνεται στην αδερφή μου και στους γονείς μου

Περίληψη

Αναπόσπαστο μέρος των χρηματοοικονομικών δραστηριοτήτων, εκατομμυρίων ανθρώπων σε όλο τον κόσμο αποτελούν πλέον οι συναλλαγές που πραγματοποιούνται μέσω πιστωτικών καρτών. Μόνο στην Ευρώπη, πάνω από τις μισές συναλλαγές πραγματοποιούνται με αυτόν τον τρόπο, γεγονός που φανερώνει πόσο διαδεδομένη είναι η χρήση αυτής της μεθόδου πληρωμής. Με την άνοδο του διαδικτύου και την ευρεία αποδοχή των έξυπνων κινητών τηλεφώνων από το σύνολο σχεδόν του πληθυσμού, αναπτύχθηκε το σύστημα ηλεκτρονικής τραπεζικής το οποίο σύντομα προωθήθηκε στις προσωπικές συσκευές των χρηστών. Με τη δραματική αυτή αύξηση των ηλεκτρονικών πληρωμών, επακόλουθο ήταν να αυξηθεί και ο αριθμός των περιπτώσεων απάτης με τις συνολικές απώλειες μόνο στην γηραιά ήπειρο ανέρχονται σε περίπου 1,8 δισεκατομμύρια ευρώ. Στόχος της παρούσας διπλωματικής εργασίας είναι η μόχλευση των δεδομένων που παρέχονται από την ηλεκτρονική τραπεζική για την ανάπτυξη μοντέλων μηχανικής μάθησης που μπορούν να προβλέπουν και να ανιχνεύουν τις περιπτώσεις οπού μια συναλλαγή θεωρείται απάτη. Για την δημιουργία του μοντέλου χρησιμοποιήθηκε πληθώρα αρχιτεκτονικών μηχανικής και βαθιάς μάθησης με στόχο τη βέλτιστη απόδοση. Για την μελέτη μας στηριχθήκαμε σε ένα σύνολο δεδομένων το οποίο απαιτούσε ειδική μεταχείριση καθώς χαρακτηρίζονταν από έντονη ανισορροπία στα δεδομένα του. Στη συνέχεια εξετάστηκε η δυνατότητα εκτέλεσης των μοντέλων σε IoT συσκευές με στόχο την αποφόρτιση του υπολογιστικού φόρτου μεταφέροντας τις υπολογιστικές διεργασίες από το κέντρο του δικτύου στις παρυφές του. Τέλος, για να προσδιορίσουμε την οικονομική σκοπιμότητα της τοποθέτησης των μοντέλων στα προσωπικά κινητά των χρηστών αντί στα κεντρικά συστήματα των τράπεζων πραγματοποιήθηκε μια σύγκριση ενεργειακού κόστους των δυο επίλογων.

Λέξεις κλειδιά

Πιστωτικές Συναλλαγές, Απάτη, Μηχανική Μάθηση, Ανισορροπία δεδομένων, Ανάλυση ενεργειακού κόστους

Abstract

Credit card transactions have become an integral part of the financial activities of millions of people around the world. In Europe alone, more than half of all transactions are carried out in this way, which shows how widespread the use of this payment method is. With the rise of the internet and the widespread acceptance of smart mobile phones by almost the entire population, the electronic banking system was developed and soon promoted to users' personal devices. With this dramatic increase in electronic payments, the number of fraud cases has also increased, with total losses in the old continent alone amounting to around €1.8 billion. The aim of this thesis is to leverage the data provided by e-banking to develop machine learning models that can predict and detect cases where a transaction is considered fraudulent. A variety of machine and deep learning architectures were used to build the model for optimal performance. For our study we relied on a dataset that required special treatment as it was characterized by a strong imbalance in its data. We then considered the possibility of running the models on IoT devices in order to offload the computational load by moving the computational processes from the center of the network to its edges. Finally, in order to determine the economic feasibility of placing the models on the users' personal mobile phones instead of the banks' central systems, an energy cost comparison of the two epilogs was carried out.

Keywords

Credit Transactions, Fraud, Machine learning, Data imbalance, Energy Cost Analysis

Ευχαριστίες

Με την ολοκλήρωση της διπλωματικής μου εργασίας θα ήθελα να εκφράσω τις θέρμες μου ευχαριστίες σε όσους συνέβαλαν στην εκπόνηση της.

Αρχικά θα ήθελα να απευθύνω τις ευχαριστίες μου στον καθηγητή του Ε.Μ.Π. κ. Δημήτριο Ασκούνη για την εμπιστοσύνη που μου έδειξε καθώς και την δυνατότητα που μου προσέφερε να εκπονήσω την διπλωματική μου εργασία πάνω σε έναν κλάδο στον οποίο ήθελα να εργαστώ και να εμβαθύνω. Ευχαριστώ ιδιαίτερα τον κ. Σωκράτη Νικολαΐδη, υποψήφιο διδάκτορα ΣΗΜΜΥ ΕΜΠ, για την πολύτιμη καθοδήγηση και αμέριστη υπομονή του, χωρίς την οποία η ολοκλήρωση της παρούσας διπλωματικής δεν θα ήταν εφικτή. Επίσης, Θα ήθελα να ευχαριστήσω τους καθηγητές κ. Δήμητρα-Θεοδώρα Κακλαμάνη και κ. Ιάκωβο Βενιέρη, οι οποίοι υπήρξαν αρωγοί καθ' όλη τη διάρκεια των σπουδών μου.

Τέλος, θα ήθελα να ευχαριστήσω την οικογένεια μου η οποία προσέφερε πάντα απλόχερα ότι χρειαζόμουν και ήταν πάντα διπλά μου.

Περιεχόμενα

Κατάλογος Εικόνων

Κατάλογος Πινάκων

Κεφάλαιο 1

Εισαγωγή

Από την αυγή της ανθρωπότητας, οι άνθρωποι είχαν την ανάγκη να αποθηκεύουν και να μεταφέρουν τα αγαθά τους. Με την ανάπτυξη των πολιτισμών, αυτήν την ανάγκη την κάλυψαν οι τράπεζες, οι οποίες επινόησαν ιδιαίτερα αποτελεσματικούς τρόπους για τη φύλαξη των αγαθών αυτών. Από το ανταλλακτικό εμπόριο έως τις σύγχρονες ψηφιακές μορφές συναλλαγών, η ιστορία των τραπεζικών συναλλαγών αποτελεί έναν καθρέφτη της προόδου στον τομέα των οικονομικών σχέσεων και της τεχνολογίας.

Με τη λήξη του Β' Παγκοσμίου Πολέμου, η βιομηχανική και τεχνολογική πρόοδος που σημειώθηκε τις δεκαετίες που ακολούθησαν δεν θα μπορούσε να αφήσει ανεπηρέαστο τον τραπεζικό κλάδο, ο οποίος εξελίχθηκε και βρήκε νέους και πιο αποδοτικούς τρόπους με τους οποίους οι καταναλωτές μπορούσαν να διεκπεραιώσουν τις συναλλαγές τους. Την ίδια χρονική περίοδο παρατηρούμε και την άνθιση της επιστήμης της πληροφορικής, η οποία, σε συνδυασμό με τη διάδοση του διαδικτύου τη δεκαετία του 1990, άλλαξε για πάντα τον τρόπο με τον οποίο βλέπουμε τις τραπεζικές συναλλαγές, καθιστώντας τη δυνατότητα των ανθρώπων να συναλλάσσονται, με οποιονδήποτε στον κόσμο, οπουδήποτε και αν βρίσκονται, εφικτή. Πλέον, το μεγαλύτερο μέρος του πληθυσμού είναι ικανό να πραγματοποιεί συναλλαγές μέσω διαδικτύου, ανεξάρτητα από το μέγεθος της συναλλαγής, από την αγορά ενός απλού καταναλωτικού προϊόντος έως την εξόφληση δανειακών υποχρεώσεων. Οι διαδικτυακές συναλλαγές είναι παντού και χρησιμοποιούνται από όλες τις ηλικιακές ομάδες.

Οι τράπεζες, ήδη από τη δεκαετία του 1990, είχαν αναπτύξει υπηρεσίες μέσω κινητών συσκευών, αλλά η τεχνολογία της εποχής δεν επαρκούσε, ώστε να κάνει αυτές τις υπηρεσίες ελκυστικές για τον μέσο χρήστη. Αυτή η δυσκολία ξεπεράστηκε με την εμφάνιση των έξυπνων κινητών συσκευών (smartphones) τη δεκαετία του 2000, όπου η επεξεργαστική ισχύ των κινητών συσκευών αυξήθηκε, καθιστώντας τες ικανές να διεκπεραιώσουν πολύπλοκες ψηφιακές διεργασίες. Οι τράπεζες ακολουθήσαν τις εξελίξεις και ανέπτυξαν εφαρμογές συμβατές με τα προσωπικά κινητά των χρηστών. Αυτός ο μετασχηματισμός υπήρξε τόσο ραγδαίος, σε βαθμό, όπου σήμερα σχεδόν το

σύνολο των τραπεζικών υπηρεσιών μπορούν να πραγματοποιηθούν μέσω των κινητών συσκευών.

Αυτόν τον νέο τρόπο συναλλαγών παρατήρησαν και εκείνοι που κακόβουλα θέλουνε να επωφεληθούν παράνομα, εξαπατώντας τους χρήστες. Οι τράπεζες, προκειμένου να προστατεύσουν τους καταναλωτές αλλά και να διατηρήσουν την αξιοπιστία τους, έχουν καταφύγει σε αρκετά μέτρα προστασίας έτσι ώστε να ελαχιστοποιηθούν οι πιθανότητες μια τέτοια απάτη να πραγματοποιηθεί.

Τέτοια μετρά προστασίας είναι η χρήση κρυπτογράφησης και πρωτοκόλλων ασφάλειας (π.χ., SSL/TLS) για την προστασία των προσωπικών και οικονομικών πληροφοριών κατά τη μετάδοσή τους μέσω του διαδικτύου. Μια πολύ διαδεδομένη μέθοδος είναι ο έλεγχος ταυτότητας δυο παραγόντων (2FA), η οποία είναι απαραίτητη για την προστασία των λογαριασμών από μη εξουσιοδοτημένη πρόσβαση. Περιλαμβάνει τη χρήση κωδικών που αποστέλλονται μέσω SMS ή εφαρμογών επαλήθευσης, καθώς και βιομετρικών χαρακτηριστικών (π.χ., δακτυλικά αποτυπώματα ή αναγνώριση προσώπου). Φυσικά, πολύ σημαντική παραμένει και η ενημέρωση και ευαισθητοποίηση των χρηστών.

Η ανάπτυξη συστημάτων τεχνητής νοημοσύνης (artificial intelligence, AI) που σημειώνεται τα τελευταία χρόνια δεν θα μπορούσε να μείνει εκτός αυτής της διαδικασίας. Τα επόμενα χρόνια, το AI θα αποτελέσει έναν πολύ σημαντικό σύμμαχο στην μάχη ενάντια σε αυτούς που θα προσπαθούν να παραβιάσουν τον τραπεζικό λογαριασμό ενός ανυποψίαστου πολίτη.

1.1 Αντικείμενο της διπλωματικής εργασίας

Όπως αναφέραμε και στην εισαγωγή, η ανάπτυξη των κινητών συσκευών, η εξέλιξη της τεχνητής νοημοσύνης, καθώς και η πρόσβαση σε μεγάλα σύνολα δεδομένων μας δίνουν τη δυνατότητα να αναλύσουμε και να μελετήσουμε τα χαρακτηριστικά τραπεζικών συναλλαγών. Μέσα από αυτήν την ανάλυση, μπορούμε να διακρίνουμε ποιες συναλλαγές χαρακτηρίζονται ως έγκυρες και ποιες ως απάτες.

Η χρήση μοντέλων μηχανικής μάθησης (machine learning, ML) αλλά και ενός νευρωνικού δικτύου αποτελεί βασικό εργαλείο σε αυτή τη διαδικασία. Ο στόχος είναι να αναπτύξουμε μοντέλα που να ελαχιστοποιούν τα σφάλματα, ώστε οι περιπτώσεις όπου το μοντέλο κατηγοριοποιεί λανθασμένα μια συναλλαγή να τείνουν στο μηδέν.

Στην παρούσα διπλωματική προτείνουμε η εκτέλεση των μοντέλων να πραγματοποιείται στις κινητές συσκευές των χρηστών, παρέχοντας άμεση ανίχνευση τυχόν απατών. Για να είναι βιώσιμη μια τέτοια λύση, θα πρέπει να είναι οικονομικά αποδοτικότερη από την υλοποίηση αντίστοιχων συστημάτων στα κεντρικά συστήματα των τραπεζών.

Ο τελικός στόχος αυτής της διπλωματικής εργασίας είναι να πραγματοποιηθεί μια λεπτομερής ανάλυση του ενεργειακού κόστους των 2 επιλογών. Μέσω αυτής της ανάλυσης, θα αξιολογηθεί η αποδοτικότητα της προτεινόμενης εφαρμογής και θα εντοπιστεί η βέλτιστη λύση για την ανίχνευση και πρόληψη τραπεζικών απατών.

1.2 Οργάνωση τόμου

Η διπλωματική εργασία έχει οργανωθεί σε 7 κεφάλαια.

Το 2ο κεφάλαιο περιλαμβάνει το θεωρητικό υπόβαθρο πάνω στο οποίο βασίστηκε η εργασία. Γίνεται μια σύντομη αναφορά στη μηχανική μάθηση και πραγματοποιείται μια σύγκριση αναμεσά στα μοντέλα μηχανικής μάθησης καθώς και στις τεχνικές υπερδειγματοληψίας / υποδειγματοληψίας.

Στο 3ο κεφάλαιο παρουσιάζονται οι τεχνολογίες που χρησιμοποιήθηκαν για την μοντελοποίηση και την ανάπτυξη του συστήματος. Αυτές είναι η προγραμματιστική γλώσσα Python, η βιβλιοθήκη Scikit-Learn, η εργαλειοθήκη ανοιχτού κώδικα imblearn και το dataset πάνω στο οποίο βασίστηκε η παρούσα διπλωματική.

Στο 4ο κεφάλαιο γίνεται η παρουσίαση του συστήματος. Αναφέρεται η μέθοδος που ακολουθήθηκε για την κατασκευή κάθε μοντέλου καθώς και οι παράμετροι που χρησιμοποιήθηκαν.

Στο 5ο κεφάλαιο παρουσιάζονται οι μετρήσεις που έγιναν για την κατανόηση και αξιολόγηση του συστήματος.

Στο 6ο κεφάλαιο πραγματοποιείται η ανάλυση ενεργειακού κόστους για να ερευνηθεί η πιο συμφέρουσα επιλογή.

Τέλος, στο 7ο κεφάλαιο έχουμε τον επίλογο στον οποίο δίνονται κάποια τελικά συμπεράσματα καθώς και μελλοντική έρευνα που μπορεί να γίνει πάνω στο σύστημα.

Κεφάλαιο 2

Θεωρητικό Υπόβαθρο

Στο δεύτερο κεφάλαιο παρουσιάζεται το θεωρητικό υπόβαθρο στο οποίο βασίζεται η διπλωματική εργασία.

2.1 Μηχανική Μάθηση

Η μηχανική μάθηση αποτελεί κλάδο της τεχνητής νοημοσύνης που στόχο έχει την ανάπτυξη προγραμμάτων που έχουν την ικανότητα να αποκτούν τις πληροφορίες που χρειάζονται ώστε να μπορούν να εκπαιδευτούν [1]. Ο Arthur Samuel στην πρωτοποριακή για την εποχή εργασία του «Some studies in Machine Learning Using the game of checkers» εισήγαγε και βελτίωσε αλγορίθμους που επιτρέπουν στα μοντέλα να χρησιμοποιούν την εμπειρία ώστε να μαθαίνουν και να βελτιώνονται. Για να το επιτύχει αυτό χρησιμοποίησε το παιχνίδι της ντάμας. Το έργο του Arthur Samuel θεωρείται σταθμός στον κλάδο της μηχανικής μάθησης καθώς τέθηκαν οι βάσεις για τεχνικές που χρησιμοποιούνται μέχρι σήμερα. Ειδικότερά στο βιβλίο αυτό ορίζει την μηχανική μάθηση ως το επιστημονικό πεδίο που δίνει τη δυνατότητα στους υπολογιστές να μαθαίνουν χωρίς να είναι ρητά προγραμματισμένοι [2]. Από αυτό συμπεραίνουμε ότι ένας αλγόριθμος μηχανικής μάθησης χρησιμοποιεί τα δεδομένα για να μάθει. Ο Mitchell με τον όρο μάθηση αναφέρεται στον τρόπο όπου ένα πρόγραμμα ή ένας αλγόριθμος χρησιμοποιεί την εμπειρία E για να μάθει, σε σχέση με μια κατηγορία εργασιών T και ένα μέτρο απόδοσης P. Η απόδοσή του στις εργασίες T, όπως αξιολογείται με το P, βελτιώνεται με βάση την εμπειρία E [3].

2.1.1 Επιβλεπόμενη Μάθηση

Επιβλεπόμενοι αλγόριθμοι μηχανικής μάθησης (supervised learning algorithms) χαρακτηρίζονται τα μοντέλα που χρησιμοποιούν και επεξεργάζονται σύνολα δεδομένων που περιέχουν χαρακτηριστικά, με όλα τα δείγματα του συνόλου να αντιστοιχίζονται σε μια ετικέτα ή στόχο. Για παράδειγμα στο γνωστό σύνολο δεδομένων Iris, το οποίο δημιουργήθηκε από τον Βρετανό βιολόγο Anderson Edgar το 1936 και περιλαμβάνει 150 παραδείγματα ιρίδων που ανήκουν σε 3 διαφορετικά είδη, κάθε παράδειγμα ίριδας είναι σημειωμένο με το είδος στο οποίο ανήκει [4].

2.1.2 Μη Επιβλεπομένη Μάθηση

Σε προβλήματα μη επιβλεπομένης μηχανικής μάθησης (unsupervised learning), ο αλγόριθμος προσπαθεί να ανακαλύψει σχέσεις ή μοτίβα χωρίς κάποια εξωτερική καθοδήγηση. Στα δεδομένα δεν υπάρχουν προκαθορισμένες ετικέτες ή στόχοι. Στόχος είναι να εντοπιστούν κρυφές σχέσεις ή να ταξινομηθούν με βάση τις ιδιότητες τους [5].

2.1.3 Ημι-επιβλεπόμενη Μάθηση

Στην ημι-επιβλεπόμενη (semi-supervised) μηχανική μάθηση, τα δεδομένα περιλαμβάνουν τόσο επισημασμένες όσο και μη επισημασμένες εγγραφές. Η διαδικασία επισημείωσης των εγγράφων μπορεί να είναι ιδιαίτερα χρονοβόρα ενώ το οικονομικό κόστος να αποδειχθεί ασύμφορο, κυρίως όταν έχουμε να αναλύσουμε μεγάλα σύνολα δεδομένων. Στον συγκεκριμένο τύπο μάθησης υποστηρίζουμε τα επισημασμένα δεδομένα με ένα πλήθος μη επισημασμένων εγγράφων. Σκοπός είναι να ενισχύσουμε την απόδοση του μοντέλου καθώς δεδομένα που δεν έχουν επισημανθεί είναι πιθανό να παρέχουν πρόσθετες πληροφορίες που βοηθά το μοντέλο να μαθαίνει και να γενικεύει, εντοπίζοντας κρυμμένα μοτίβα ή δομές στα δεδομένα [6].

2.1.4 Ενισχυτική Μάθηση

Στην ενισχυτική μάθηση (reinforcement learning) το σύστημα εκπαιδεύεται και βελτιώνεται μέσω της διαδικασίας δοκιμής και λάθους. Όταν το σύστημα καταφέρνει να εκτελέσει σωστά μια εργασία λαμβάνει θετική ανατροφοδότηση ενώ σε αντίθετη περίπτωση αρνητική. Σε αυτή τη διαδικασία το μοντέλο προσαρμόζεται και ξανά προσπαθεί χρησιμοποιώντας παλαιά και νέα δεδομένα. Είναι σύνηθες οι αλγόριθμοι ενισχυτικής μηχανικής μάθησης να χρησιμοποιούν πρώιμα μοντέλα, τα οποία είναι πιθανό να αποδίδουν χαμηλά, ώστε να συλλέξουν πληροφορίες και να αυξήσουν την αποδοτικότητα τους τις επόμενες φορές που θα ξανά προσπαθήσουν [6].

2.1.5 Αυτοεπιβλεπόμενη μάθηση

Η αυτοεπιβλεπόμενη μάθηση (self-supervised learning), αποτελεί κομμάτι της μη επιβλεπομένης μηχανικής μάθησης. Με αυτήν την τεχνική το σύστημα μαθαίνει από τα δεδομένα που δεν έχουν επισημανθεί με ετικέτες χωρίς να απαιτεί κάποιον εξωτερικό παράγοντα να κατηγοριοποιήσει αυτά τα δεδομένα. Η βασική ιδέα πίσω από την αυτοεπιβλεπόμενη μηχανική μάθηση είναι ότι το ίδιο το σύστημα μπορεί να δημιουργήσει τις ετικέτες από τα δεδομένα εισόδου μέσω της διαδικασίας της αυτοεκπαίδευσης [7].

2.2 Μοντέλα Μηχανικής Μάθησης

Υπάρχουν πολλά μοντέλα μηχανικής μάθησης, το καθένα από τα οποία έχει δικά του χαρακτηριστικά. Σε αυτή την ενότητα θα κάνουμε αναφορά σε κάποια βασικά μοντέλα μηχανικής μάθησης, καθώς και τον τρόπο με τον οποίο αυτά λειτουργούν. Επιπλέον θα δούμε με ποιες μεθόδους προβλέπουν αποτελέσματα με βάση τα δεδομένα που τους παρέχονται. Αρχικά σημαντικό είναι να κάνουμε μια ανάλυση του προβλήματος που θέλουμε να λύσουμε ώστε να μπορέσουμε να επιλέξουμε το κατάλληλο μοντέλο. Η επιλογή του μοντέλου εξαρτάται από διάφορους παράγοντες όπως η φύση των δεδομένων και του προβλήματος, ο χρόνος και οι υπολογιστικοί πόροι που έχουμε στην διάθεση μας.

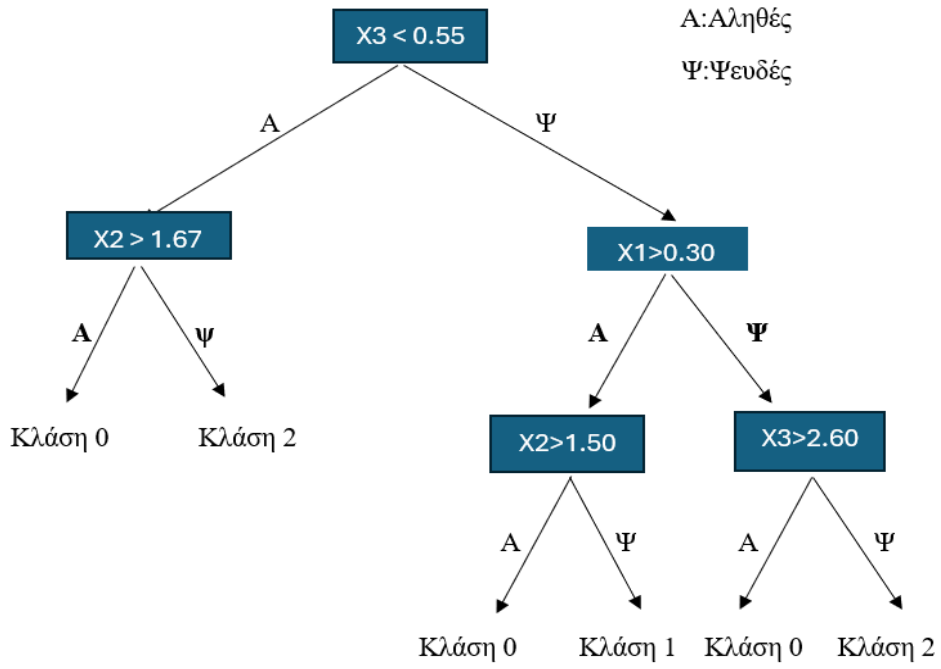
Για τον εντοπισμό του κατάλληλου μοντέλου μπορεί να χρειαστεί να πειραματιστούμε και να αξιολογήσουμε διαφορετικές προσεγγίσεις. Επομένως είναι κρίσιμο να κατανοήσουμε τα χαρακτηριστικά του κάθε μοντέλου αλλά και τους περιορισμούς που αυτά έχουν, ώστε να είμαστε σε θέση να κάνουμε την βέλτιστη επιλογή.

2.2.1 Δέντρα απόφασης

Ένα από τα πιο γνωστά και δημοφιλή μοντέλα μηχανικής μάθησης είναι τα δέντρα αποφάσεων (decision trees). Ένα δέντρο απόφασης μπορούμε να το αποτυπώσουμε ως έναν ακυκλικό γράφο [8].

Το δέντρο απόφασης περιλαμβάνει μια ρίζα (root node) η οποία είναι ο αρχικός κόμβος του δέντρου, το σημείο δηλαδή που ξεκινά η διαδικασία λήψης αποφάσεων και βασίζεται σε κάποιο χαρακτηριστικό η συνθήκη.

Σε κάθε κόμβο απόφασής συναντάμε ένα χαρακτηριστικό (feature) και μια συνθήκη (condition) που χρησιμοποιούνται για τον διαχωρισμό των δεδομένων σε ξεχωριστά υποσύνολα. Τέλος σε ένα δέντρο απόφασης παρατηρούμε τα φύλλα (leaf nodes), τα οποία αποτελούν τους κόμβους πρόβλεψης του δέντρου. Σε αυτούς τους κόμβους βρίσκεται η τελική πρόβλεψη ή η απόφαση που κάνει το δέντρο για μια εγγραφή εισόδου. Υπάρχει μόνο ένα μονοπάτι που μπορεί να μας οδηγήσει από τη ρίζα σε κάποιο φύλλο του δέντρου.



Εικόνα 1: Παράδειγμα δέντρου απόφασης

Η Εικόνα 1 παρουσιάζει ένα υποτυπώδες δέντρο απόφασης που έχει προσαρμοστεί για την ταξινόμηση δεδομένων με τρία χαρακτηριστικά σε τρεις κλάσεις.

Αν για παράδειγμα το δείγμα ελέγχου είναι $x = [0.731.860.10]^T$ ταξινομείται στην κλάση 0.

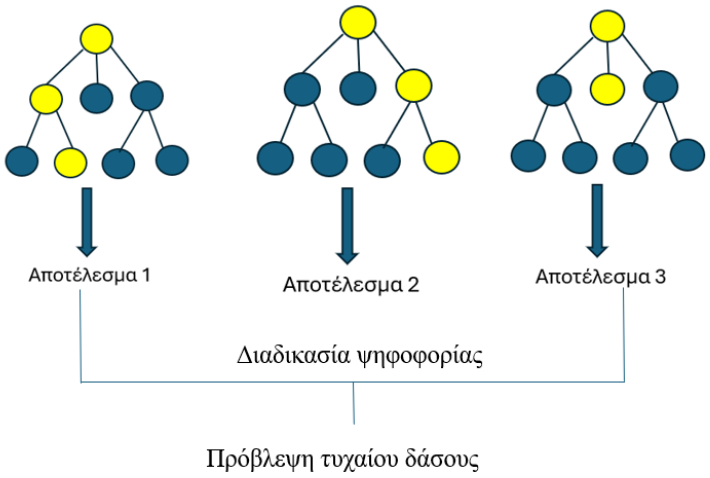
Μπορούμε να εφαρμόσουμε τα δέντρα αποφάσεων σε ένα ευρύ φάσμα περιπτώσεων. Με βάση τα χαρακτηριστικά ασθενών μπορούν να γίνουν ιατρικές προβλέψεις ή να εντοπιστούν απάτες σε οικονομικές συναλλαγές. Διαδεδομένη είναι η χρήση του συγκεκριμένου μοντέλου στο κατακερματισμό του καταναλωτικού κοινού με βάση των αγοραστικών τους συμπεριφορών.

Τα δέντρα αποφάσεων ίσως να μην είναι το πιο ισχυρό εργαλείο που έχουμε στην διάθεσή μας για την εκπαίδευση μοντέλων μηχανικής μάθησης αλλά συχνά προτιμώνται λόγω της ευκολίας κατασκευής τους και της ερμηνευσιμοτητάς τους [8].

2.2.2 Τυχαία Δάση

Τα δέντρα αποφάσεων αποτελούν αλγόριθμους που βασίζονται στη χρήση δέντρων για τη λήψη αποφάσεων. Ο αλγόριθμος τυχαίου δάσους (random forest) δημιουργεί ένα σύνολο από δέντρα αποφάσεων, με το καθένα να εκπαιδεύεται σε ένα τυχαία επιλεγμένο υποσύνολο των δεδομένων εκπαίδευσης, χρησιμοποιώντας τυχαία χαρακτηριστικά. Καθώς τα δέντρα εκπαιδεύονται, χρησιμοποιείται μια διαδικασία ψηφοφορίας για να καθοριστεί η τελική πρόβλεψη. Στην περίπτωση της ταξινόμησης κάθε δέντρο παρέχει μια πρόβλεψη ή ψήφο για την κατηγορία στην οποία ανήκει μια εγγραφή ενώ στην παλινδρόμηση κάθε δέντρο δίνει μια πρόβλεψη για την τιμή. Ύστερα το μοντέλο τυχαίου δάσους καταλήγει στην τελική απόφαση, βασισμένη στην πλειοψηφία των ψήφων αν έχουμε πρόβλημα ταξινόμησης ενώ στην παλινδρόμηση υπολογίζει τον μέσο όρο των προβλέψεων από όλα τα δέντρα [8].Ο αλγόριθμος τυχαίου δάσους έγινε ιδιαίτερα δημοφιλές λόγω των λιγοστών παραμέτρων που

απαιτούνται για την ρύθμιση τους αλλά και για το μεγάλο πλήθος των προβλημάτων πρόβλεψης που μπορούν να αντιμετωπίσουν [9].



Εικόνα 2: Γραφική αναπαράσταση Random Forest

Στην Εικόνα 2 βλέπουμε ένα τυχαίο δάσος , το οποίο έχει δημιουργήσει τρία δέντρα απόφασης. Κάθε δέντρο είναι μια απόφαση κατανομής για τα δεδομένα, που χωρίζει τα δείγματα σε κλάσεις χρησιμοποιώντας σύνολα συνθηκών. Στην κορυφή των σχημάτων υπάρχει ένας κόμβος ρίζας ο οποίος εφαρμόζει μια συνθήκη για να αποφασίσει πως θα διαχωρίσει τα δεδομένα. Κάθε επόμενο επίπεδο ή κόμβος στο δέντρο χρησιμοποιεί διαφορετικές συνθήκες για να συνεχίσει τον διαχωρισμό των δεδομένων. Τα φύλλα κόμβων στο κάτω μέρος κάθε δέντρου περιέχουν τις τελικές κλάσεις για τα δεδομένα που φτάνουν εκεί.

2.2.3 K-Means

Ένα παράδειγμα μη επιβλεπομένης μάθησης αποτελεί ο αλγόριθμος K-Means. Ο αλγόριθμος k-means είναι επαναληπτικός, δηλαδή τα δεδομένα ανακατανέμονται μεταξύ των διάφορων ομάδων (συστάδων) έως ότου έχουμε το αποτέλεσμα που μας ικανοποιεί. Χρησιμοποιώντας αυτόν τον αλγόριθμο, είμαστε σε θέση να επιτύχουμε μεγάλη ομοιότητα μεταξύ των εγγράφων που βρίσκονται στην ιδιά συστάδα ενώ η απόκλιση αυτών που βρίσκονται σε διαφορετικές είναι μεγάλη [10]. Ο αλγόριθμος k-means εφαρμόζεται ως εξής: χρησιμοποιούμε την παράμετρο k για να καθορίσουμε τον αριθμό των κέντρων δηλαδή των συστάδων. Για να τοποθετήσουμε ένα δείγμα σε μια συστάδα, υπολογίζουμε την απόσταση του σημείου από το κέντρο της συστάδας,

χρησιμοποιώντας συνήθως την Ευκλείδεια απόσταση. Στόχος μας είναι να μειώσουμε το άθροισμα των αποστάσεων των σημείων από τα κέντρα των συστάδων, καθώς και το άθροισμα των τετραγώνων των αποστάσεων των σημείων από το κέντρο της συστάδας τους. Η αρχικοποίηση του k γίνεται τυχαία και τοποθετούμε κάθε σημείο στο πλησιέστερο κέντρο. Ο αλγόριθμος σταματάει όταν συμπληρώνεται ο μέγιστος αριθμός επαναλήψεων που έχει οριστεί ή όταν τα κέντρα δεν αλλάζουν σημαντικά [11].

Είναι συχνό φαινόμενο ο αλγόριθμος k-means να παράγει καλά αποτελέσματα αλλά είναι πιθανό να μην είναι αποδοτικός από άποψη χρόνου ενώ μπορεί να μην επιτυγχάνει την βέλτιστη κλιμάκωση [10].

2.2.4 Logistic Regression

Η λογιστική παλινδρόμηση (logistic regression) είναι ένας εποπτευόμενος αλγόριθμος μηχανικής μάθησης που χρησιμοποιείται για την επίλυση προβλημάτων ταξινόμησης, όπου η μεταβλητή στόχος έχει κατηγορική μορφή. Με τον όρο κατηγορική μορφή εννοούμε ότι οι μεταβλητές παίρνουν περιορισμένο αριθμό διακριτών τιμών, δηλαδή οι τιμές τους αναπαριστούν διακριτές ομάδες ή κλάσεις. Στόχος της είναι να δημιουργήσει ένα μαθηματικό μοντέλο που συνδέει τα χαρακτηριστικά του συνόλου δεδομένων με τις κατηγορίες στόχου, παρέχοντας την πιθανότητα να ανήκει ένα νέο δείγμα σε μία από αυτές τις κατηγορίες. Η λογιστική παλινδρόμηση μοιάζει με τη γραμμική παλινδρόμηση η οποία και αυτή χρησιμοποιεί μια γραμμική σχέση για να συνδέσει τα χαρακτηριστικά με την έξοδο. Ωστόσο η λογιστική παλινδρόμηση χρησιμοποιεί την λογιστική συνάρτηση (sigmoid) για να μετατρέπει την έξοδο σε πιθανότητα. Η γραμμική συνάρτηση εισόδου στην περίπτωση της λογιστικής παλινδρόμησης έχει την εξής μορφή:

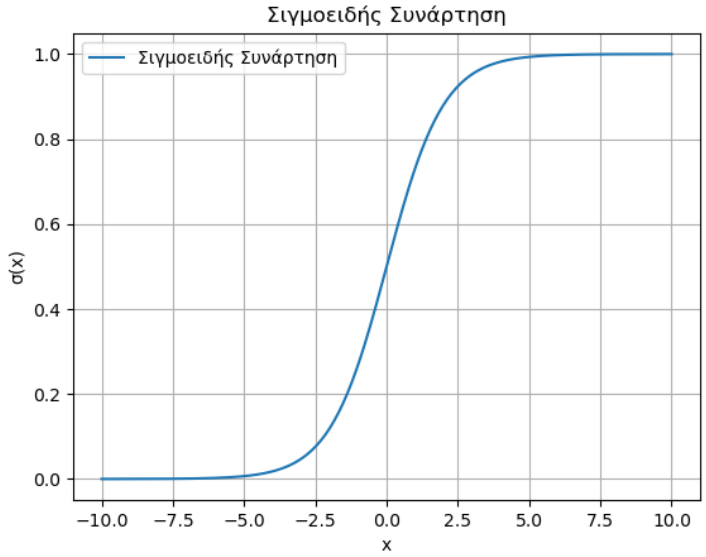
$$z = w_1 x_1 + w_2 x_2 + \cdots + w_n x_n$$

όπου $w_1, w_2 \ldots w_n$ είναι τα βάρη των χαρακτηριστικών.

Η σιγμοειδής συνάρτηση (ή λογιστική συνάρτηση) έχει την εξής μαθηματική μορφή:

$$S(t) = \frac{1}{1+e^{-z}}$$

Η σιγμοειδής συνάρτηση απεικονίζεται γραφικά με την εξής μορφή:



Εικόνα 3: Γραφική αναπαράσταση Σιγμοειδούς συνάρτησης

Επειδή η έξοδος της λογιστικής παλινδρόμησης είναι πιθανότητα (δηλαδή μια τιμή μεταξύ 0 και 1) μπορεί να χρησιμοποιηθεί για ταξινόμηση, αφού μπορούμε να αποφασίσουμε σε ποια κατηγορία ανήκει το δείγμα [12].

Εάν η πιθανότητα είναι μεγαλύτερή από το 0.5 το δείγμα ανήκει στην κατηγορία 1

Εάν η πιθανότητα είναι μικρότερη από το 0.5 το δείγμα ανήκει στην κατηγορία 0

2.2.5 Gradient Boosting Algorithm

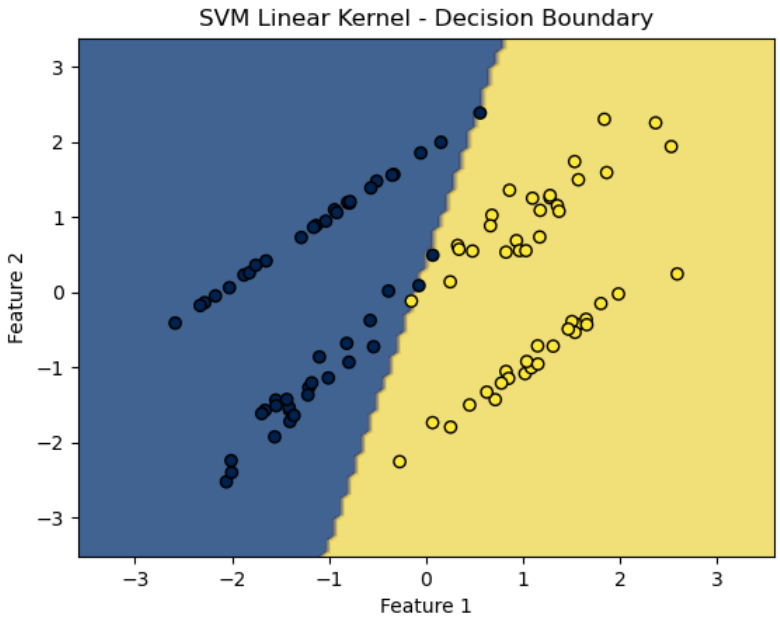
Η προσέγγιση των αλγορίθμων boosting στην μηχανική μάθηση είναι να βελτιώσουν την απόδοση ενός συστήματος, δημιουργώντας ένα ισχυρό μοντέλο που θα προκύψει από τον συνδυασμό πολλών αδύναμων.

Έστω ότι έχουμε ένα σύνολο δεδομένων εκπαίδευσης $D = \{x_i, y_i\}^N$, i=1 οπού το x_i είναι η είσοδος δηλαδή τα χαρακτηριστικά της εγγραφής που εξετάζουμε και y_i η τιμή που επιθυμούμε να προβλέψουμε. Επιδίωξη του Gradient Boosting Algorithm είναι να βρει με ποιον τρόπο συνδέεται η είσοδος x με την έξοδο y, την συνάρτηση δηλαδή που θα προβλέπει με την μεγαλύτερη ακρίβεια. Αυτό το καταφέρνει με το να ελαχιστοποιεί την συνάρτηση απώλειας η οποία μετράει την διαφορά την πρόβλεψης με την πραγματική τιμή.

ο Gradient Boosting Algorithm προσθέτει σταδιακά τις νέες συναρτήσεις που διορθώνουν τα λάθη των προηγούμενων. Το τελικό μοντέλο που δημιουργείται είναι το άθροισμα αυτών των συναρτήσεων τα οποία τα βάρη τους είναι διαφορετικά [13].

2.2.6 Μηχανή Υποστήριξης Διανυσμάτων

Η Μηχανή Υποστήριξης Διανυσμάτων (support vector machine, SVM) παρουσιάστηκε για πρώτη φορά το 1992 από τους Boser, Guyon και Vapnik και είναι μια μέθοδος πρόβλεψης που χρησιμοποιείται για την ταξινόμηση και την παλινδρόμηση, η οποία βασίζεται στις αρχές της μηχανικής μάθησης. Στόχος της είναι να μεγιστοποιήσει την ακρίβεια των προβλέψεων, ενώ ταυτόχρονα αποφεύγει την υπερπροσαρμογή (overfitting) των δεδομένων [14]. Στα προτερήματα του είναι η υψηλή του απόδοση όσο αφορά την γενίκευση αλλά και η δυνατότητα του να εφαρμοστεί σε πολλούς τύπους προβλημάτων ταξινόμησης. Η αποτελεσματικότητα αυτή αντισταθμίζεται από τον υψηλό χρόνο εκπαίδευσης που απαιτείται ιδίως όταν ο όγκος των δεδομένων είναι μεγάλος [11]. Από τα δεδομένα εκπαίδευσης, έχουμε ένα σύνολο εισόδων που αναπαρίστανται από τα διανύσματα x_i, όπου κάθε διάστημα περιλαμβάνει έναν αριθμό χαρακτηριστικών. Κάθε ένα από αυτά τα διανύσματα συνδυάζεται με την αντίστοιχη ετικέτα y_i και συνολικά υπάρχουν m τέτοια ζεύγη $(i = 1 \dots m)$. Τα δεδομένα εκπαίδευσης μπορούν να θεωρηθούν ως επισημασμένα σημεία στον χώρο εισόδου, όπως φαίνεται στην Εικόνα 4. Για δύο κατηγορίες δεδομένων που είναι καλά διαχωρισμένα, ο στόχος της εκμάθησης είναι να βρεθεί μια γραμμή ή ένα υπερεπίπεδο που να τα χωρίζει σωστά, έτσι ώστε να έχει τη μεγαλύτερη απόσταση από τα δεδομένα και των δύο κατηγοριών.



Εικόνα 4: Γραμμικό διάγραμμα απόφασης του SVM: Διαχωριστική γραμμή που διακρίνει δύο κατηγορίες σε ένα 2-διαστατικό χαρακτηριστικό χώρο

.

Ωστόσο, ανάλογα με τη φύση των δεδομένων, υπάρχουν δύο βασικοί τύποι SVM που χρησιμοποιούνται: το γραμμικό SVM και το μη γραμμικό SVM.

Γραμμικό SVM: Η γραμμή απόφασης (υπερεπίπεδο) ορίζεται ως:

$$w^T \mathrm{x} + b = 0$$

Το w είναι το διάνυσμα που καθορίζει τη διεύθυνση κάθετη στο υπερεπίπεδο, το x αντιπροσωπεύει το σημείο δεδομένων και το b είναι ο όρος προκατάληψης. Η συνάρτηση απόφασης για την ταξινόμηση ενός νέου σημείου δεδομένων εκφράζεται μέσω της εξίσωσης:

$F(x) = w^T x + b$

Την κλάση του δείγματος την καθορίζει το πρόσημο της συνάρτησης.

$$Y \geq sign(f(x))$$

Ο στόχος είναι να καθοριστούν τα βέλτιστα w και b που μεγιστοποιούν το περιθώριο μεταξύ των κλάσεων, ενώ ταυτόχρονα μειώνουν τα σφάλματα στην ταξινόμηση.

Μη γραμμικό SVM: Το SVM μπορεί να χρησιμοποιήσει την τεχνική του πυρήνα για να αντιστοιχίσει τον αρχικό χώρο χαρακτηριστικών σε έναν χώρο μεγαλύτερης

διάστασης, όπου τα δεδομένα γίνονται γραμμικά διαχωρίσιμα. Η συνάρτηση απόφασης εκφράζεται ως εξής:

$$f(x) = \sum_{i=1}^{n} \alpha_i y_i K(x, x_i) + b$$

όπου a είναι οι πολλαπλασιαστές Lagrange, y_i οι ετικέτες των κλάσεων, $K(x, x_i)$ η συνάρτηση πυρήνα και b ο όρος προκατάληψης. Όπως και στο γραμμικό SVM, το πρόσημο της συνάρτησης καθορίζει την κλάση του δείγματος [11].

$$\hat{y} = \text{sign}\big(f(\boldsymbol{x})\big)$$

2.3 Αξιολόγηση

2.3.1 Διασταυρούμενη επικύρωση.

Μια πολύ διαδεδομένη τεχνική για την αξιολόγηση μοντέλων είναι η διασταυρουμένη επικύρωση (cross validation), που κατά την χρήση της αξιολογείται το μοντέλο και αποφεύγουμε το πρόβλημα της υπερπροσαρμογης (overfitting). Η βασική ιδέα της διασταυρουμένης επικύρωσης βρίσκεται στον χωρισμό των δεδομένων σε ένα σύνολο υποσυνόλων και η εκπαίδευση και αξιολόγηση του μοντέλου σε διαφορετικούς συνδυασμούς αυτών. Καταφέρνουμε λοιπόν να έχουμε μια πιο λεπτομερή ανάλυση από το να ακολουθούσαμε την συνήθη τακτική του χωρισμού του συνόλου δεδομένων σε σύνολο εκπαίδευσης και σύνολο επικύρωσης.

Διασταυρούμενη επικύρωση k-πτυχών

Η εκδοχή που χρησιμοποιείται πιο συχνά είναι αυτή της διασταυρουμένης επικύρωσής των k-πτυχών (k-fold cross validation).

Η διασταυρουμένη επικύρωση παίρνει ένα σύνολο δεδομένων και το χωρίζει σε k υποσύνολα (folds). Το μοντέλο χρησιμοποιεί k-1 υποσύνολα για να εκπαιδευτεί. Το υποσύνολο που έμεινε εκτός από την διαδικασία εκπαίδευσης χρησιμοποιείται για την αξιολόγηση του μοντέλου. Η διαδικασία θα επαναληφθεί συνολικά k φορές έως

ότου όλα τα υποσύνολα να χρησιμοποιηθούν ακριβώς μια φορά ως υποσύνολο επικύρωσης. Η τελική αξιολόγηση του μοντέλου θα εκφραστεί ως ο μέσος ορός των επιδόσεων του μοντέλου σε όλα τα k υποσύνολα. Με την εφαρμογή αυτής της τεχνικής καταφέρνουμε να εκπαιδεύσουμε το μοντέλο μας στο σύνολο του διαθέσιμου συνόλου δεδομένων, αυξάνοντας την αξιοπιστία του μοντέλου. Συν τοις άλλοις οι πιθανότητες το μοντέλο μας να προσαρμοστεί υπερβολικά στα δεδομένα εισόδου μειώνεται αφού αυτό εκπαιδεύεται και αξιολογείται σε διαφορετικά υποσύνολα. Όλα τα δεδομένα χρησιμοποιούνται σε όλες τις φάσεις τις διαδικασίας. Στα μειονεκτήματα αυτής της τεχνικής μπορούμε να παρατηρήσουμε ότι η διαδικασία της επικύρωσης k-πτυχών μπορεί να αποδειχθεί πολύ χρονοβόρα και με μεγάλο κόστος ιδίως αν τα δεδομένα που καλούμαστε να επεξεργαστούμε χαρακτηρίζονται από μεγάλο όγκο και η δομή τους είναι περιπλοκή [11].

Αφού κατασκευάσουμε τα μοντέλα μηχανικής μάθησης θα πρέπει να τα αξιολογήσουμε για να διαπιστώσουμε την αποδοτικότητα και ανθεκτικότητα τους. Αυτό σημαίνει ότι κρίνονται με βάση το πόσο ικανοποιητικά μπορούν να αποδώσουν ακόμα και όταν υπάρχουν παραλλαγές στα δεδομένα που δέχονται, καθώς και το πόσο μπορούν να εφαρμοστούν σε πραγματικές συνθήκες. Κατά την αξιολόγηση δεν ελέγχουμε μόνο εάν το μοντέλο που έχουμε δημιουργήσει λειτουργεί σωστά αλλά και το κατά πόσο είναι κατάλληλο για την εργασία που θέλουμε να εκτελέσουμε. Στόχος μας είναι το μοντέλο μας να διατηρεί την ίδια αποτελεσματικότητα όχι μόνο στα δεδομένα στα οποία δέχτηκε κατά την εκπαίδευση του αλλά και σε νέα, άγνωστα σε αυτό δεδομένα. Με αυτόν τον τρόπο μπορούμε να εντοπίσουμε και να αποφύγουμε περιπτώσεις υπερπροσαρμογής (overfitting) ή υποπροσαρμογής (underfitting).

Συνηθέστερη μέθοδος αξιολόγησης αποτελεί η μετρική ορθότητας (accuracy), όσο μεγαλύτερος είναι αυτός ο δείκτης τόσο ακριβέστερο θεωρείται το μοντέλο. Σε αυτή την ενότητα θα δούμε γιατί η παραπάνω μετρική δεν θεωρείται πανάκεια και γιατί μοντέλα με πολύ υψηλό δείκτη ορθότητας μπορεί να είναι αναξιόπιστα αλλά και θα αναλύσουμε και μερικούς άλλους δείκτες.

2.3.2 Μετρικές ακρίβειας ταξινόμησης

Για την αξιολόγηση μοντέλων ταξινόμησης ένα πολύ ισχυρό εργαλείο που έχουμε στην διάθεση μας είναι ο πίνακας σύγχυσης (confusion matrix). Ένα μοντέλο ταξινόμησης προβλέπει την κατηγορία στην οποία ανήκουν οι εγγραφές ενός συνόλου δεδομένων με βάση τα χαρακτηριστικά τους. Ο πίνακας σύγχυσης μετρά την αποδοτικότητα ενός μοντέλου συγκρίνοντας τις προβλέψεις που έκανε με τα πραγματικά αποτελέσματα. Αυτή η σύγκριση αποτυπώνεται στον πίνακα σύγχυσης που αποτελείται από c γραμμές και c στήλες.

Δυαδική ταξινόμηση

Όταν το πρόβλημα ταξινόμησης είναι δυαδικό (binary classification) το c είναι ίσο με 2 και οι κλάσεις μπορούν να πάρουν δυο τιμές. Την τιμή «θετική» (positive) και την τιμή «αρνητική» (negative). Ο πίνακας σύγχυσης θα χωρίσει τα δεδομένα σε 4 κατηγορίες.

1. Αληθώς θετικά (True Positives, TP) ο αριθμός των θετικών περιπτώσεων που ο ταξινομητής προέβλεψε σωστά ως θετικές.

2. Αληθώς αρνητικά (True Negatives, TN), ο αριθμός των αρνητικών περιπτώσεων που ο ταξινομητής προέβλεψε σωστά ως αρνητικές.

3. Ψευδώς θετικά (False Positives, FP), ο αριθμός των αρνητικών περιπτώσεων που ο ταξινόμητής προέβλεψε λανθασμένα ως θετικά.

4. Ψευδώς αρνητικά(False Negatives,FN), ο αριθμός των θετικών περιπτώσεων που ο ταξινομητής προέβλεψε λανθασμένα ως αρνητικά.

	Πρόβλεψη		
	Θετική	Αρνητική	
Πραγματικότητα: Θετική	Αληθώς Θετικά (True Positives, TPs):	Ψευδώς Αρνητικά (False Negatives, FNs):	
Πραγματικότητα: Αρνητική	Ψευδώς Θετικά (False Positives, FPs):	Αληθώς Αρνητικά (True Negatives, TNs):	

Πίνακας 1: Πίνακας Σύγχυσης (Confusion Matrix)

Οι σωστές προβλέψεις καταγράφονται στη διαγώνιο του πίνακα σύγχυσης [15].

Οι μετρικές αξιολόγησης που εφαρμόζονται είναι:

Ορθότητα (accuracy): Η ευκολία κατανόησής αυτής της μετρικής την έχει καταστήσει μια από τις πιο διαδεδομένες μετρικές αξιολόγησής για τα μοντέλα μηχανικής μάθησης. Ωστόσο όταν το σύνολο δεδομένων που επιθυμούμε να μελετήσουμε είναι ανισομερές, δηλαδή όταν η ποσότητα των δεδομένων της μια κλάσης είναι πολύ μεγαλύτερη από την άλλη, η μετρική αυτή ενδέχεται να είναι παραπλανητική. Η εξίσωση που περιγράφει την μετρική της ορθότητας είναι:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$$

Η μετρική αυτή λοιπόν υπολογίζει το ποσοστό των σωστών προβλέψεων προς τον συνολικό αριθμό αυτών.

Ακρίβεια (precision): Αποτελεί τον λόγο των θετικών δειγμάτων που κατηγοριοποιήθηκαν σωστά προς των συνολικό αριθμό των παρατηρήσεων που προβλέφθηκαν ως θετικά.

$$Precision = \frac{TP}{TP+FP}$$

Ανάκληση (recall or sensitivity): Αποτελεί τον λόγο των σωστά κατηγοριοποιημένων θετικών δειγμάτων ως προς όλα τα θετικά δείγματα:

$$Recall = \frac{TP}{TP+FN}$$

Η μετρική αξιολόγησης της ανάκλησης είναι κρίσιμη, σε περιπτώσεις οπού το κόστος των ψευδών αρνητικών προβλέψεων είναι υψηλό [11].

F1-score: Το f1-score αποτελεί μια μετρική η οποία συνδυάζει την μετρική της ακρίβειας και αυτής της ανάκλησης λαμβάνοντας τον αρμονικό μέσο όρο τους. Στην περίπτωση που ένα μοντέλο ταξινόμησης Α παρουσιάζει υψηλότερη ανάκληση ενώ ένα μοντέλο ταξινόμησης Β παρουσιάζει υψηλότερη ακρίβεια, η μετρική f1-score μπορεί να προσδιορίσει ποιος από τους δυο ταξινομητές είναι πιο αποδοτικός.

Το F1-score ενός μοντέλου ταξινόμησης υπολογίζεται ως εξής:

$$F1_{score} = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}$$

όπου P η ακρίβεια και R η ανάκληση του μοντέλου ταξινόμησης [16].

2.4 Βαθιά Μάθηση

Τομέα έντονης ενασχόλησης των ερευνητών, στον κλάδο της τεχνητής νοημοσύνης, αποτελεί η βαθιά μάθηση (deep learning) [11].

Σε αντίθεση με την μηχανική μάθηση (machine learning) όπου οι αλγόριθμοι εκπαιδεύονται να εκτελούν εργασίες χωρίς να τους έχουμε ρητά προγραμματίσει χρησιμοποιώντας ένα σύνολο δεδομένων [15]. Η βαθιά μάθηση που αποτελεί υπό κατηγορία της μηχανικής μάθησης χρησιμοποιεί βαθιά νευρωνικά δίκτυα (deep neural networks, DNNs) ώστε να προσομοιάσει τον τρόπο λειτουργίας και την δομή του ανθρωπίνου εγκέφαλου. Η δυνατότητα των μοντέλων βαθιάς μάθησης να τροποποιούν τον εαυτό τους, χωρίς την ανθρώπινη παρέμβαση, τους επιτρέπουν να αναλύουν μεγάλα σύνολα δεδομένων και να επιτυγχάνουν βέλτιστα αποτελέσματα [17]. Μπορούμε λοιπόν να χαρακτηρίσουμε τα μοντέλα βαθιάς μάθησης μια πολύ εξελιγμένη έκδοση των αλγορίθμων μηχανικής μάθησης [18].

Μοντέλα βαθιάς μάθησης χρησιμοποιούνται για την επεξεργασία φυσικής γλώσσας, την αναγνώριση ομιλίας την κατασκευή αυτονόμων αυτοκίνητων, καθώς και σε πλήθος άλλων εφαρμογών [19].

Διαφορές Μηχανικής Μάθησης και Βαθιάς Μάθησης

Μπορεί η βαθιά μάθηση να αποτελεί υποκατηγορία της μηχανικής μάθησης αλλά παρατηρούνται οι παρακάτω διάφορες.

- Κατά την κατασκευή των μοντέλων μηχανικής μάθησης, η προεπεξεργασία δεδομένων αποτελεί απαραίτητη προϋπόθεση. Στην βαθιά μάθηση, αν και η προεπεξεργασια δεδομένων εξακολουθεί να υφίσταται, η εξαγωγή χαρακτηριστικών πραγματοποιείται αυτόματα από το ίδιο το μοντέλο λόγω της αρχιτεκτονικής των βαθιών νευρωνικών δικτύων. Αυτό ελαχιστοποιεί τον χρόνο που απαιτείται για τη δημιουργία χρήσιμων χαρακτηριστικών [11].
- Η δομή των μοντέλων μηχανικής μάθησης μπορεί να χαρακτηριστεί ως σχετικά απλή. Τα τεχνητά νευρωνικά δίκτυα στα οποία βασίζεται η βαθιά μάθηση είναι πολυεπίπεδα και όπως ο ανθρώπινος εγκέφαλος πολύπλοκα.

- Οι αλγόριθμοι βαθιάς μάθησής χρειάζονται πολύ μεγαλύτερα σύνολα δεδομένων, από τους αλγορίθμους μηχανικής μάθησης, λόγω της πολύπλοκης δομής πολλαπλών επίπεδων [18].

2.4.1 Τεχνητά Νευρωνικά Δίκτυα

Η διαδικασία της ανάπτυξης των τεχνητών νευρωνικών δικτυών, για την προσομοίωση των βιολογικών νευρικών συστημάτων, προέκυψε από την ανάγκη μας να δημιουργήσουμε ευφυείς υπολογιστικές μηχανές μέσω της αυτοργάνωσης και της μάθησης [20].

Τα τεχνητά νευρωνικά δίκτυα αποτελούν την πιο διαδεδομένη δομή μοντέλων βαθιάς μάθησης στην βάση των οποίων βρίσκεται ο τεχνητός νευρώνας. Ο τεχνητός νευρώνας (artificial neuron) ή perceptron χρησιμοποιείται με τον ίδιο τρόπο όπως και ο βιολογικός νευρώνας, για την μεταφορά δηλαδή πληροφορίας από τον έναν νευρώνα στον άλλον. Οι νευρώνες στα τεχνητά νευρωνικά δίκτυα χρησιμοποιούν τις εισόδους για να παράγουν έναν γραμμικό συνδυασμό και στη συνέχεια τον διοχετεύουν σε μια μη γραμμική συνάρτηση ενεργοποίησης ώστε να παραχθεί το αποτέλεσμα, δηλαδή η έξοδος [20].

Τα τεχνητά νευρωνικά δίκτυα αποτελούνται από πολλούς τέτοιους νευρώνες οι οποίοι είναι συνδεδεμένοι σε έναν ακυκλικό γράφο. Η έξοδος ενός νευρώνα συνήθως είναι η είσοδος ενός αλλού.

Συνήθως οι νευρώνες αυτοί οργανώνονται σε επίπεδα. Όλα τα τεχνητά νευρωνικά δίκτυα διαθέτουν το επίπεδο εισόδου (input layer) όπου εισάγονται τα δεδομένα, το επίπεδο εξόδου (output layer), στο οποίο εξάγεται το αποτέλεσμα του δικτύου, ενώ όλα τα αλλά επίπεδα χαρακτηρίζονται ως κρυφά επίπεδα [21].

2.5 Προβλήματα Υποεκπαίδευσης και Υπερπροσαρμογής

Ένας από τους πιο σημαντικούς κίνδυνους που αντιμετωπίζουμε κατά την κατασκευή μοντέλων μηχανικής μάθησης είναι η υπερπροσαρμογή (overfitting) και η υπόεκπαίδευση (underfitting). Κατά την κατασκευή μοντέλων μηχανικής μάθησης σκοπός μας είναι να μοντελοποιήσουμε την σχέση που συνδέει το σύνολο χαρακτηριστικών X και το αντίστοιχο σύνολο Y. Αυτή η σχέση μπορεί να ερμηνευτεί

ως μια συνάρτηση (f: X→Y) ή μια κατανομή P(Y |X), που περιγράφει πως κατανέμονται τα στοιχεία του Y με βάση τα χαρακτηριστικά X.

2.5.1 Υπερπροσαρμογή (Overfitting)

Το πρόβλημα της υπερπροσαρμογής το αντιμετωπίζουμε όταν ο αλγόριθμος αντί να κατανοήσει την σχέση που διέπει τις μεταβλητές X και Y, μειώνει το σφάλμα απομνημονεύοντάς τις εγγραφές της εκπαίδευσης. Βασίζεται σε παραδείγματα τα οποία δεν συνεισφέρουν στην πραγματική κατανόηση της ουσιαστικής σχέσης εισόδου- εξόδου.

2.5.2 Υποεκπαίδευση (Underfitting)

Στον αντίποδα, όταν δεν έχουμε αρκετά δεδομένα ή όταν το μοντέλο δεν έχει αρκετή χωρητικότητα ως προς μια δύσκολη διεργασία και η εκπαίδευση του μοντέλου δεν επαρκεί για να αποκαλύψει τη σχέση μεταξύ των χαρακτηριστικών εισόδου X και των αποκρίσεων Y, προκύπτει το πρόβλημα της υποεκπαίδευσης [22].

2.6 Προ επεξεργασία δεδομένων και αντιμετώπιση της Ανισόρροπης Κατανομής

Κατά την κατασκευή μοντέλων μηχανικής μάθησης, συνήθως κάνουμε την παραδοχή ότι ο αριθμός των δεδομένων στις κατηγορίες που θέλουμε να επεξεργαστούμε είναι ισόποσα κατανεμημένος. Ωστόσο, στην πραγματικότητα αυτό είναι πολύ σπάνιο. Συχνά παρατηρούμε ότι τα δεδομένα μιας ή περισσοτέρων κατηγοριών υπερτερούν σημαντικά των άλλων. Ως αποτέλεσμα το μοντέλο που θα κατασκευάσουμε τείνει να μεροληπτεί υπερ της κυρίαρχης κατηγορίας, ακόμα και αν το μοντέλο τεχνικά είναι άρτιο. Είναι συχνό φαινόμενο τα δεδομένα που μειονεκτούνε αριθμητικά να παρέχουν χρήσιμη γνώση για το μοντέλο μας. Για αυτό τον λόγο η εκμάθηση μοντέλων μηχανικής μάθησης από σύνολα δεδομένων που δεν είναι ομοιόμορφα κατανεμημένα αποτελεί τομέα έντονης ερευνάς. Ο τομέας αυτός συνηθίζεται να αποκαλείται εκμάθηση από ανισόρροπα δεδομένα (learning from imbalanced data).

Σε αυτό το κεφάλαιο, εξετάζουμε το πως μπορούμε να αντιμετωπίσουμε το πρόβλημα της ανισορροπίας δεδομένων και αναλύουμε τεχνικές και μεθόδους με τις οποίες μπορούμε να ισορροπήσουμε τον αριθμό των δειγμάτων και να αποφύγουμε την μεροληψία αξιοποιώντας όλα μας τα δεδομένα ώστε να επιτύχουμε καλύτερη γενίκευση και ακρίβεια στις προβλέψεις του μοντέλου μας [22].

2.6.1 No Under/Over Sampling

Όπως αναφέραμε και στην εισαγωγή του κεφαλαίου, πριν την κατασκευή μοντέλων μηχανικής μάθησης θα πρέπει να διαβεβαιώσουμε ότι τα δεδομένα τα οποία θα εισάγουμε στο μοντέλο είναι ισορροπημένα. Ωστόσο η εφαρμογή τεχνικών για την εξισορρόπηση των δεδομένων δεν είναι πανάκεια. Όταν δεν εφαρμόζουμε κάποια τεχνική εξισορρόπησης δεδομένων αλλά χρησιμοποιούμε ακατέργαστα δεδομένα ονομάζουμε την διαδικασία χωρίς υπερ/υποδειγματοληψίας . Αυτή η διαδικασία μπορεί να φανεί ιδιαίτερα αποδοτική καθώς κατά την διαδικασία της εξισορρόπησης είναι πιθανό είτε να χάσουμε δεδομένα είτε να εισάγουμε ψευδοδεδομένα και να επαναληφθούν υπάρχουσες εγγραφές.

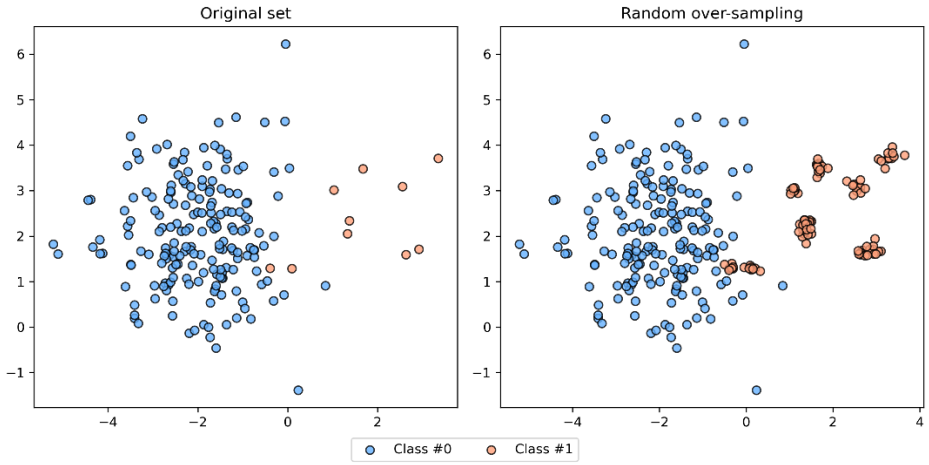
Πριν από την εφαρμογή οποιασδήποτε τεχνικής επεξεργασίας δεδομένων είναι απαραίτητο να διαχωριστεί το αρχικό σύνολο δεδομένων σε σύνολο εκπαίδευσης, επικύρωσης και ελέγχου.

2.6.2 Random Oversampling

Η τυχαία υπερδειγματοληψία (random oversampling) αποτελεί μια απλή αλλά ιδιαίτερα αποτελεσματική μέθοδο για την εξισορρόπηση δεδομένων. Χρησιμοποίει την τεχνική της επαναδειγματοληψίας, δηλαδή επιλέγονται τυχαία εγγραφές από την μειοψηφούσα κατηγορία οι οποίες αντιγράφονται και προστίθενται στο σύνολο δεδομένων που χρησιμοποιείται για την εκπαίδευση του μοντέλου. Η επιλογή των δεδομένων που θα αντιγράφουν θα πρέπει να γίνει από το σύνολο εκπαίδευσης και όχι από το αρχικό σύνολο δεδομένων. Για αυτό η διαδικασία θα πρέπει να εκτελεστεί μετά τον διαχωρισμό του αρχικού συνόλου δεδομένων σε σύνολο εκπαίδευσης και σύνολο επικύρωσης. Σε αντίθετη περίπτωση η τυχαιότητα της επιλογής θα επηρεαστεί σε σημείο που το σύνολο των δεδομένων που θα δημιουργηθεί δεν θα αντικατοπτρίζει την πραγματική κατανομή των δεδομένων. Επιπλέον κάθε φορά που

μια εγγραφή επιλέγεται τυχαία αυτή αντιγράφεται και επανατοποθετείτε στο σύνολο δεδομένων ώστε να μπορεί να επιλεγεί ξανά σε επόμενη δειγματοληψία. Αυτή η συνθήκη διασφαλίζει ότι υπάρχουν αρκετές εγγραφές ώστε η κατανομή των δεδομένων να εξισορροπηθεί.

Σε αντίθετη περίπτωση δηλαδή αν κάθε στοιχείο της κατηγορίας που μειοψηφούσε επιλεγόταν και αντιγραφόταν μια φορά θα ελλόχευε ο κίνδυνος το σύνολο δεδομένων να παραμείνει ανισόρροπο ακόμα και μετά την τεχνική του random oversampling.



Εικόνα 5: Λειτουργία του Random Oversampling

Η εικόνα 5 παρουσιάζει το σύνολο δεδομένων πριν και μετά την εφαρμογή της τυχαίας υπερδειγματοληψίας [23].

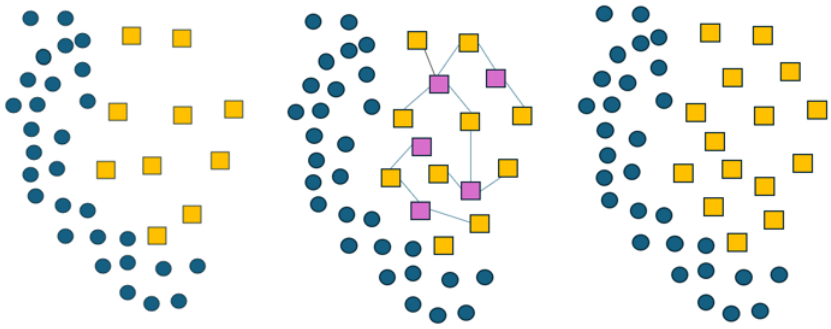
2.6.3 Weight Class

Μια ακόμη τεχνική για να αντιμετωπίσουμε την ανισορροπία μεταξύ των κλάσεων είναι η χρήση βαρών (class weights). Αυτή τη τεχνική δίνει μεγαλύτερη βαρύτητα στην μειοψηφούσα κατηγορία και έτσι επιτυγχάνουμε να μειώσουμε την προκατάληψη του μοντέλου προς την κλάση με τις περισσότερες παρατηρήσεις. Θα πρέπει να είμαστε προσεχτικοί με το ποσό υψηλά βάρη θα ορίσουμε στην μειοψηφούσα κλάση αφού υπάρχει ο κίνδυνος να υπάρξει προκατάληψη προς αυτήν. Με αλλά λόγια η συνάρτηση κόστους του αλγορίθμου αυξάνει σημαντικά το κόστος όταν γίνει μια λάθος πρόβλεψη προς την μειοψηφούσα κατηγορία (στην οποία έχουμε προσθέσει τα βάρη) και λιγότερο όταν το λάθος έχει γίνει στην κυρίαρχη κλάση [24].

2.6.4 SMOTE

Η μέθοδος SMOTE (Synthetic Minority Over-sampling Technique) αποτελεί και αυτή μια τεχνική υπερδειγματοληψίας όπως και η τυχαία υπερδειγματοληψία. Σε διαφορά με την τυχαία υπερδειγματοληψία αντί απλά να επιλεγεί τυχαία δείγματα και να τα προσθέτει στο σύνολο εκπαίδευσης, το SMOTE δημιουργεί νέα συνθετικά δείγματα.

Αρχικά καθορίζεται ο αριθμός των δειγμάτων που θα πρέπει να δημιουργηθεί. Αυτός είναι ένας ακέραιος αριθμός N ο οποίος θα επιτύχει μια 1-1 αναλογία των κλάσεων. Στη συνέχεια επιλέγεται τυχαία μια εγγραφή από την κατηγορία που μειοψηφεί από το σύνολο εκπαίδευσης. Συνεχίζει εντοπίζοντας τους k πλησιέστερους γείτονες του, συνήθως 5 στον αριθμό. Τέλος υπολογίζεται η διαφορά μεταξύ της εγγραφής που επιλέχθηκε και των k πλησιέστερων γειτόνων. Αυτήν η διαφορά πολλαπλασιάζεται με έναν αριθμό ο οποίος βρίσκεται μεταξύ 0-1. Το αποτέλεσμα που βρίσκουμε το προσθέτουμε στην αρχική εγγραφή. Με αυτόν τον τρόπο καταφέρνουμε να δημιουργήσει μια νέα εγγραφή η οποία βρίσκεται στην ευθεία μεταξύ των δυο γειτόνων. Αν τα χαρακτηριστικά είναι κατηγορικά μια από τις δυο τιμές επιλέγεται τυχαία [25].



Εικόνα 6: Οπτική απεικόνιση του Smote

2.6.5 SMOTE Tomek

Υβριδική μέθοδος δειγματοληψίας αποτελεί η SMOTE and Tomek, η οποία συνδυάζει την τεχνική SMOTE και Tomek Links. Βελτιώνει την κλασική προσέγγιση SMOTE αφού υπάρχει η πιθανότητα, τα νέα δείγματα που θα δημιουργηθούν να επικαλύπτουν το ένα το άλλο και να οδηγηθούμε σε υπερεκπαίδευση επειδή τα όρια μεταξύ των δειγμάτων δυσκολεύονται να καθοριστούν. Για την αντιμετωπίσει αυτού του προβλήματος χρησιμοποιείται η τεχνική Tomek Links ώστε να καθαρίσει τον θόρυβο

που δημιουργείτε μετά την δημιουργία συνθετικών δειγμάτων από το SMOTE. Χρησιμοποιούμε δηλαδή την ιδιότητα του SMOTE να παράγει συνθετικά δεδομένα για την μειοψηφική κλάση με την ικανότητα των Tomek Links να αφαιρεί τις θορυβώδης εγγραφές στα αρχικά δείγματα αλλά και στα παραγόμενα μέσω του SMOTE δεδομένα. Παρατηρείται μεγαλύτερη ακρίβεια όταν συνδυάζονται αυτές οι τεχνικές πάρα όταν χρησιμοποιούντα ξεχωριστά. Επιλέγουμε τυχαία εγγραφές από την μειοψηφούσα κλάση. Υπολογίζουμε την απόσταση των εγγράφων αυτών και από τους k πλησιέστερους γείτονες και πολλαπλασιάζουμε την διαφορά με έναν αριθμό, ο οποίος βρίσκεται μεταξύ του 0 και του 1. Το αποτέλεσμα το προσθέτουμε στην μειοψηφούσα κλάση. Επαναλαμβάνουμε μέχρι να επιτύχουμε την επιθυμητή αναλογία. Από την κατηγορία με τα περισσότερα δεδομένα επιλέγουμε τυχαία κάποιες εγγραφές. Αν το πιο κοντινό γειτονικό δείγμα ανήκει στην μειοψηφούσα κατηγορία, αφαιρούνται και τα δυο. Δηλαδή αφαιρούμε δείγματα τα οποία είναι υπερβολικά παρόμοια αλλά ανήκουν σε διαφορετικές κλάσεις [26].

2.7. Ανάλυση Ενεργειακού κόστους

Κατά την ανάλυση ενεργειακού κόστους υπολογίζουμε το ενεργειακό κόστος που απαιτείται για να εκτελεστεί μια συγκεκριμένη δραστηριότητα. Για αυτήν την ανάλυση εξετάζουμε την ενέργεια που απαιτείται αλλά και το πόσο κοστίζει αυτή η ενέργεια. Μια τέτοια ανάλυση μας βοηθά να διακρίνουμε ποιες επιλογές είναι πιο οικονομικές αλλά και να εξοικονομήσουμε πόρους και χρήματα [27].

Κεφάλαιο 3

Τεχνολογίες – Εργαλεία

Σε αυτό το κεφάλαιο γίνεται μια σύντομη παρουσίαση των τεχνολογιών που χρησιμοποιήθηκαν για την εκπόνηση της διπλωματικής εργασίας.

3.1 Python

Η python άρχισε να αναπτύσσεται στα τέλη της δεκαετίας του 1980 και άρχισε να εφαρμόζεται στα τέλη του 1989, από τον κύριο συγγραφέα της, τον Ολλανδό Guido van Rossum. Πηρέ το όνομα της από τους Monty Python αφού ο Guido van Rossum ήταν μεγάλος θαυμαστής τους την περίοδο που δημιουργούσε την γλώσσα και ήθελε να δώσει σε αυτήν την νέα γλώσσα ένα όνομα το οποίο θα ήταν σύντομο, μοναδικό και ελαφρώς μυστηριώδες.

Η python είναι μια γλώσσα προγραμματισμού, υψηλού επιπέδου και χαρακτηρίζεται ως γενικής χρήσης, μπορεί δηλαδή να χρησιμοποιηθεί στην επίλυση πολλών διαφορετικών προβλημάτων. Το σύστημα που διαθέτει η python είναι δυναμικού τύπου και έχει αυτόματη μνήμη διαχείρισης. Επίσης έχει ενσωματωμένη μια μεγάλη τυπική βιβλιοθήκη, παρέχοντας εργαλεία που μπορούν να χρησιμοποιηθούν σε πλήθος εργασιών [28].

Η γλώσσα python είναι μια διαδραστική και αντικειμενοστραφής γλώσσα διερμηνέας η οποία είναι πολύ σαφής στην σύνταξη και κατέχει αξιοσημείωτη δύναμη. Είναι συμβατή με αρκετές παραλλαγές της Unix, συμπεριλαμβανομένων Linux και macOS αλλά και Windows [29].

Παρακάτω θα διακρίνουμε τις διαφορές μεταξύ της python και άλλων προγραμματιστικών γλωσσών διερμηνέων όπως η Java, η JavaScript και η C++

- Μπορεί τα προγράμματα της Java να τρέχουν πιο γρηγορά από τα προγράμματα της python αλλά χρειάζονται πολύ περισσότερο χρόνο να αναπτυχθούν. Τα προγράμματα της python είναι πιθανό να είναι 3-5 φορές μικρότερά λόγω της δυναμικής τυπολογικής και στους υψηλού επιπέδου τύπους δεδομένων. Για αυτούς τους λογούς η python είναι καταλληλότερη ως συνδετική γλώσσα ενώ η java προτείνεται ως γλώσσα υλοποίησης χαμηλού

επιπέδου. Το αποτέλεσμα μπορεί να είναι εξαιρετικό εάν συνδυάσουμε αυτές τις δυο γλώσσες προγραμματισμού.

- Οι αντικειμενοστραφείς ιδιότητες της python είναι παρόμοιες με αυτής της JavaScript. Στις δυο αυτές γλώσσες προγραμματισμού δεν απαιτείται να χρησιμοποιήσουμε κλάσεις για να αναπτύξουμε τον κώδικα, το οποίο τις κάνει κατάλληλες όταν θέλουμε να αναπτύξουμε γρηγορά ένα πρόγραμμα. Σε αντίθεση με την JavaScript η Python μπορεί να χρησιμοποιηθεί για την ανάπτυξη πολύ μεγαλύτερων προγραμμάτων.
- Όσα αναφέρθηκαν για την περίπτωση της Java ισχύουν και για την γλώσσα προγραμματισμού C++. Ένας κώδικας γραμμένος σε python είναι συνήθως 5-10 φορές μικρότερος από έναν αντίστοιχο κώδικα γραμμένο σε C++. Η χρήση της python ως συνδετικής γλώσσας για το συνδυασμό δομικών στοιχείων γραμμένων σε C++, μπορεί να αποφέρει εξαιρετικά αποτελέσματα [30].

3.2 Scikit-Learn

Μια βιβλιοθήκη μηχανικής μάθησης ανοιχτού κώδικα που είναι γραμμένη σε python είναι η scikit-learn. Με την χρήση της scikit-learn μπορούμε με εύκολο και γρήγορο τρόπο να ενσωματώσουμε μεθόδους μηχανικής μάθησης σε κώδικα που είναι γραμμένος στη γλώσσα προγραμματισμού python. Περιλαμβάνει μεθόδους ταξινόμησης, παλινδρόμησης, εκτίμηση πίνακα συνδιακύμανσης, μείωσης διαστάσεων και επεξεργασία δεδομένων. Η συγκεκριμένη βιβλιοθήκη είναι διαθέσιμη για ένα μεγάλο πλήθος λειτουργικών συστημάτων και η εγκατάσταση της είναι αρκετά απλή. Ενώ ταυτόχρονα η βιβλιοθήκη επιδέχεται συνέχεις βελτιώσεις και αναβαθμίσεις. Χρησιμοποιείται ευρέως τόσο σε ερευνητικά έργα όσο και για εμπορικούς σκοπούς [31].

3.3 imblearn

Μια εργαλειοθήκη ανοιχτού κώδικα για την προγραμματιστική γλώσσα python, η οποία μας παρέχει ένα σύνολο εργαλείων για την αντιμετώπιση των μη

ισορροπημένων συνόλων δεδομένων, ένα πρόβλημα το οποίο συναντάμε συχνά στην μηχανική μάθηση, είναι η εργαλειοθήκη imlearn ή imbalanced-learn.

Σε αυτήν την βιβλιοθήκη συναντάμε μεθόδους όπως:

- Υποδειγματοληψίας
- Υπερδειγματοληψίας
- Συνδυασμός υπόδειγματοληψίας και υπερδειγματοληψίας
- Μέθοδοι μάθησης από σύνολα

Το συγκεκριμένο εργαλείο κυκλοφορεί και διανέμεται με άδεια Massachusetts Institute of Technology (MIT). Επίσης υπάρχει πλήρης συμβατότητα με την βιβλιοθήκη scikit-learn και αποτελεί κομμάτι του scikit-learn-contrib, ενός έργου που σκοπό έχει να επεκτείνει τις βασικές δυνατότητες του scikit-learn [32].

3.4 Pandas

Μια βιβλιοθήκη με πολλές δομές δεδομένων, χρήσιμη για να επεξεργαστούμε δομημένα σύνολα δεδομένων, είναι η βιβλιοθήκη pandas. Η βιβλιοθήκη pandas αναπτύσσεται συνεχώς από το 2008, με σκοπό να εισάγει στην γλώσσα προγραμματισμού python εξειδικευμένα εργαλεία που χρησιμοποιούνται στην στατιστική ανάλυση, την επεξεργασία βάσεων δεδομένων, την οικονομική επιστήμη, κ.α.

Η pandas έχει καταστήσει την γλώσσα προγραμματισμού python πιο ελκυστική για τον χρήστη και χρησιμοποιείται για επαγγελματικούς σκοπούς αλλά είναι ευρέως διαδεδομένη και στους ακαδημαϊκούς κύκλους [33].

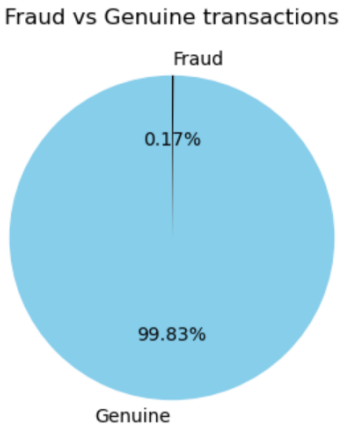
3.5 Dataset

Για την πραγματοποίηση αυτής της εργασίας βασιστήκαμε σε ένα σύνολο δεδομένων που περιέχει συναλλαγές που πραγματοποιήθηκαν από Ευρωπαίους καταναλωτές τον Σεπτέμβριο του 2013 σε διάστημα δυο ημέρων. Αποτελείται μόνο από αριθμητικά δεδομένα τα οποία είναι αποτέλεσμα μετασχηματισμού PCA. Λόγω του νόμου περί πνευματικών δικαιωμάτων δεν γνωρίζουμε τι αντιπροσωπεύουν τα 28 χαρακτηριστικά κάθε συναλλαγής εκτός από την στήλη Time και την στήλη Amount. Η στήλη Time μας δείχνει τα δευτερόλεπτα που μεσολάβησαν μεταξύ κάθε

συναλλαγής σε σχέση με την πρώτη. Ενώ η στήλη Amount μας δείχνει το ποσό της συναλλαγής. Η τελευταία στήλη Class μας υποδεικνύει με 1 τις περιπτώσεις όπου η συναλλαγή είναι απάτη και με 0 όταν η συναλλαγή είναι έγκυρη. Το συγκεκριμένο dataset το εντοπίσαμε στην διαδικτυακή πλατφόρμα Kaggle.

Το Kaggle είναι μια διαδικτυακή κοινότητα, θυγατρική της google, που επιτρέπει στους χρήστες τον διαμοιρασμό συνόλων δεδομένων για την δημιουργία μοντέλων τεχνητής νοημοσύνης [34].

Κάνοντας μια πρώτη ανάλυση του συνόλου δεδομένων, παρατηρούμε ότι το συγκεκριμένο σύνολο δεδομένων παρουσιάζει έντονη ανισορροπία.



Εικόνα 7: Κατανομή συνόλου δεδομένων

Από τις 284807 συναλλαγές που εντοπίστηκαν, οι 284315 θεωρούνται έγκυρες με τις απάτες να ανέρχονται σε μόλις 492.

Αυτό σημαίνει ότι οι απάτες αποτελούν 0.172 % του συνόλου δεδομένων

Κεφαλαίο 4

Μοντελοποιηση

Στο κεφάλαιο αυτό παρουσιάζεται η μοντελοποίηση του συστήματος.

Κατά την ανάπτυξη του κώδικα, εφαρμόστηκαν τα μοντέλα μηχανικής μάθησης Decision Tree, Random Forest, Logistic Regression, K-Means, Gradient Boosting Algorithm και Support Vector Machine.

Ενώ για την αντιμετώπιση της ανισορροπίας των δεδομένων, χρησιμοποιήθηκαν οι τεχνικές Random Oversampling, Smote, Smote-Tomek και Weight class ενώ δοκιμάσαμε και την εκτέλεση των μοντέλων χωρίς κάποια τεχνική υπερδειγματοληψίας ή υπόδειγματοληψίας.

Για να μετρήσουμε την απόδοση των μοντέλων χρησιμοποιήθηκαν οι μετρικές recall, precision, f1-score και accuracy.

Πριν ξεκινήσουμε την κατασκευή των μοντέλων μας, επεξεργαστήκαμε τα δεδομένα με τους τρόπους που αναλύουμε παρακάτω. Σκοπός είναι να προετοιμάσουμε τα δεδομένα για την εκπαίδευση.

Αρχικά αφαιρούμε τη στήλη class από το σύνολο δεδομένων και την αποθηκεύουμε σε μια μεταβλητή y, που περιέχει τις ετικέτες στόχου. Στη συνέχεια δημιουργούμε δυο υποσύνολα, που θα χρησιμοποιηθούν για την εκπαίδευση (X_train) και αξιολόγηση (Y_train) των μοντέλων, διατηρώντας την ισορροπία στις κατηγορίες.

Έπειτα εφαρμόζουμε την τεχνική quantile normalization.

Quantile normalization

Πρόκειται για μια στατιστική μέθοδο που εφαρμόζεται για την τυποποίηση δεδομένων, που προκύπτουν από διαφορετικά σύνολα, έτσι ώστε να είναι συγκρίσιμα μεταξύ τους. Κατά την διαδικασία του quantile normalization τα δεδομένα ταξινομούνται και κάθε τιμή αντικαθίσταται με τον μέσο όρο των αντίστοιχων θέσεων σε όλα τα σύνολα δεδομένων. Με αυτόν τον τρόπο, πετυχαίνουμε ομοιομορφία στα δεδομένα ενώ διατηρούν την ίδια κατανομή. Τεχνικές διάφορες που

μπορεί να προκύπτουν από διαφορά στις συνθήκες ή διαφορετικών εργαλείων μέτρησης, απομακρύνονται [35].

Μοντελοποιηση του μοντέλου Decision Tree

Στην πρώτη εφαρμογή δεν θα χρησιμοποιήσουμε κάποια μέθοδό υποδειγματοληψίας ή υπερδειγματοληψίας.

Το πρώτο βήμα είναι να δημιουργήσουμε ένα αντικείμενο StratifiedKFold για να χωρίσουμε τα δεδομένα μας σε 5 υποκατηγορίες, χωρίς να αλλοιώσουμε την αναλογία των κλάσεων. Χρησιμοποιούμε τα δεδομένα αυτά για να εφαρμόσουμε την διασταυρουμένη επικύρωση, με βάση τα δεδομένα X_train και Y_train.

Για το δέντρο απόφασης ορίζουμε τις υπερπαραμέτρους max_depth από 2 μέχρι 12 με βήμα 2 και Min_samples_leaf από 1 έως 6.

Το max_depth είναι μια παράμετρος που ορίζει το μέγιστο βάθος ενός δέντρου απόφασης και επηρεάζει την απόδοση του μοντέλου. Η επιλογή της συγκεκριμένης παραμέτρου θα πρέπει να γίνεται με σύνεση καθώς ένας πολύ μεγάλος αριθμός max_depth μπορεί να οδηγήσει σε υπερπροσαρμογή ενώ ένας πολύ χαμηλός σε υποπροσαρμογή [36].

Το Min_samples_leaf είναι ο ελάχιστος αριθμός δειγμάτων που απαιτείται να υπάρχουν σε ένα κόμβο φύλλου. Βοηθά το μοντέλο να παραμένει απλό, αποφεύγοντας την υπερπροσαρμογή [37].

Στην συνέχεια δημιουργούμε ένα GridSearchCV και το εκπαιδεύουμε στα υποσύνολα x_train και y_train.

Με το GridSearchCV καταφέρνουμε να εντοπίζουμε τις καλύτερες υπερπαραμέτρους σε μοντέλα μηχανικής μάθησης. Όλες οι πιθανές τιμές εξετάζονται ελέγχοντας όλους τους πιθανούς συνδυασμούς. Τελικός σκοπός είναι να βρεθεί ο συνδυασμός που θα μεγιστοποιεί την απόδοση του μοντέλου. Είναι ένα χρήσιμο και απλό εργαλείο, το οποίο όμως μπορεί να αποβεί ιδιαίτερα χρονοβόρο και δαπανηρό [38].

Επιλέγονται οι βέλτιστες υπερπαράμετροι με βάση τη μετρική ανάκλησης που παράγεται από τη διαδικασία της διασταυρούμενης επικύρωσης. Η μετρική της

ανάκλησης επιλέχθηκε ως η πιο κατάλληλη για την περίπτωση μας, καθώς ο στόχος μας είναι να μειώσουμε τις περιπτώσεις όπου μια απατή μπορεί λανθασμένα να προβλεφθεί ως έγκυρη (false negatives). Τέλος, με βάση το καλύτερο μοντέλο που βρέθηκε προσπαθούμε να προβλέψουμε τις ετικέτες του X_test.

Τυπώνουμε τον πίνακα σύγχυσης και τις μετρικές recall, precision, F1-score και accuracy.

Επαναλαμβάνουμε την διαδικασία εφαρμόζοντας τις τεχνικές Random Oversampling, SMOTE, SMOTE Tomek και Weight class

Μοντελοποίηση του μοντέλου Random Forest

Χρησιμοποιούμε την ιδιά διαδικασία για την υλοποίηση του Random Forest και ορίζουμε max_depth από 2 μέχρι 12 με βήμα 2 και n_estimators (αριθμός των δέντρων απόφασης) από 50 έως 200 με βήμα 50.

Μοντελοποίηση του μοντέλου K-means

Ορίζουμε τον αριθμό των κοντινότερων γειτόνων από 3 έως 10.

Η μέθοδος μέτρησης της απόστασης που χρησιμοποιούμε είναι είτε η “euclidean” είτε η “manhatan”.

Για να υπολογίσουμε την βαρύτητα μεταξύ των γειτόνων, είτε θεωρούμε ότι όλοι οι γείτονες έχουν την ιδιά βαρύτητα (uniform), είτε ότι τα βάρη είναι αντιστρόφως ανάλογα με την απόσταση (distance).

Το GridSearchCV θα βρει τον καλύτερο συνδυασμό των παραπάνω παραμέτρων ώστε να μεγιστοποιείται η αποδοτικότητα του μοντέλου.

Μοντελοποίηση του μοντέλου Logistic Regression

Οι δυο αλγόριθμοι που χρησιμοποιούνται για το μοντέλο Logistic Regression είναι οι (liblinear και saga).

Ορίζουμε το max_iter=5000 ως το μέγιστο αριθμό επαναλήψεων που μπορεί να εκτελέσει το μοντέλο.

Η τιμή tol=1e-4, ορίζει την ανοχή. Δηλαδή αν η αλλαγή από μια επανάληψη είναι μικρότερη της ανοχής, η διαδικασία σταματά.

Τέλος C: [0.1, 1, 10] το οποίο καθορίζει το πόσο χαλαρό ή αυστηρό θα είναι το μοντέλο.

Μοντελοποίηση του μοντέλου Gradient Boosting Algorithm

Για την εφαρμογή του Gradient Boosting Algorithm χρησιμοποιήθηκε ο ρυθμός μετάδοσης (learning_rate=0.01,0.05 και 0.1) Ο αριθμός των δέντρων ορίστηκε στα 100 200 και 300. Το max_depth είναι ίσο με 10, ενώ η παράμετρος subsample που καθορίζει το ποσοστό των δεδομένων που θα χρησιμοποιηθούν για την εκπαίδευση ορίστηκε στο 0.8 και 1.0.

Μοντελοποίηση του μοντέλου Support Vector Machine

Λόγω του μεγάλου όγκου δεδομένων που έχουμε στην διάθεση μας, την περιορισμένη υπολογιστική ισχύ την οποία διαθέτουμε αλλά και τον χρονοβόρο και περίπλοκο τρόπο λειτουργίας του μοντέλου Support Vector Machine, αναγκαστήκαμε να παραλείψουμε την εκτέλεση του GridSearchCV ώστε να μειώσουμε την υπολογιστική ισχύ που απαιτούνταν για την υλοποίηση του συγκεκριμένου κώδικα.

Για την δημιουργία πιο ευέλικτων και μη γραμμικών διαχωρισμών αναμεσά στις κλάσεις των δεδομένων, χρησιμοποιήθηκε ο πυρήνας Radial Basis Function (rbf). Η τιμή scale υπολογίζει το gamma ως: $\frac{1}{n \times \sigma^2}$

Όπου n ο αριθμός των χαρακτηριστικών και σ^2 η διακύμανση των δεδομένων.

Η ποινή για τα σφάλματα που κάνει το μοντέλο ορίστηκε σε C=0.1.

Μοντελοποίηση νευρωνικού δικτύου

Από το σύνολο δεδομένων “data” διαχωρίζουμε τα δεδομένα x και την ετικέτα στόχο “class”. Στη συνέχεια, διακρίνουμε το σύνολο εκπαίδευσης από το σύνολο ελέγχου ή αξιολόγησης. Το 30% των δεδομένων χρησιμοποιείται για το σύνολο δοκιμής (test set). Τα δεδομένα που χρησιμοποιούνται για την εκπαίδευση καταλαμβάνουν το 70%. Από αυτό το ποσοστό, το 20% απαρτίζει το (validation set), το οποίο χρησιμοποιείται για την ρύθμιση των υπερπαραμέτρων του μοντέλου, καθώς και για τον έλεγχο του

early stopping, προκειμένου να αποφευχθεί η υπερβολική εκπαίδευση του μοντέλου. Τέτοιες παράμετροι είναι η ταχύτητα μάθησης, ο αριθμός εποχών (epochs) κ.λπ. Το 80% που περισσεύει από το σύνολο εκπαίδευσης, χρησιμοποιείται για την πραγματική εκπαίδευση του μοντέλου.

Στη συνέχεια, κλιμακώνουμε τα χαρακτηριστικά των δεδομένων με την μέθοδο του quantile normalization και υπολογίζουμε τα βάρη για τις δυο κατηγορίες, με βάση το ποσό συχνά εμφανίζονται.

Δημιουργούμε το νευρωνικό δίκτυο, το οποίο αποτελείται από πυκνές στρώσεις (dense layers), κάθε μια με 256 νευρώνες και λειτουργία ενεργοποίησης “relu”, ενώ χρησιμοποιείται και batchNormalization για να ομαλοποιήσει τις εισόδους κάθε στρώσης και να επιταχύνει την εκπαίδευση. Επιπλέον για να αποφύγουμε την υπερ προσαρμογή εφαρμόζεται η τεχνική Dropout, όπου τυχαίοι νευρώνες απενεργοποιούνται κατά την διάρκεια της εκπαίδευσης. Η συνάρτηση ενεργοποίησης Sigmoid, χρησιμοποιείται στο τελευταίο στρώμα, η οποία είναι κατάλληλη για δυαδική ταξινόμηση, όπως η ανίχνευση απάτης.

Συνεχίζουμε χρησιμοποιώντας τον βελτιστοποίητη Adam, ο οποίος είναι ένας αλγόριθμος που χρησιμοποιείται στην εύρεση των καλυτέρων ρυθμίσεων για την μεγαλύτερη ακρίβεια των προβλέψεων. Ορίζουμε την συνάρτηση απώλειας (binary_cross_entropy), καθώς και τις μετρικές που θα εξεταστούν κατά την εκπαίδευση.

Εκπαιδεύουμε το μοντέλο με τα δεδομένα εκπαίδευσης (X_train και Y_train) και ελέγχουμε την απόδοση του μοντέλου σε κάθε εποχή με τα δεδομένα επικύρωσης (x_validate και y_validate). Τέλος, κάνουμε τις προβλέψεις για το σύνολο εκπαίδευσης και εκτυπώνουμε τα αποτελέσματα αξιολόγησης

Κεφαλαίο 5

Αποτελέσματα

Σε αυτό το κεφάλαιο θα δούμε τα αποτελέσματα των μοντέλων τα οποία δημιουργήσαμε στο προηγούμενο κεφάλαιο, χρησιμοποιώντας το σύνολο δεδομένων ελέγχου.

Λόγω της φύσης του συνόλου δεδομένων που χρησιμοποιούμε, δεν μπορούμε να βασιστούμε στην μετρική accuracy. Πάρα την υψηλή ακρίβεια που μπορεί να πέτυχει είναι πιθανό να μην καταφέρει να ταξινομήσει σωστά τα πιο σπάνια παραδείγματα. Η συγκεκριμένη μετρική μπορεί να είναι παραπλανητική σε περιπτώσεις ανισόρροπων δεδομένων καθώς μπορεί να υπάρξει μεγάλη μεροληψία υπέρ της κυρίαρχης κατηγορίας. Για παράδειγμα αν είχαμε ένα σύνολο δεδομένων όπου το 99% των περιπτώσεων ήταν της κλάσης Α και μόνο το 1% ήταν της κλάσης Β, ένα μοντέλο που ταξινομεί όλες τις περιπτώσεις ως κλάση Α θα έδινε ακρίβεια 99%. Αν και η ακρίβεια είναι πολύ υψηλή, το μοντέλο είναι μη αποτελεσματικό, αφού δεν μπορεί να ανιχνεύσει καμία περίπτωση της κλάσης Β [39].

Για αυτόν τον λόγο θα ερευνήσουμε τα αποτελέσματα των τριών άλλων μετρικών.

Τα αποτελέσματα του Decision Tree συνοψίζονται στον Πίνακα 2:

DECISION TREE				
	RECALL	PRECISION	F1-SCORE	ACCURACY
NO SAMPLING	0.5845	0.8829	0.7033	0.9991
OVERSAMPLING	0.8450	0.0452	0.0859	0.9700
SMOTE	0.8591	0.0601	0.1124	0.9773
SMOTE AND TOMEK	0.0563	0.0833	0.0672	0.9973
WEIGHT CLASS	0.9084	0.0296	0.0574	0.9503

Πίνακας 2: Αποτελέσματα Decision Tree

Στα αποτελέσματα του δέντρου απόφασης παρατηρούμε ότι η μετρική της ανάκλησης (recall) έχει την μεγαλύτερη απόδοση κατά την χρήση της τεχνικής weight class (0.9084) ενώ το μεγαλύτερο precision εντοπίζεται στην τεχνική της μη υποδειγματοληψίας ή υπερδειγματοληψίας, στην ιδιά τεχνική παρατηρούμε και το μεγαλύτερο f1-score.

Τα αποτελέσματα του Random Forest συνοψίζονται στον Πίνακα 3:

RANDOM FOREST				
	RECALL	PRECISION	F1-SCORE	ACCURACY
NO SAMPLING	0.7535	0.8458	0.9995	0.9639
OVERSAMPLING	0.8380	0.1944	0.3156	0.9939
SMOTE	0.8661	0.3098	0.4564	0.9965
SMOTE AND TOMEK	0.7394	0.9210	0.8203	0.9994
WEIGHT CLASS	0.8169	0.3853	0.5237	0.9975

Πίνακας 3: Αποτελέσματα Random Forest

Παρατηρούμε το μεγαλύτερο recall κατά την χρήση της τεχνικής του Smote (0.8661), ενώ το μεγαλύτερο precision κατά την χρήση της εφαρμογής Smote And Tomek. Ενώ στην μετρική f1-score όταν χρησιμοποιούμε την τεχνική της μη υποδειγματοληψίας ή υπερδειγματοληψίας.

Τα αποτελέσματα του K-Means συνοψίζονται στον Πίνακα 4:

K-MEANS				
	RECALL	PRECISION	F1-SCORE	ACCURACY
NO SAMPLING	0.7464	0.9636	0.8412	0.9995
OVERSAMPLING	0.8098	0.5958	0.6865	0.9987
SMOTE	0.8309	0.3881	0.5291	0.9975
SMOTE AND TOMEK	0.8239	0.4642	0.5939	0.9981
WEIGHT CLASS	0.7535	0.9816	0.8525	0.9995

Πίνακας 4: Αποτελέσματα K-Means

Στα αποτελέσματα του μοντέλου K-Means εντοπίζουμε το καλύτερο recall κατά την χρήση της τεχνική Smote (0.8309) και το πιο υψηλό f1-score κατά την χρήση της τεχνικής weight class. Το precision αποτελεί την πιο υψηλή μετρική του μοντέλου και το εντοπίζουμε κατά την χρήση της τεχνικής weight class.

Τα αποτελέσματα του Logistic Regression συνοψίζονται στον Πίνακα 5:

LOGISTIC REGRESSION				
	RECALL	PRECISION	F1-SCORE	ACCURACY
NO SAMPLING	0.5633	0.6779	0.9991	0.8510
OVERSAMPLING	0.8873	0.0522	0.0987	0.9729
SMOTE	0.8802	0.0491	0.0931	0.9713
SMOTE AND TOMEK	0.8802	0.0493	0.0935	0.9715
WEIGHT CLASS	0.8873	0.0529	0.1	0.9733

Πίνακας 5: Αποτελέσματα Logistic Regression

Βρίσκουμε το μεγαλύτερο recall κατά την χρήση της τεχνικής random oversampling (0.8873) ενώ το μεγαλύτερο precision στην τεχνική μη υποδειγματοληψίας ή υπερδειγματοληψίας. Στην ίδια τεχνική παρατηρούμε και το υψηλότερο f1-score.

Τα αποτελέσματα του Gradient Boosting Algorithm συνοψίζονται στον Πίνακα 6:

GRADIENT BOOSTING ALGORITHM				
	RECALL	PRECISION	F1-SCORE	ACCURACY
NO SAMPLING	0.7394	0.7342	0.7368	0.9991
OVERSAMPLING	0.7253	0.8956	0.8015	0.9994
SMOTE	0.7535	0.7278	0.7404	0.9991
SMOTE AND TOMEK	0.7676	0.8195	0.7927	0.9993
WEIGHT CLASS	0.7253	0.9449	0.8207	0.9994

Πίνακας 6: Αποτελέσματα Gradient Boosting Algorithm

Το recall έχει την μεγαλύτερη αποδοτικότητα κατά την χρήση της εφαρμογής SMOTE and Tomek (0.7676) ενώ το f1-score με την τεχνική weight class. Το precision παρουσιάζει πολύ καλή απόδοση κατά την χρήση της τεχνικής weight class.

Τα αποτελέσματα του Support Vector Machine συνοψίζονται στον Πίνακα 6:

SUPPORT VECTOR MACHINE				
	RECALL	PRECISION	F1-SCORE	ACCURACY
NO SAMPLING	0.1971	0.3274	0.9986	0.9655
OVERSAMPLING	0.8661	0.1066	0.1899	0.9876
SMOTE	0.8802	0.0882	0.1604	0.9846
SMOTE AND TOMEK	0.8802	0.0882	0.1604	0.9846
WEIGHT CLASS	0.7535	0.1725	0.2808	0.9935

Πίνακας 7: Αποτελέσματα Support Vector Machine

Στο τελευταίο μας μοντέλο μηχανικής μάθησης (support vector machine), η μετρική της ανάκλησης έχει την υψηλότερη απόδοση κατά την χρήση της τεχνικής SMOTE και SMOTE AND TOMEK (0.8802)

Τα αποτελέσματα του Νευρωνικού δικτύου συνοψίζονται στον Πίνακα 7:

ARTIFICIAL NEURAL NETWORK			
Recall	Precision	F1-score	Accuracy
0.845588	0.851852	0.848708	0.99952

Πίνακας 8: Αποτελέσματα Νευρωνικού δικτύου

Τα ευρήματα του νευρωνικού δικτύου δείχνουν μια ικανοποιητική απόδοση. Η μετρική accuracy φτάνει στο 0.99952 ενώ του precision το 0.851852. Το recall είναι 0.845588, που σημαίνει ότι το μοντέλο είναι σε θέση να εντοπίζει σωστά τις περισσότερες περιπτώσεις των θετικων παραδειγμάτων, αν και με κάποια λάθη (False Negatives). Η μετρική του F1-score ανέρχεται σε 0.848708.

Ο πίνακας Confusion Matrix δείχνει 115 True Positives και 21 False Negatives, πράγμα που υποδηλώνει ότι το μοντέλο μπορεί να εντοπίζει την πλειονότητα των θετικων περιπτώσεων, ενώ στην αναγνώριση της θετικής κλάσης υπάρχει ένα μικρό ποσοστό λαθών.

Confusion Matrix	
85287	20
21	115

Πίνακας 9: Confusion Matrix Νευρωνικού δικτύου

Κεφάλαιο 6

Ανάλυση Ενεργειακής Αποδοτικότητας Αλγορίθμων Ανίχνευσης Απάτης

Επόμενο βήμα μετά την λήψη των αποτελεσμάτων είναι η αξιολόγηση του ενεργειακού κόστους και η σύγκριση της αποδοτικότητας των αλγορίθμων. Αυτό που θα προσπαθήσουμε να διερευνήσουμε είναι το αν είναι πιο βιώσιμο τα μοντέλα αυτά να υπάρχουν σε έναν κεντρικό διακομιστή (server) ή να τοποθετηθούν σε συσκευές με μικρότερη ισχύ π.χ. τα προσωπικά κινητά των χρηστών. Για να μετρήσουμε αυτό το κόστος αρχικά επιλέγουμε τα μοντέλα που για το πρόβλημα μας αποδείχθηκαν πιο αποδοτικά στην διαδικασία ανίχνευσης απάτης.

Η επιλογή των μοντέλων έγινε με γνώμονα την φύση του προβλήματος που κληθήκαμε να επιλύσουμε, δηλαδή την ανάγκη για υψηλή ακρίβεια στην αναγνώριση απατών. Για τον λόγο αυτό προτιμήθηκαν οι μετρικές recall και F1-score. Το recall έχει την δυνατότητα να εντοπίζει πόσες από τις πραγματικές απάτες εντοπίστηκαν. Σε περιπτώσεις τραπεζικών απατών είναι κρίσιμο να ελαχιστοποιηθούν τα σφάλματα, διότι η παράληψη μιας απάτης, δηλαδή η μη ανίχνευση της, μπορεί να προκαλέσει σοβαρές συνέπειες.

Το F1-score που αποτελεί έναν συνδυασμό των μετρικών ακρίβειας (precision) και ανάκλησης (recall), μπορεί να αποδώσει μια εικόνα για την γενικότερη αποτελεσματικότητα του μοντέλου, σε ένα σύνολο δεδομένων που χαρακτηρίζεται από έντονη ανισορροπία στα δεδομένα του. Δίνοντας έμφαση όχι μόνο στην δυνατότητα αναγνώρισης αλλά και στην σωστή κατηγοριοποίηση των δεδομένων.

Τα μοντέλα αυτά είναι:

- Δέντρο απόφασης με την χρήση weight class
- Τυχαίο δάσος χωρίς την χρήση υπο/ υπερ δειγματοληψίας
- Logistic regression χωρίς την χρήση υπο/ υπερ δειγματοληψίας
- Logistic regression με την χρήση oversampling

Για να μετρήσουμε το κόστος που θα έχουν αυτά τα μοντέλα σε έναν κεντρικό διακομιστή, θα χρησιμοποιήσουμε την διαδικτυακή πλατφόρμα Kaggle.

Το πρώτο μοντέλο που εκτελέστηκε σε περιβάλλον Kaggle είναι το logistic regression χωρίς υπο/υπερ δειγματοληψία. Για όλα τα μοντέλα μας χρησιμοποιήσαμε 100.000 δείγματα. Ο συνολικός χρόνος που χρειάστηκε ο αλγόριθμος logistic regression χωρις την χρήση υπο/ υπερ δειγματοληψίας για την εκτέλεση ήταν 147 δευτερόλεπτα (2 λεπτά και 27 δευτερόλεπτα). Αυτό σημαίνει ότι επιτυγχάνουμε ρυθμό 676.81 επαναλήψεις το δευτερόλεπτο.

Παρακάτω ακολουθούν τα αποτελέσματα των υπολοίπων μοντέλων.

Το μοντέλο logistic regression με τη χρήση της τυχαίας υπερδειγματοληψίας χρειάστηκε 152 δευτερόλεπτα (2 λεπτά και 32 δευτερόλεπτα) δηλαδή επιτυγχάνουμε 657.62 επαναλήψεις το δευτερόλεπτο.

Το μοντέλο τυχαίων δασών χωρίς την χρήση υπο/υπερδειγματοληψίας χρειάστηκε 644 δευτερόλεπτα ώστε να επιτευχθούν 155.13 επαναλήψεις το δευτερόλεπτο.

Το μοντέλο δέντρου απόφασης εκτελέστηκε σε 2 λεπτά και 36 δευτερόλεπτα, δηλαδή σε 156 δευτερόλεπτα και έκανε 639.17 επαναλήψεις

Στον παρακάτω πίνακα φαίνονται τα αποτελέσματα των μετρήσεων.

Μοντέλο	**Χρόνος (sec)**	**It/s**
Random Forest no over/under sampling	147	676.81
Logistic Regression no over/under sampling	152	657.62
Logistic Regression random oversampling	644	155.13
Decision Tree class weights	156	639.17

Πίνακας 10: Μετρήσεις Απόδοσης Αλγορίθμων Μηχανικής Μάθησης σε Διακομιστή: Χρόνος Εκτέλεσης και It/s

Σύμφωνα με τον διαδικτυακό ιστότοπο Kaggle, ο ιστότοπος χρησιμοποιεί τον επεξεργαστή Intel Xeon E5. Η κατανάλωση ενέργειας ανέρχεται στα 115 watts. Η τιμή αυτή αναφέρεται στη μέγιστη θερμική απόδοση που χρειάζεται ο επεξεργαστής ώστε να λειτουργεί σωστά κάτω από πλήρες φόρτο.

Αν πολλαπλασιάσουμε αυτήν την τιμή, με τα δευτερόλεπτα που απαιτούνται για την εκτέλεση ενός μοντέλου, μπορούμε να γνωρίζουμε την ενέργεια που θα καταναλωθεί από τον επεξεργαστή σε αυτήν την περίοδο.

Ο τύπος δίνεται από την σχέση:

Ενέργεια (σε Joules) = Ισχύς (σε Watts) × Χρόνος (σε δευτερόλεπτα)

Παρακάτω ακολουθούν οι υπολογισμοί για την ενέργεια που απαιτείται από τον επεξεργαστή για την εκτέλεση κάθε μοντέλου

Random Forest no over/under sampling:

115W×147sec= 16905 Joules

Logistic Regression no over/under sampling:

115W×152sec=17480 Joules

Logistic Regression random oversampling:

115W×644sec=74060 Joules

Decision Tree class weights:

115W×156sec=17940 Joules

Η συνηθέστερη μέθοδος χρέωσης του ρεύματος είναι οι κιλοβατώρες (kWh), οπότε για την καλύτερη κατανόηση των δεδομένων μας, θα μετατρέψουμε τα Joules σε kWh.

Μετατροπή σε kWh ⟶ 1kWh = 3.600.000 Joules

Δημιουργούμε τον παρακάτω πίνακα

Μοντέλο	**Χρόνος (sec)**	**It/s**	**kWh**
Random Forest no over/under sampling	147	676.81	0.00470 kWh
Logistic Regression no over/under sampling	152	657.62	0.00485 kWh
Logistic Regression random oversampling	644	155.13	0.02057 kWh
Decision Tree class weights	156	ε639.17	0.00498 kWh

Πίνακας 11: Μετρήσεις Απόδοσης Αλγορίθμων Μηχανικής Μάθησης σε Διακομιστή: Χρόνος Εκτέλεσης, It/s και kWh

Η τιμή της kWh στις 6 Φεβρουάριου 2025, ανέρχεται στα 0.1398 €/kWh. Οπότε διαμορφώνουμε τον παρακάτω πίνακα

Μοντέλο	Χρόνος (sec)	It/s	kWh	Κόστος (€)
Random Forest no over/under sampling	147	676.81	0.00470 kWh	0.00066 €
Logistic Regression no over/under sampling	152	657.62	0.00485 kWh	0.00068 €
Logistic Regression random oversampling	644	155.13	0.02057 kWh	0.00288 €
Decision Tree class weights	156	639.17	0.00498 kWh	0.00070 €

Πίνακας 12: Μετρήσεις Απόδοσης Αλγορίθμων Μηχανικής Μάθησης σε Διακομιστή: Χρόνος Εκτέλεσης, It/s, kWh και κόστους

Ο πίνακας 12 παρουσιάζει τα αποτελέσματα από την εκτέλεση 100.000 συναλλαγών στον κεντρικό διακομιστή. Στην πραγματικότητα όμως μια μεγάλη τράπεζα μπορεί να εκτελέσει εκατομμύρια συναλλαγές καθημερινά. Για την περίπτωση μας θα υποθέσουμε ότι ο διακομιστής θα πρέπει να εξυπηρετήσει 1.000.000 πελάτες που κατά μέσο όρο διενεργούν 10 συναλλαγές την ημέρα. Αυτό θα ανάγκαζε το διακομιστή να τρέχει αυτά τα μοντέλα 10.000.000 φορές κάθε μέρα. Οπότε τα αποτελέσματα μας διαμορφώνονται ως εξής :

Μοντέλο	Χρόνος (sec)	It/s	kWh	Κόστος (€)
Random Forest no over/under sampling	14.700	676.81	0.470	0.066
Logistic Regression no over/under sampling	15.200	657.62	0.485	0.068
Logistic Regression random oversampling	64.400	155.13	2.057	0.288
Decision Tree class weights	15.600	639.17	0.498	0.070 €

Πίνακας 13: Μετρήσεις Απόδοσης Αλγορίθμων Μηχανικής Μάθησης σε Διακομιστή: Χρόνος Εκτέλεσης, It/s, kWh και κόστους για 10 εκατομμύριά συναλλαγές

Από τον πίνακα 13 συμπεραίνουμε ότι το κόστος είναι ασφαλώς αυξημένο αλλά παραμένει σημαντικά μικρό σε όλα τα μοντέλα. Ο χρόνος όμως που απαιτείται για την εκτέλεση αυτών των μοντέλων μπορεί να διακυμανθεί από 4 έως 18 ώρες ανάλογα το μοντέλο.

Μέτρηση ενεργειακού κόστους σε IoT συσκευές

Σε αυτό το σημείο θα μετρήσουμε το ενεργειακό κόστος των μοντέλων στην περίπτωση που τα μοντέλα εκτελούνται σε IoT συσκευές, όπως τα έξυπνά κινητά τηλεφωνά. Για να διερευνήσουμε αυτό το κόστος χρησιμοποιήσαμε μια συσκευή Raspberry Pi.

Το Raspberry Pi αποτελεί μια μικρότερη εκδοχή ενός υπολογιστή, που έχει την δυνατότητα να εκτελεί εργασίες με ιδιαίτερη αποτελεσματικότητα. Μπορεί να διεκπεραιώσει εργασίες όπως η περιήγηση στο διαδίκτυο, η δημιουργία και επεξεργασία εγγράφων, η αποστολή και λήψη email και πολλές άλλες. Δημιουργήθηκε από ερευνητές του πανεπιστήμιου του Κέιμπριτζ με στόχο να αυξηθεί η ενασχόληση των φοιτητών για την πληροφορική. Διαθέτει πολλά πλεονεκτήματα, όπως το χαμηλό κόστος και ότι είναι συμβατό με πολλές γλώσσες προγραμματισμού. Επίσης διαθέτει ισχυρό επεξεργαστή και μπορεί να χρησιμοποιηθεί ως φορητός υπολογιστής αν συνδεθεί σε οθόνη. Στα μειονεκτήματα του εντοπίζουμε την έλλειψη ψήκτρας ή ανεμιστήρα που μπορεί να οδηγήσει σε υπερθέρμανση, την απουσία ισχυρού επεξεργαστή γραφικών αλλά και ότι δεν υποστηρίζει Windows OS.

Αποτελεί κορυφαία επιλογή για εκπαιδευτικούς και ερευνητές για την εκμάθηση προγραμματισμού και την ανάπτυξη ενσωματωμένων συστημάτων [40].

Κατά την ερευνά μας χρησιμοποιήσαμε την έκδοση Rassberry pi zero 2w.

Τα παρακάτω αποτελέσματα προέκυψαν αφού εκτελέσαμε τα μοντέλα 5 φορές, χρησιμοποιώντας 10 δείγματα ανά εκτέλεση

Μοντέλο	**Χρόνος (sec)**	**It/s**
Random Forest no over/under sampling	0.775	12.962
Logistic Regression no over/under sampling	0.3555	35.222
Logistic Regression random oversampling	0.164	64.006
Decision Tree class weights	0.713	59.736

Πίνακας 14: Μετρήσεις Απόδοσης Αλγορίθμων Μηχανικής Μάθησης σε Raspberry Pi: Χρόνος Εκτέλεσης και It/s

Η κατανάλωση ενέργειας του Rassberry pi zero 2w υπολογίστηκε στα 3 Watt.

Υπολογίζουμε την ενέργεια που καταναλώνεται

Random Forest no over/under sampling

$3 \times 0.775 = 2.325$ joule

Logistic Regression no over/under sampling

$3 \times 0.3555 = 1.0665$ joule

Logistic Regression random oversampling

$3 \times 0.164 = 0.492$ joule

Decision Tree class weights

$3 \times 0.713 = 2.139$ Joule

Μετατρέπουμε σε Kw/h και δημιουργούμε τον παρακάτω πίνακα.

Μοντέλο	**Χρόνος (sec)**	**It/s**	**kWh**
Random Forest no over/under sampling	0.775	12.962	0.0000006458
Logistic Regression no over/under sampling	0.3555	35.222	0.0000002963
Logistic Regression random oversampling	0.164	64.006	0.0000001367
Decision Tree class weights	0.713	59.736	0.0000005942

Πίνακας 15: Μετρήσεις Απόδοσης Αλγορίθμων Μηχανικής Μάθησης σε Raspberry Pi: Χρόνος Εκτέλεσης,It/s και kWh

Και υπολογίζουμε το κόστος

Μοντέλο	Χρόνος (sec)	It/s	kWh	Κόστος (€)
Random Forest no over/under sampling	0.775	12.962	0.0000006458	0.0000000903
Logistic Regression no over/under sampling	0.3555	35.222	0.0000002963	0.0000000414
Logistic Regression random oversampling	0.164	64.006	0.0000001367	0.0000000191
Decision Tree class weights	0.713	59.736	0.0000005942	0.0000000831

Πίνακας 16: Μετρήσεις Απόδοσης Αλγορίθμων Μηχανικής Μάθησης σε Raspberry Pi: Χρόνος Εκτέλεσης,It/s και κόστους

Τα παραπάνω αποτελέσματα αναδεικνύουν ότι το ενεργειακό κόστος είναι εξαιρετικά χαμηλό και στις 2 περιπτώσεις.

Αυτό που μπορούμε να παρατηρήσουμε είναι ότι τα μοντέλα που κατασκευάσαμε είναι το ίδιο λειτουργικά ακόμα και σε συσκευές με περιορισμένους υπολογιστικούς πόρους, όπως το Rasberry Pi. Αυτό μας δίνει την δυνατότητα να ισχυριστούμε ότι η επιλογή για την εκτέλεση των μοντέλων σε IoT συσκευές είναι η ενδεδειγμένη, αφού αν και η διαφορά του ενεργειακού κόστους είναι αμελητέα, η άμεση ανταπόκριση, η ανεξαρτησία από κεντρικούς διακομιστές αλλά και η μειωμένη πιθανότητα δυσλειτουργιών που παρατηρείται στην περίπτωση της τοπικής εκτέλεσης, καθιστούν αυτήν την επιλογή πιο αξιόπιστη.

Κεφάλαιο 7

Επίλογος

Στο κεφάλαιο αυτό παρατίθενται γενικά συμπεράσματα της διπλωματικής, καθώς και η μελλοντική εργασία που μπορεί να γίνει με βάση αυτή.

7.1 Συμπεράσματα

Τα μοντέλα τεχνητής νοημοσύνης που κατασκευάστηκαν κατά την διάρκεια αυτής της διπλωματικής εργασίας απέδειξαν ότι μπορούμε να βασιστούμε σε μοντέλα μηχανικής μάθησης ή βαθιάς μάθησης ώστε να είμαστε σε θέση να αναγνωρίζουμε τις απάτες που διενεργούνται μέσω πιστωτικών συναλλαγών.

Αναδεικνύοντας την αποτελεσματικότητα της τεχνητής νοημοσύνης αλλά και τον κρίσιμο ρόλο που θα παίξει στα επόμενα χρόνια στην ασφάλεια των τραπεζικών συναλλαγών και όχι μόνο.

Τα μοντέλα Δέντρο απόφασης με την χρήση weight class, Τυχαίο δάσος χωρίς την χρήση υπο/ υπερ δειγματοληψίας , Logistic regression χωρίς την χρήση υπο/ υπερ δειγματοληψίας, Logistic regression με την χρήση oversampling, επιλέχθηκαν λόγω της υψηλών αποτελεσμάτων στις μετρικές recall και F1-score. Μετρικές που κρίνονται ιδιαίτερα σημαντικές για την επίλυση του προβλήματος μας.

Από τα αποτελέσματα των μετρήσεων είδαμε ότι μπορούμε να πετυχαίνουμε πολύ υψηλή απόδοση αυτών των μοντέλων με ελάχιστο ενεργειακό κόστος.

Μπορεί το ενεργειακό κόστος από μόνο του να μην μας δίνει μια ξεκάθαρη απάντηση για το ποια από τις 2 επιλογές θα πρέπει να επιλέξουμε, υπάρχουν όμως και άλλες παράμετροι που θα πρέπει να εξεταστούν.

Δείξαμε ότι τα μοντέλα μας αν τρέξουν σε έναν κεντρικό διακομιστή μπορεί να χρειαστούν από 4 έως 18 ώρες ανάλογα το μοντέλο που θα επιλέξουμε. Η καθυστέρηση αυτή μπορεί να αυξηθεί σημαντικά σε περίπτωση που αυξηθεί ο όγκος των συναλλαγών που θα πρέπει ο διακομιστής να διαχειριστεί. Αν και το ενεργειακό κόστος παραμείνει ιδιαίτερα χαμηλό (μικρότερό του 1€ την ημέρα), η ανάλυση των συναλλαγών απαιτεί την ενασχόληση του διακομιστή για σημαντικό μέρος της

ημέρας, καταναλώνοντας πόρους που θα μπορούσαν να αξιοποιηθούν σε άλλες ενέργειες.

Στην 2η επιλογή, την εκτέλεση των μοντέλων σε κινητές συσκευές, ο χρήστης εκτελεί την επεξεργασία τοπικά πετυχαίνοντας άμεση ανταπόκριση, χωρίς να χρειάζεται να περιμένει στην ουρά ενός κεντρικού διακομιστή, οποίος είναι πιθανό να επεξεργάζεται και να αναλύει εκατοντάδες αν όχι χιλιάδες συναλλαγές κάθε δεδομένη χρονική στιγμή. Επίσης ένας κεντρικός διακομιστής μπορεί να παρουσιάσει προβλήματα στην λειτουργία του για μια σειρά από λογούς, όπως υπερφόρτωση, σφάλματα λογισμικού και περίοδο που ο διακομιστής να είναι εκτός λειτουργίας λόγω συντήρησης ή αναβάθμισης, αφήνοντας τους χρήστες για αυτό το χρονικό διάστημα ευάλωτους.

Εν κατακλείδι η εκτέλεση των μοντέλων σε έναν κεντρικό διακομιστή μπορεί να αποβεί μη βιώσιμη σε βάθος χρόνου αλλά και αναποτελεσματική. Στην αντίθετη περίπτωση μπορούμε να επιτύχουμε μεγαλύτερη ταχύτητά και μεγαλύτερη αμεσότητα στην ανάλυση των συναλλαγών και δεν βασιζόμαστε σε εξωτερικούς πόρους, οι οποίοι είναι πιο ευάλωτοι σε δυσλειτουργίες και επιθέσεις. Αυτό καθιστά την τοποθέτηση των μοντέλων στα προσωπικά κινητά των χρηστών την πιο αποδοτική και ασφαλής επιλογή.

7.2 Μελλοντική εργασία

Η παρούσα διπλωματική εργασία αποτελεί γερή βάση για μελλοντική εργασία προς πάρα πολλές κατευθύνσεις.

- **Βελτιστοποίηση παραμέτρων.** Όπως αναφέρθηκε και στο κεφάλαιο της μοντελοποίησης του Support Vector Machine η περιορισμένη υπολογιστή ισχύ που είχαμε στην διάθεση μας, οδήγησε στο να μην βρούμε τις βέλτιστες υπερπαραμέτρους του μοντέλου, που ως αποτέλεσμα είχε να μην έχουμε την βέλτιστη απόδοση του μοντέλου. Με τη δυνατότητα πρόσβασης σε πιο ισχυρούς υπολογιστικούς πόρους, η εύρεση και βελτιστοποίηση αυτών των παραμέτρων θα μπορούσε να μας οδηγήσει σε πιο αποδοτικά αποτελέσματα και να επηρεάσει την τελικη επιλογή των πιο αποδοτικών λύσεων στο σύνολο των μοντέλων.

- **Ανάπτυξη του συστήματος ώστε να λειτουργεί σε πραγματικές κινητές συσκευές**. Για την προσομοίωση αυτών των μοντέλων σε κινητές συσκευές χρησιμοποιούσαμε μια συσκευή Raspberry Pi. Επόμενο βήμα είναι η δημιουργία μιας εφαρμογής συμβατή για τα έξυπνα κινητά των χρηστών, ώστε να διερευνηθούν τυχόν προβλήματα και αδυναμίες αλλά και να χρησιμοποιηθούν τα μοντέλα για τον λόγο που δημιουργήθηκαν.

- **Ανάλυση κόστους οφέλους.** Η σύγκριση του ενεργειακού κόστους για την επιλογή τοποθέτησης των μοντέλων είναι μόνο μια από τις πολλές παραμέτρους που πρέπει να εξεταστούν για να ληφθεί η τελική απόφαση. Υπάρχουν παράμετροι που θα πρέπει να ληφθούν υπόψη, όπως το κόστος αγοράς και συντήρησης ενός κεντρικού διακομιστή, τα έξοδα ασφάλειας , τα έξοδα διαφήμισης και προώθησης για την εφαρμογή στην περίπτωση που γίνει η επιλογή για την τοποθέτηση των μοντέλων στις κινητές συσκευές κλπ. Η τελική επιλογή θα πρέπει να γίνει με βάση μια ανάλυση κόστους οφέλους η οποία θα παρουσιάζει όλα τα κόστη και τα οφέλη που θα έχει η κάθε επιλογή.

Βιβλιογραφία

[1] I. Chatterjee, *Machine Learning and Its Application: A Quick Guide for Beginners*. Bentham Science Publishers, 2021.

[2] A. L. Samuel, 'Some Studies in Machine Learning Using the Game of Checkers'.

[3] T. M. Mitchell, *Machine learning*, Nachdr. στο McGraw-Hill series in Computer Science. New York: McGraw-Hill, 2013.

[4] I. Goodfellow, Y. Bengio, και A. Courville, *Deep Learning*. Cambridge, Mass: The MIT Press, 2016.

[5] C. M. Bishop, *Pattern recognition and machine learning*. στο Information science and statistics. New York: Springer, 2006.

[6] Q. Bi, K. E. Goodman, J. Kaminsky, και J. Lessler, 'What is Machine Learning? A Primer for the Epidemiologist', *American Journal of Epidemiology*, σελ. kwz189, Οκτωβρίου 2019, doi: 10.1093/aje/kwz189.

[7] J. Gui *κ.ά.*, 'A Survey on Self-Supervised Learning: Algorithms, Applications, and Future Trends', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, τ. 46, τχ. 12, σελ. 9052–9071, Σεπτεμβρίου 2024, doi: 10.1109/TPAMI.2024.3415112.

[8] 'Μολλάς Ιωάννης (2023 Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης (ΑΠΘ)) Τοπική ερμηνεία μοντέλων μηχανικής μάθησης'. Ημερομηνία πρόσβασης: 4 Ιανουάριος 2025. [Έκδοση σε ψηφιακή μορφή]. Διαθέσιμο στο: https://freader.ekt.gr/eadd/index.php?doc=54400&lang=el#p=38

[9] E. S. Gérard Biau, 'A random forest guided tour', Απριλίου 2016.

[10] MARGARET H. DUNHAM, *DATA MINING 'ΕΙΣΑΓΩΓΙΚΑ ΚΑΙ ΠΡΟΗΓΜΕΝΑ ΘΕΜΑΤΑ ΕΞΟΡΥΞΗΣ ΓΝΩΣΗΣ ΑΠΟ ΔΕΔΟΜΕΝΑ'*.

[11] Ιακωβος Στ Βενιέρης,Δήμητρα-Θεοδώρα Ι.Κακλαμάνη,Εμμανουήλ Α.Βαρβαρίγος,Αθανάσιος Δ.Παναγόπουλος, *ΚΙΝΗΤΟΣ ΥΠΟΛΟΓΙΣΜΟΣ ΚΑΙ ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ*. ΤΖΙΟΛΑ.

[12] Ekaba Bisong, *Building Machine Learning and Deep Learning Models on Google Cloud Platform*.

[13] Candice Bentéjac1, · Anna Csörgő2, και · Gonzalo Martínez-Muñoz3, 'A comparative analysis of gradient boosting algorithms'. 24 Αύγουστος 2020.

[14] V. Jakkula, 'Tutorial on Support Vector Machine (SVM)'.

[15] '(PDF) Confusion Matrix'. Ημερομηνία πρόσβασης: 25 Ιανουάριος 2025. [Έκδοση σε ψηφιακή μορφή]. Διαθέσιμο στο: https://www.researchgate.net/publication/355096788_Confusion_Matrix

[16] S. A. Hicks *κ.ά.*, 'On evaluation metrics for medical applications of artificial intelligence', *Sci Rep*, τ. 12, σελ. 5979, Απριλίου 2022, doi: 10.1038/s41598-022-09954-8.

[17] G. Falavigna, *Deep Learning for Beginners*. IT: CNR-Ircres, 2022. Ημερομηνία πρόσβασης: 14 Ιανουάριος 2025. [Έκδοση σε ψηφιακή μορφή]. Διαθέσιμο στο: https://doi.org/10.23760/978-88-98193-2022-04

[18] 'Deep Learning vs. Machine Learning – What's The Difference?' Ημερομηνία πρόσβασης: 14 Ιανουάριος 2025. [Έκδοση σε ψηφιακή μορφή]. Διαθέσιμο στο: https://levity.ai/blog/difference-machine-learning-deep-learning

[19] Koosha Sharifani Mahyar Amini, 'Machine Learning and Deep Learning: A Review of Methods and Applications', 2023.

[20] W. S. Sarle, 'Neural Networks and Statistical Models'.

[21] ‘CS231n Convolutional Neural Networks for Visual Recognition’. Ημερομηνία πρόσβασης: 14 Ιανουάριος 2025. [Έκδοση σε ψηφιακή μορφή]. Διαθέσιμο στο: https://cs231n.github.io/neural-networks-1/
[22] Marcus Gallagher, N. Moustafa, και Erandi Lakshika (Eds.), ‘AI 2020: Advances in Artificial Intelligence’, παρουσιάστηκε στο 33rd Australasian Joint Conference, AI 2020 Canberra, ACT, Australia, November 29–30, 2020 Proceedings, 2020.
[23] A. Y. Liu, ‘The Effect of Oversampling and Undersampling on Classifying Imbalanced Text Datasets’.
[24] Kamaldeep, ‘How to Improve Class Imbalance using Class Weights in Machine Learning?’, Analytics Vidhya. Ημερομηνία πρόσβασης: 11 Ιανουάριος 2025. [Έκδοση σε ψηφιακή μορφή]. Διαθέσιμο στο: https://www.analyticsvidhya.com/blog/2020/10/improve-class-imbalance-class-weights/
[25] A. Fernandez, S. Garcia, F. Herrera, και N. V. Chawla, ‘SMOTE for Learning from Imbalanced Data: Progress and Challenges, Marking the 15-year Anniversary’, *Journal of Artificial Intelligence Research*, τ. 61, σελ. 863–905, Απριλίου 2018, doi: 10.1613/jair.1.11192.
[26] J.-H. Wang, C.-Y. Liu, Y.-R. Min, Z.-H. Wu, και P.-L. Hou, ‘Cancer Diagnosis by Gene-Environment Interactions via Combination of SMOTE-Tomek and Overlapped Group Screening Approaches with Application to Imbalanced TCGA Clinical and Genomic Data’, *Mathematics*, τ. 12, τχ. 14, Art. τχ. 14, Ιανουαρίου 2024, doi: 10.3390/math12142209.
[27] Y. Qiu, C. Dobbelaere, και S. Song, ‘Energy Cost Analysis and Operational Range Prediction Based on Medium- and Heavy-Duty Electric Vehicle Real-World Deployments across the United States’, *World Electric Vehicle Journal*, τ. 14, τχ. 12, Art. τχ. 12, Δεκεμβρίου 2023, doi: 10.3390/wevj14120330.
[28] L. V. Tulchak και A. O. Marchuk, ‘History of Python’.
[29] ‘General Python FAQ’, Python documentation. Ημερομηνία πρόσβασης: 15 Ιανουάριος 2025. [Έκδοση σε ψηφιακή μορφή]. Διαθέσιμο στο: https://docs.python.org/3/faq/general.html
[30] ‘Comparing Python to Other Languages’, Python.org. Ημερομηνία πρόσβασης: 15 Ιανουάριος 2025. [Έκδοση σε ψηφιακή μορφή]. Διαθέσιμο στο: https://www.python.org/doc/essays/comparisons/
[31] Oliver Kramer, *Machine Learning for Evolution Strategies*. Springer, 2016.
[32] F. N. Guillaume Lemaˆıtre και Christos K. Aridas, ‘Imbalanced-learn: A Python Toolbox to Tackle the Curse of Imbalanced Datasets in Machine Learning’. 7/16.
[33] Wes Mckinney, ‘pandas: a Foundational Python Library for Data Analysis and Statistics’. Ιανουάριος 2011.
[34] ‘What is a kaggle? | Kaggle’. Ημερομηνία πρόσβασης: 16 Ιανουάριος 2025. [Έκδοση σε ψηφιακή μορφή]. Διαθέσιμο στο: https://www.kaggle.com/discussions/general/a
[35] Y. Zhao, L. Wong, και W. W. B. Goh, ‘How to do quantile normalization correctly for gene expression data analyses’, *Sci Rep*, τ. 10, τχ. 1, σελ. 15534, Σεπτεμβρίου 2020, doi: 10.1038/s41598-020-72664-6.
[36] Rian Oktafiani1, , Arief Hermawan2, και , Donny Avianto3, ‘Max Depth Impact on Heart Disease Classification: Decision Tree and Random Forest’, *JURNAL RESTI*.

[37] ‘DecisionTreeClassifier’, scikit-learn. Ημερομηνία πρόσβασης: 21 Ιανουάριος 2025. [Έκδοση σε ψηφιακή μορφή]. Διαθέσιμο στο: https://scikit-learn/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html
[38] D. M. Belete και M. D. Huchaiah, ‘Grid search in hyperparameter optimization of machine learning models for prediction of HIV/AIDS test results’, *International Journal of Computers and Applications*, τ. 44, τχ. 9, σελ. 875–886, Σεπτεμβρίου 2022, doi: 10.1080/1206212X.2021.1974663.
[39] C. G. Weng και J. Poon, ‘A New Evaluation Measure for Imbalanced Datasets’.
[40] Hirak Ghael, ‘A Review Paper on Raspberry Pi and its Applications’, Ιανουάριος 2020.

Συντομογραφίες – Αρκτικόλεξα – Ακρωνύμια

IoT	Internet of Things
SSL	Secure Sockets Layer
TLS	Transport Layer Security
2FA	Two-Factor Authentication
SVM	Support Vector Machine
AI	Artificial Intelligence
DNNs	Deep Neural Networks
κ.α.	και άλλα
κλπ	και λοιπά
π.χ.	παραδείγματος χάρη
kWh	κιλοβάτ ανά ώρα

Απόδοση Ξενόγλωσσων Όρων

Απόδοση	Ξενόγλωσσος όρος
Έξυπνα κινητά τηλεφωνά	smartphones
τεχνητή νοημοσύνη	artificial intelligence
Μηχανικής μάθηση	machine learning
Νευρωνικό δίκτυο	neural network
κινητές συσκευές	mobile devices
αλγόριθμος	algorithm
Επιβλεπόμενη μάθηση	Supervised learning
μη επιβλεπομένη μάθηση	Unsupervised learning
Ημι-επιβλεπόμενη μάθηση	Semi-supervised learning
Ενισχυτική Μάθηση	Reinforcement learning
Αυτοεπιβλεπόμενη μάθηση	Self-supervised learning
δέντρα αποφάσεων	Decision trees
χαρακτηριστικό	Feature
συνθήκη	Condition
τυχαία δάση	Random forests
Μηχανή Υποστήριξης Διανυσμάτων	Support Vector Machine
Υπερπροσαρμογή	Overfitting
Υποπροσαρμογή	Underfitting
πίνακας σύγχυσης	Confusion matrix
δυαδική ταξινόμηση	Binary classification
Ορθότητα	accuracy
Ακρίβεια	precision
Ανάκληση	recall
βαθιά μάθηση	deep learning
βαθιά νευρωνικά δίκτυα	deep neural networks
τεχνητός νευρώνας	artificial neuron
επίπεδο εισόδου	input layer
επίπεδο εξόδου	output layer

ακατέργαστα δεδομένα	raw data
τυχαία υπερδειγματοληψία	random oversampling
εργαλειοθήκη ανοιχτού κώδικα	open-source toolkit
σύνολο δεδομένων	dataset
υπερπαραμέτρους	hyperparameters
ενεργειακό κόστος	energy cost