ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

«ΑΝΑΛΥΣΗ ΚΑΙ ΠΡΟΒΛΕΨΗ ΜΕΤΕΩΡΟΛΟΓΙΚΩΝ ΔΕΔΟΜΕΝΩΝ ΜΕΣΩ ΤΗΣ ΔΙΑΔΙΚΑΣΙΑΣ ΑΝΑΚΑΛΥΨΗΣ ΓΝΩΣΗΣ ΣΕ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ»

Ιωάννης Τσαλούμας
ΑΜ: 18390024

Επιβλέπων: Χρήστος Τρούσσας

UNIVERSITY OF WEST ATTICA

FACULTY OF ENGINEERING

DEPARTMENT OF INFORMATICS AND COMPUTER ENGINEERING

DIPLOMA THESIS

« ANALYSIS AND PREDICTION OF METEOROLOGICAL DATA USING THE PROCESS OF KNOWLEDGE DISCOVERY IN DATABASES»

Tsaloumas Ioannis
RN: 18390024

Supervisor: Christos Troussas

Η Διπλωματική Εργασία εξετάστηκε και βαθμολογήθηκε από την κάτωθι τριμελή επιτροπή:

Χρήστος Τρούσσας Επ. Καθηγητής	Ακριβή Κρούσκα Μεταδιδακτορική Ερευνήτρια	Παναγιώτα Τσελέντη ΕΔΙΠ
Christos Troussas Digitally signed by Christos Troussas Date: 2023.07.20 14:59:32 +03'00'	Akrivi Krouska Digitally signed by Akrivi Krouska Date: 2023.07.19 17:27:12 +03'00'	Panagiota Tselenti Digitally signed by Panagiota Tselenti Date: 2023.07.20 16:54:30 +03'00'

ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΔΙΠΛΩΜΑΤΙΚΗΣ ΕΡΓΑΣΙΑΣ

Ο κάτωθι υπογεγραμμένος Ιωάννης Τσαλούμας του Κωνσταντίνου, με αριθμό μητρώου 18390024 φοιτητής του Πανεπιστημίου Δυτικής Αττικής της Σχολής ΜΗΧΑΝΙΚΩΝ του Τμήματος ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ,

δηλώνω υπεύθυνα ότι:

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

Ο Δηλών

Ιωάννης Τσαλούμας

ΕΥΧΑΡΙΣΤΙΕΣ

Με την εκπόνηση της διπλωματικής μου εργασίας κλείνει ένας κύκλος, για να ξεκινήσει κάτι καινούριο. Σε αυτό το σημείο θα ήθελα να ευχαριστήσω την οικογένεια μου που ήταν δίπλα μου με τις θυσίες της όλα αυτά τα χρόνια. Ακόμα θα ήθελα να ευχαριστήσω τους φίλους μου για τις ωραίες στιγμές που έκαναν να φοιτητικά μου χρόνια καλύτερα και ομορφότερα. Τέλος θα ήθελα να ευχαριστήσω τον καθηγητή μου Χρήστο Τρούσσα για τις γνώσεις που μου παρείχε αλλά και την εμπιστοσύνη που έδειξε στο πρόσωπο μου.

ΠΕΡΙΛΗΨΗ

Στην καθημερινή ζωή, οι ανθρώπινες δραστηριότητες, όπως η εργασία, η αναψυχή και η κοινωνικοποίηση, εξαρτώνται σε μεγάλο βαθμό από τις καιρικές συνθήκες. Σωστές προβλέψεις βοηθούν τους ανθρώπους να προετοιμαστούν και να προσαρμοστούν καλύτερα. Με την πάροδο του χρόνου και την εξέλιξη και την άνοδο της τεχνολογίας, η μηχανική μάθηση μπορεί να δώσει λύσεις και αξιόπιστες προβλέψεις. Στη μελέτη αυτή θα ακολουθήσουμε τη διαδικασία ανακάλυψης της γνώσης από βάσεις δεδομένων προκειμένου να εκπαιδεύσουμε διάφορα μοντέλα μηχανικής μάθησης χρησιμοποιώντας την βιβλιοθήκη scikit-learn της γλώσσας προγραμματισμού Python, και το εργαλείο ανοιχτού κώδικα WEKA. Οι αλγόριθμοι μηχανικής μάθησης που θα χρησιμοποιηθούν είναι οι K-κοντινότεροι-γείτονες, η Λογιστική παλινδρόμηση, τα Δέντρα απόφασης, τα Τυχαία δάση, τα Τυχαία Δάση, ο AdaBoost, ο Απλοϊκός Bayes, οι Μηχανές Διανυσμάτων Υποστήριξης και το Νευρωνικό δίκτυο πολλών επιπέδων. Η μεταβλητή "στόχος" παίρνει τιμές "Rain" και "Not Rain". Στην συνέχεια θα αξιολογήσουμε αυτούς τους κατηγοριοποιητές και θα προτείνουμε τον πιο αξιόπιστο βάσει των αποτελεσμάτων της παρούσας μελέτης. Ο κατηγοριοποιητής RandomForest είναι ο πιο βέλτιστος για την πρόβλεψη βροχόπτωσης συγκριτικά με τους υπόλοιπους κατηγοριοποιητές τόσο στο scikit-learn όσο και στο WEKA.

Λέξεις Κλειδιά: πρόβλεψη καιρικών συνθηκών, τεχνητή νοημοσύνη, μηχανική μάθηση, επιστήμη των δεδομένων, επιβλεπόμενη μάθηση, κατηγοριοποίηση, κατηγοριοποιητές, ανακάλυψη γνώσης από βάσεις δεδομένων, Python, Scikit-learn, WEKA.

ABSTRACT

In everyday life, human activities such as work, leisure, and socialization are highly dependent on weather conditions. Accurate predictions help people prepare and adapt better. Over time, with the evolution and advancement of technology, machine learning can provide solutions and reliable forecasts. In this study, we will follow the process of knowledge discovery from databases to train various machine learning models using the scikit-learn library in Python and the open-source tool WEKA. The machine learning algorithms to be used include K-Nearest Neighbors, Logistic Regression, Decision Trees, Random Forests, AdaBoost, Naive Bayes, Support Vector Machines, and Multilayer Perceptron Neural Network. The target variable takes values "Rain" and "Not Rain". Subsequently, we will evaluate these classifiers and propose the most reliable one based on the results of this study. The RandomForest classifier is the most optimal for rainfall prediction compared to the other classifiers, both in scikit-learn and WEKA.

Keywords: weather prediction, artificial intelligence, machine learning, data science, supervised learning, classification, classifiers, knowledge discovery from databases, Python, Scikit-learn, WEKA.

ΠΕΡΙΕΧΟΜΕΝΑ

Πίνακας Εικόνων

Πίνακας Εξισώσεων

Πίνακας Γραφικών Παραστάσεων

Πίνακας Πινάκων

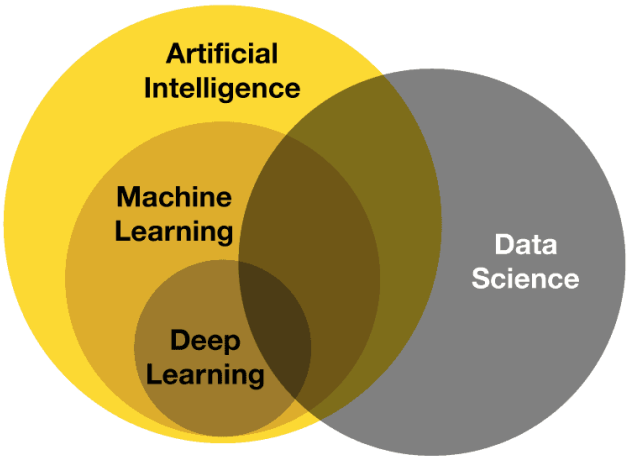
Πίνακας Συντομεύσεων

False Negative	FN
False Positive	FP
k Nearest Neighbors	KNN
Principal Components Analysis	PCA
Support Vector Machines	SVMs
True Negative	TN
True Positive	TP
Knowledge Discovery in Databases	KDD

Κεφάλαιο 1: Εισαγωγή

Σε αυτό το κεφάλαιο θα γίνει εισαγωγή στην διπλωματική εργασία που πραγματεύεται την ανάλυση μετεωρολογικών δεδομένων και την πρόβλεψη τους ακολουθώντας τη διαδικασία ανακάλυψης γνώσης από βάσεις δεδομένων. Αρχικά, θα ορίσουμε την τεχνητή νοημοσύνη και θα παρουσιάσουμε μερικές εφαρμογές που αποτελούν προϊόντα διάφορων ερευνών. Στη συνέχεια θα τονιστεί η ανάγκη δημιουργίας αξιόπιστων μοντέλων πρόβλεψης καιρικών συνθηκών και θα παρουσιαστεί ο λόγος για τον οποίο επιλέχθηκαν δεδομένα από διάφορες περιοχές της Αυστραλίας για την παρούσα διπλωματική εργασία.

Λένε ότι τα δεδομένα αποτελούν το πετρέλαιο του 21ου αιώνα. Καθημερινά ακούμε και διαβάζουμε για εξελίξεις πάνω σε τομείς όπως Big Data, Data Science, Data Mining, Machine Learning που θα μεταβάλλουν τη ζωή μας και θα προσδιορίσουν στο άμεσο μέλλον την καθημερινότητά μας. Αναμφίβολά, η τεχνητή νοημοσύνη (Artificial Intelligence) είναι ένας κλάδος της πληροφορικής που έχει γνωρίσει ραγδαία άνοδο τα τελευταία χρόνια. Ο κλάδος αυτός ασχολείται με τη δημιουργία και την ανάπτυξη μηχανών και λογισμικών που μπορούν να μιμηθούν ή να προσομοιώσουν την ανθρώπινη νοημοσύνη στην επίτευξη λογικής σκέψης, στην επίλυση προβλημάτων, στην αναγνώρισης προτύπων, και στην κατανόηση της γλώσσας και της εκμάθησης από προηγούμενες εμπειρίες.



***Εικόνα 1**: Απεικόνιση πεδίων AI, ML, Deep Learning και Data Science σε σύνολα.*

Εφαρμογές της τεχνητής νοημοσύνης έχουν ήδη αρχίσει να χρησιμοποιούνται όλο ένα και περισσότερο. Ένας από τους τομείς που τα τελευταία χρόνια η τεχνητή νοημοσύνη έχει συνεισφέρει ώστε να γίνουν άλματα είναι η αναγνώριση προσώπου. Στην έρευνα [1] οι συγγραφείς παρουσιάζουν έναν ανιχνευτή προσώπου με μάσκα πραγματικού χρόνου, χρησιμοποιώντας Συνελικτικά Νευρωνικά Δίκτυα (Convolutional Neural Networks - CNNs) κατά την πανδημική περίοδο COVID-19. Άλλες μελέτες του κλάδου έχουν στόχο την αναγνώριση συγκεκριμένων συναισθημάτων από διάφορες

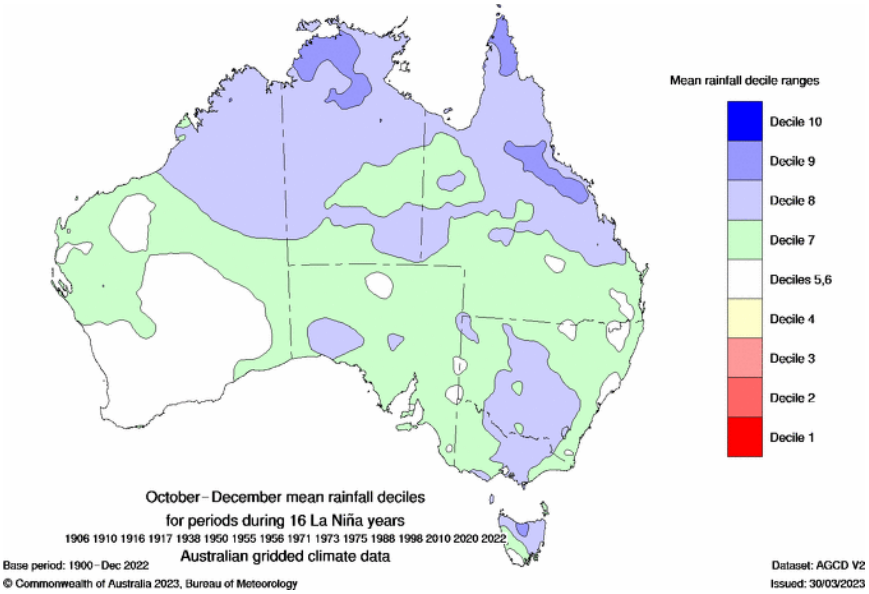
μορφές [2][3], όπως για παράδειγμα μέσω ενός προσώπου ή ακόμα και μέσω γραπτού λόγου που συναντάται καθημερινά σε αναρτήσεις στα μέσα κοινωνικής δικτυωτής [4-6].

Άλλος ένας τομέας που η τεχνητή νοημοσύνη μπορεί να συνεισφέρει τα μέγιστα είναι η εκπαίδευση. Το παιχνίδι quiz για κινητά τηλέφωνα που παρουσιάζεται στην έρευνα [7] χρησιμοποιεί τεχνητή νοημοσύνη για να προσαρμόζει το περιεχόμενο και τη δυσκολία των ερωτήσεων στις ανάγκες του κάθε φοιτητή, προωθώντας την εκμάθηση και τη συνεργασία μεταξύ των φοιτητών τριτοβάθμιας εκπαίδευσης στον τομέα του προγραμματισμού. Σε άλλη έρευνα πάλι [8], η εξόρυξη δεδομένων χρησιμοποιείται για να αναλύσει τα δεδομένα που προκύπτουν από την αξιολόγηση των φοιτητών, βοηθώντας τους εκπαιδευτικούς να κατανοήσουν καλύτερα τις αδυναμίες και τις ανάγκες των μαθητών τους και να προσαρμόσουν αναλόγως τις μεθόδους διδασκαλίας. Τέλος, άλλη έρευνα [9] εξετάζει πώς μπορούν τα ευφυή συστήματα διδασκαλίας (Intelligent Tutoring Systems) να ενισχύσουν την αποτελεσματικότητα της εκπαίδευσης μέσω της προσαρμογής και της μοντελοποίησης γνωσιακής διάγνωσης, προτείνοντας ένα νέο πλαίσιο που συνδυάζει επαυξημένη πραγματικότητα και παιδαγωγικές τεχνικές για τη βελτίωση της αναγνωστικής κατανόησης στην ειδική αγωγή και εκπαίδευση.

Εκτός όμως από την εκπαίδευση και την αναγνώριση προσώπου, η τεχνητή νοημοσύνη μπορεί να συνεισφέρει και σε άλλους τομείς της καθημερινότητας, ένας από τους οποίους είναι το περιβάλλον. Στην καθημερινή ζωή, οι ανθρώπινες δραστηριότητες, όπως η εργασία, η αναψυχή και η κοινωνικοποίηση, εξαρτώνται σε μεγάλο βαθμό από τις καιρικές συνθήκες. Σωστές προβλέψεις βοηθούν τους ανθρώπους να προετοιμαστούν και να προσαρμοστούν ανάλογα. Στη γεωργία, η πρόβλεψη του καιρού είναι ζωτικής σημασίας, καθώς οι καλλιέργειες επηρεάζονται άμεσα από τις καιρικές συνθήκες. Συνεπώς ακριβείς προβλέψεις μπορούν να βοηθήσουν τους αγρότες να προγραμματίσουν τη φροντίδα και τη συγκομιδή των καλλιεργειών τους, μειώνοντας τις απώλειες και βελτιώνοντας την παραγωγή. Στον τομέα της ενέργειας, η πρόβλεψη του καιρού είναι κρίσιμη για την παραγωγή και κατανάλωση ενέργειας, ειδικά όταν πρόκειται για ανανεώσιμες πηγές όπως η ηλιακή και αιολική ενέργεια. Η γνώση των καιρικών συνθηκών βοηθά τις εταιρείες ενέργειας να προβλέπουν τη ζήτηση και να διαχειρίζονται την παραγωγή πιο αποδοτικά. Στον τομέα των μεταφορών, οι προβλέψεις καιρού επηρεάζουν τη λειτουργία των οδικών, σιδηροδρομικών, αεροπορικών και ναυτιλιακών μεταφορών άρα ακριβείς καιρικές προβλέψεις βοηθούν στην προετοιμασία και συντήρηση των υποδομών, στην αποφυγή ατυχημάτων και στη βελτίωση της ασφάλειας των επιβατών. Επίσης, επιτρέπουν στις αεροπορικές και ναυτιλιακές εταιρείες να προγραμματίζουν τα δρομολόγια τους πιο αποτελεσματικά, αποφεύγοντας τυχόν καθυστερήσεις. Τέλος, η πρόβλεψη του καιρού είναι καίρια για την αντιμετώπιση καιρικών φαινόμενων όπως καταιγίδες, πλημμύρες,

ξηρασίες και καύσωνες, που μπορούν να έχουν καταστροφικές επιπτώσεις στο περιβάλλον, ειδικά κατά την περίοδο και κλιματικής κρίσης που διανύουμε.

Όπως γίνεται επομένως εύκολα αντιληπτό, υπάρχει άμεση ανάγκη για δημιουργία αξιόπιστων μοντέλων που θα μπορούν να προβλέψουν τις καιρικές συνθήκες ανά τόπους σε πραγματικό χρόνο. Στην παρούσα διπλωματική θα μελετήσουμε πρόβλεψη βροχόπτωσης από δεδομένα διάφορων πόλεων της Αυστραλίας. Η Αυστραλία είναι μια χώρα που έχει έρθει αντιμέτωπη με τις συνέπειες της κλιματικής αλλαγής καθώς είναι ευάλωτη σε πλημμύρες, ιδίως κατά τις περιόδους εκδήλωσης του φαινομένου La Niña που συνδέεται στενά με ισχυρές βροχοπτώσεις και πλημμύρες στα βόρεια και ανατολικά τμήματα της χώρας. Το φαινόμενο La Niña προκαλείται όταν οι ισημερινοί άνεμοι στον Ειρηνικό Ωκεανό γίνονται πιο δυνατοί με αποτέλεσμα να αλλάζουν τα θαλάσσια ρεύματα τραβώντας τις ψυχρότερες μάζες νερού μακριά από τις ειρηνικές ακτές της Νότιας Αμερικής. Αυτό έχει ως αποτέλεσμα την ψύξη του κεντρικού και ανατολικού Ειρηνικού Ωκεανού. Η αλλαγή κατεύθυνσης των ανέμων συχνά οδηγεί επίσης στη συσσώρευση θερμού νερού στη βόρεια πλευρά της Αυστραλίας, επηρεάζοντας το κλίμα της περιοχής. Η αύξηση των θερμοκρασιών του ωκεανού στον δυτικό Ειρηνικό Ωκεανό δημιουργεί ευνοϊκές συνθήκες για αέρηδες, ανάπτυξη συννεφιάς και έντονη βροχόπτωση στα ανατολικά και βόρεια τμήματα της Αυστραλίας [10][11]. Η σχέση μεταξύ της έντασης της La Niña και της βροχόπτωσης είναι στενά συνδεδεμένη. Στο λεκανοπέδιο του Murray–Darling, η μέση βροχόπτωση κατά τη διάρκεια όλων των 18 ετών που εκδηλώθηκε La Niña από το 1900 ήταν 22% υψηλότερη από το μακροπρόθεσμο μέσο όρο, με τις σοβαρές πλημμύρες του 1955, 1988, 1998 και 2010 να συνδέονται όλες με το La Niña. Τα πιο βροχερά έτη στα χρονικά για την Αυστραλία σημειώθηκαν κατά τη διάρκεια των ισχυρών γεγονότων της La Niña το 2010-2012 και το 1974. Το φαινόμενο La Niña κατά τα έτη 2010-12 ήταν ιδιαίτερα καταστροφικό, με εκτεταμένες πλημμύρες σε όλη την Αυστραλία [11].



***Εικόνα 2**: Βροχόπτωση στην Αυστραλία κατά τη διάρκεια των μηνών Οκτωβρίου έως Δεκεμβρίου, κατά τη διάρκεια δεκατριών ισχυρών γεγονότων La Niña.*

Το φαινόμενο δεν πραγματοποιείται σε καθορισμένες ημερομηνίες κάθε χρόνο, γεγονός που κάνει την πρόληψη του δυσκολότερη. Το μοτίβο της θερμοκρασίας της επιφάνειας της θάλασσας στον Ειρηνικό Ωκεανό ορίζουν αν θα συμβεί ένα έντονο ή ασθενές La Niña. Ωστόσο, τα περισσότερα La Niña συμβαίνουν κάθε 2-7 χρόνια, ενώ σπάνια μπορεί να συμβεί δύο φορές σε μια χρονιά. Εφόσον εμφανιστεί, το La Niña, παραμένει για τουλάχιστον 5 μήνες.

***Εικόνα 3**: Οι καταστροφικές πλημμύρες στην Αυστραλία το 2022.*

Στην προηγούμενη ενότητα καταγράφηκαν οι επιπτώσεις που μπορεί να προκαλέσει η εκδήλωση του φαινομένου La Niña στην Αυστραλία. Για τον λόγο αυτό καθίσταται αναγκαία η εύρεση ενός αξιόπιστου μοντέλου που θα μπορεί να προσφέρει ακριβείς προβλέψεις για τις καιρικές συνθήκες ανά περιοχή της χώρας. Με την άνοδο της τεχνολογίας, η μηχανική μάθηση μπορεί να βοηθήσει στην πρόβλεψη καιρικών συνθηκών και πιο συγκεκριμένα της βροχόπτωσης έτσι ώστε να ληφθούν έγκαιρα μέτρα πολιτικής προστασίας που θα αμβλύνουν τον κίνδυνο και τις συνέπειες από πλημύρες και καταστροφές.

Στην παρούσα διπλωματική θα αναλύσουμε μετεωρολογικά δεδομένα από διάφορες περιοχές της Αυστραλίας και θα ακολουθήσουμε τη διαδικασία ανακάλυψης γνώσης σε βάσεις δεδομένων έτσι

ώστε να προτείνουμε το πιο αξιόπιστο μοντέλο/αλγόριθμο μηχανικής μάθησης. Η διπλωματική έχει χωριστεί σε 5 κεφάλαια ξεκινώντας από την εισαγωγή που αναδεικνύεται η ανάγκη ύπαρξης αξιόπιστων μοντέλων πρόβλεψης καιρικών συνθηκών εστιάζοντας στην Αυστραλία ως μια χώρα που έχει πληγεί από την κλιματική αλλαγή στο πρόσφατο παρελθόν.

Η δομή των υπολοίπων κεφαλαίων είναι η ακόλουθη:

- Κεφάλαιο 2: Θεωρητικό Υπόβαθρο και Ανασκόπηση Βιβλιογραφίας. Στο κεφάλαιο αυτό, αρχικά ορίζεται η μηχανική μάθηση και τα διάφορα είδη της και συνέχεια εμπεριέχεται μια συνοπτική ανάλυση όλων των αλγορίθμων μηχανικής μάθησης που θα χρησιμοποιήσουμε για την εξόρυξη δεδομένων που θα ακολουθήσει. Ακόμη, θα αναφερθούμε διεξοδικά στην διαδικασία ανακάλυψης γνώσης από βάσεις δεδομένων που θα ακολουθήσουμε και τα επιμέρους στάδια που αυτή περιλαμβάνει (Συλλογή, Προεπεξεργασία, Μετασχηματισμό, Εξόρυξη Δεδομένων και Αξιολόγηση Γνώσης), αλλά και τις μετρικές βάσει των οποίων θα αξιολογήσουμε κάθε μοντέλο.
- Κεφάλαιο 3: Μεθοδολογία. Στο κεφάλαιο αυτό θα παρουσιαστεί το περιβάλλον του WEKA και η γλώσσα προγραμματισμού Python που θα χρησιμοποιηθεί. Θα παρουσιαστούν οι βιβλιοθήκες της γλώσσας προγραμματισμού που θα χρησιμοποιήσουμε και το σύνολο δεδομένων της Αυστραλίας. Ακόμη στο κεφάλαιο αυτό θα παρουσιαστεί η προεπεξεργασία και ο μετασχηματισμός δεδομένων καθώς και οι παράμετροι που θα χρησιμοποιήσουμε για όλους τους κατηγοριοποιητές.
- Κεφάλαιο 4: Παρουσίαση Αποτελεσμάτων. Στο κεφάλαιο αυτό θα παρουσιαστούν τα αποτελέσματα της εφαρμογής της KDD διαδικασίας με Python και με χρήση του εργαλείου WEKA. Στην συνέχεια θα αξιολογήσουμε τους αλγορίθμους σε κάθε περιβάλλον ξεχωριστά αλλά και συνολικά αναδεικνύοντας τον πιο αποτελεσματικό.
- Κεφάλαιο 5: Συμπεράσματα. Στο κεφάλαιο αυτό θα παρουσιαστούν τα συμπεράσματα και τα πορίσματα που θα προκύψουν από την εκπόνηση της παρούσας διπλωματικής εργασίας. Παράλληλα θα δοθούν και κάποιες μελλοντικές επεκτάσεις.
- Στο τέλος παρουσιάζεται η βιβλιογραφία.

Κεφάλαιο 2: Θεωρητικό Υπόβαθρο και Ανασκόπηση της Βιβλιογραφίας

Στο κεφάλαιο αυτό θα ορίσουμε τι είναι μηχανική μάθηση και θα παρουσιάσουμε τα βασικά είδη της. Ακόμη θα παρουσιάσουμε την Διαδικασία Ανακάλυψης Γνώσεις από Βάσεις Δεδομένων που περιλαμβάνει την Συλλογή, την Προ-επεξεργασία, το Μετασχηματισμό, την Εξόρυξη Δεδομένων και την Αξιολόγηση Γνώσης. Στη συνέχεια στο κεφάλαιο αυτό θα παρουσιαστεί το θεωρητικό υπόβαθρο των αλγορίθμων μηχανικής μάθησης, ενώ τέλος, θα αναφερθούμε και διάφορες μετρικές αξιολόγησης του κάθε μοντέλου βάσει των οποίων θα γίνει η συγκριτική μελέτη για την ανάδειξη του αποδοτικότερου.

2.1 Τι είναι μηχανική μάθηση

Η μηχανική μάθηση είναι μια υποκατηγορία της τεχνητής νοημοσύνης (AI) που ασχολείται με την ανάπτυξη αλγορίθμων και μοντέλων που επιτρέπουν στους υπολογιστές να "μάθουν" από δεδομένα χωρίς να χρειάζεται να προγραμματίζονται ρητά [12]. Με τη μηχανική μάθηση, οι υπολογιστές μπορούν να βελτιώσουν την απόδοσή τους καθώς παρέχονται νέα δεδομένα και εμπειρίες. Ο Tom Mitchell, στο βιβλίο του "Machine Learning" [12], παρέθεσε έναν διάσημο ορισμό για τη μηχανική μάθηση: "Λέμε ότι ένα πρόγραμμα υπολογιστή μαθαίνει από μια εμπειρία E σχετικά με κάποιο καθήκον T και κάποιο μέτρο απόδοσης P, αν η απόδοσή του στο καθήκον T, μετρημένη από το P, βελτιώνεται με την εμπειρία E." Ο ορισμός αυτός υπογραμμίζει την αναγκαιότητα των τριών κύριων συστατικών στη μηχανική μάθηση: την εμπειρία (προηγούμενα δεδομένα), το καθήκον (πρόβλημα που προσπαθούμε να λύσουμε) και το μέτρο απόδοσης (πώς αξιολογούμε την επίδοση του μοντέλου). Ο ορισμός του Mitchell εξηγεί πώς τα συστήματα μηχανικής μάθησης επιδιώκουν να βελτιώσουν την απόδοσή τους μέσα από την εμπειρία και τη διαδικασία εκμάθησης. Ένας άλλος ορισμός για τη μηχανική μάθηση προέρχεται από τον Christopher Bishop, στο βιβλίο του "Pattern Recognition and Machine Learning" [13]. Ο Bishop ορίζει τη μηχανική μάθηση ως: "Ένα σύνολο μεθόδων που μπορούν αυτόματα να αναγνωρίζουν πολύπλοκα μοτίβα στα δεδομένα και να χρησιμοποιούν τα αναγνωρισμένα μοτίβα για να κάνουν προβλέψεις σε νέα δεδομένα.". Ο ορισμός αυτός επικεντρώνεται στην ικανότητα των αλγορίθμων μηχανικής μάθησης να ανακαλύπτουν και να αναγνωρίζουν πολύπλοκα μοτίβα και σχέσεις μέσα στα δεδομένα, που στη συνέχεια μπορούν να χρησιμοποιηθούν για να κάνουν προβλέψεις ή να λαμβάνουν αποφάσεις για νέες, άγνωστες καταστάσεις.

2.2 Είδη Μηχανικής Μάθησης

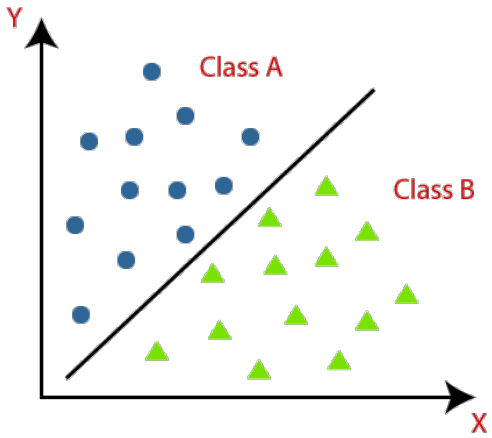
Υπάρχουν τρεις βασικοί τύποι μηχανικής μάθησης που αφορούν την επιβλεπόμενη μάθηση (Supervised Learning), την μη επιβλεπόμενη μάθηση (Unsupervised Learning) και την ενισχυτική μάθηση (Reinforcement Learning). Η επιβλεπόμενη μάθηση περιλαμβάνει ως κύριες μεθόδους την κατηγοριοποίηση και την παλινδρόμηση, ενώ η μη επιβλεπόμενη μάθηση το μετασχηματισμό και τη συσταδοποίηση, έννοιες οι οποίες αναλύονται στη συνέχεια. Η ενισχυτική μάθηση ασχολείται κυρίως με διάφορες οντότητες που ονομάζονται πράκτορες που παίρνουν τις αποφάσεις τους από το περιβάλλον για να εκτελέσουν κάποια ενέργεια.

2.2.1 Επιβλεπόμενη Μάθηση

Η επιβλεπόμενη μάθηση είναι μια από τις πιο κοινές μεθόδους που χρησιμοποιούνται στη μηχανική μάθηση όταν θέλουμε να προβλέψουμε ένα σίγουρο αποτέλεσμα από μία δεδομένη είσοδο [12]. Οι αλγόριθμοι μηχανικής μάθησης εκπαιδεύονται με ένα σύνολο δεδομένων που περιλαμβάνει την είσοδο και την αντίστοιχη επιθυμητή έξοδο (ετικέτες). Στόχος των αλγορίθμων αποτελεί η εύρεση της συνάρτησης που συνδέει τις εισόδους με τις εξόδους, ώστε να μπορούν να κάνουν προβλέψεις για νέα, ανεξάρτητα δεδομένα. Σε αυτό το τύπο μηχανικής μάθησης ανήκει η κατηγοριοποίηση και η παλινδρόμηση. Οι αλγόριθμοι k πλησιέστερων γειτόνων (k Nearest Neighbors – kNN), η γραμμική παλινδρόμηση (Linear Regression), η λογιστική παλινδρόμηση (Logistic Regression), τα Δέντρα Απόφασης (Decision Trees), τα Τυχαία Δάση (Random Forest), οι Μηχανές Διανυσμάτων Υποστήριξης (Support Vector Machines), η μάθηση κατά Bayes και τα νευρωνικά δίκτυα (Neural Networks) αποτελούν παραδείγματα αλγορίθμων που εφαρμόζουν επιβλεπόμενη μάθηση.

2.2.1.1 Κατηγοριοποίηση

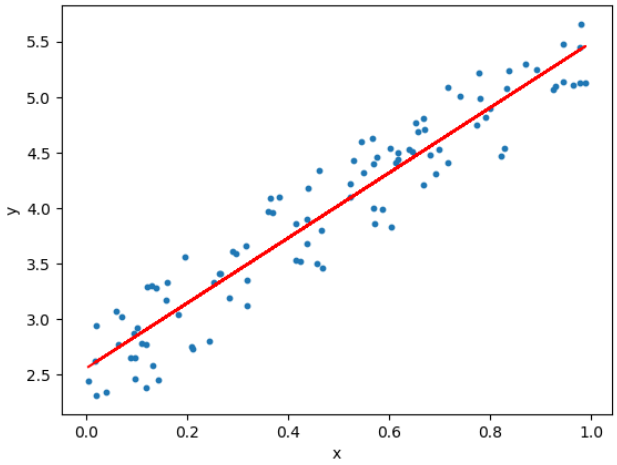
Η κατηγοριοποίηση είναι η διαδικασία κατά την οποία ένα μοντέλο εκπαιδεύεται για να αναγνωρίζει και να ταξινομεί δεδομένα σε προκαθορισμένες κατηγορίες. Τα προβλήματα κατηγοριοποίησης είναι προβλήματα επιβλεπόμενης μάθησης, όπου το μοντέλο μαθαίνει από ένα σύνολο εκπαίδευσης που περιέχει παραδείγματα μαζί με τις αντίστοιχες ετικέτες (κατηγορίες) τους. Κατά την εκπαίδευση, το μοντέλο προσπαθεί να κατανοήσει τα χαρακτηριστικά και τις σχέσεις μεταξύ των δεδομένων, ώστε να μπορεί να κάνει προβλέψεις για την κατηγορία των νέων δεδομένων που δεν έχουν επισημανθεί ακόμα. Διαδεδομένοι μηχανικής μάθησης που μπορούν να χρησιμοποιηθούν για προβλήματα κατηγοριοποίησης είναι ο αλγόριθμος k-πλησιέστερων γειτόνων (k-NN), η λογιστική παλινδρόμηση, τα δέντρα αποφάσεων, τα τυχαία δάση και τα νευρωνικά δίκτυα.



Εικόνα 4: Κατηγοριοποίηση

2.2.1.2 Παλινδρόμηση

Η παλινδρόμηση είναι μια διαδικασία στην επιβλεπόμενη μάθηση, κατά την οποία το μοντέλο εκπαιδεύεται για την πρόβλεψη συνεχών τιμών αντί για κατηγορίες, όπως στην κατηγοριοποίηση. Απώτερος σκοπός του μοντέλου αποτελεί η εύρεση μιας σχέσης μεταξύ των ανεξάρτητων μεταβλητών (χαρακτηριστικά) και της εξαρτημένης μεταβλητής (στόχος), ώστε να μπορεί να προβλέψει την τιμή της εξαρτημένης μεταβλητής για νέα δεδομένα. Γνωστοί αλγόριθμοι παλινδρόμησης είναι η γραμμική παλινδρόμηση, η πολυώνυμη παλινδρόμηση, η λογιστική παλινδρόμηση καθώς και η Ridge και Lasso παλινδρόμηση. Αυτοί οι αλγόριθμοι χρησιμοποιούν διαφορετικές τεχνικές για να προσαρμόσουν το μοντέλο στα δεδομένα εκπαίδευσης και να ελαχιστοποιήσουν το σφάλμα πρόβλεψης.



Εικόνα 5: Γραμμική Παλινδρόμηση

2.2.2 Μη Επιβλεπόμενη Μάθηση

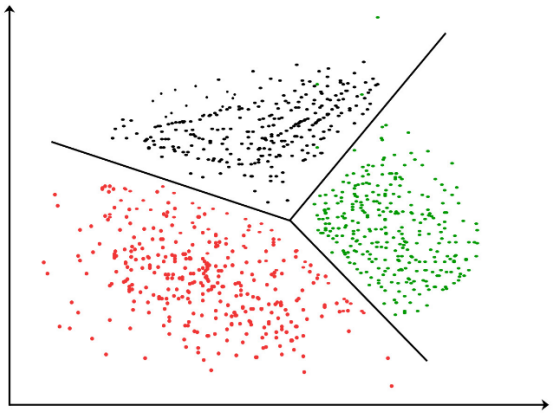
Η μη επιβλεπόμενη μάθηση είναι ένας τύπος μηχανικής μάθησης κατά τον οποίο το μοντέλο εκπαιδεύεται με δεδομένα που δεν έχουν προκαθορισμένες ετικέτες. Σε αντίθεση με την επιβλεπόμενη μάθηση, η μη επιβλεπόμενη μάθηση δεν έχει "σωστές" απαντήσεις για να καθοδηγήσει το μοντέλο κατά την εκπαίδευση. Αντίθετα, το μοντέλο εστιάζει στην ανακάλυψη των κρυφών δομών, σχέσεων ή μοτίβων στα δεδομένα με βάση τα χαρακτηριστικά τους. Όπως και στη επιβλεπόμενη μάθηση, έτσι και στη μη επιβλεπόμενη μάθηση υπάρχουν δύο υποκατηγορίες, ο μετασχηματισμός δεδομένων και συσταδοποίηση. Αλγόριθμοι μη επιβλεπόμενης μάθησης είναι η ανάλυση κυρίων συνιστωσών (Principal Components Analysis – PCA), ο K-Means, ο PAM (Partitioning Around Medoids) και ο DBSCAN.

2.2.2.1 Μετασχηματισμός

Η χρήση αλγορίθμων για τη δημιουργία νέων αναπαραστάσεων του συνόλου δεδομένων μας ονομάζεται μετασχηματισμός (Transformation) δεδομένων. Η πιο κοινή εφαρμογή αλγορίθμων μετασχηματισμού είναι η μείωση διαστάσεων (Dimensionality Reduction). Η μείωση διαστάσεων είναι μια τεχνική που στοχεύει στην απλοποίηση των δεδομένων, μειώνοντας τον αριθμό των χαρακτηριστικών ή των μεταβλητών που χρησιμοποιούνται για την αναπαράστασή τους. Πραγματοποιείται κυρίως κατά την προ-επεξεργασία των δεδομένων και έχει ως στόχο τη συμπίεση των δεδομένων διατηρώντας τη σχετική πληροφορία, καθώς η ύπαρξη πολλών διαστάσεων μπορεί να αποτελέσει πρόκληση όταν υπάρχει περιορισμένος χώρος αποθήκευσης και υπολογιστικής απόδοσης των αλγορίθμων που εφαρμόζονται. Αλγόριθμοι που χρησιμοποιούνται για μείωση διαστάσεων είναι η ανάλυση κυρίων συνιστωσών, η γραμμική διαχωριστική ανάλυση, η κανονικοποίηση και η τυποποίηση.

2.2.2.2 Συσταδοποίηση

Η συσταδοποίηση (clustering) είναι μια τεχνική μη επιβλεπόμενης μάθησης, η οποία αποσκοπεί στην οργάνωση των δεδομένων σε ομάδες ή συστάδες, με βάση την ομοιότητα ή τη διαφορά των χαρακτηριστικών τους. Οι αλγόριθμοι συσταδοποίησης εργάζονται χωρίς προκαθορισμένες ετικέτες ή κατηγορίες, αλλά προσπαθούν να ανακαλύψουν τη δομή των δεδομένων με βάση την αυτόματη ανίχνευση μοτίβων στα δεδομένα. Παραδείγματα τέτοιων αλγόριθμων είναι ο K-Means, και ο DBSCAN.



***Εικόνα 6**: Συσταδοποίηση (Clustering)*

2.2.3 Ενισχυτική Μάθηση

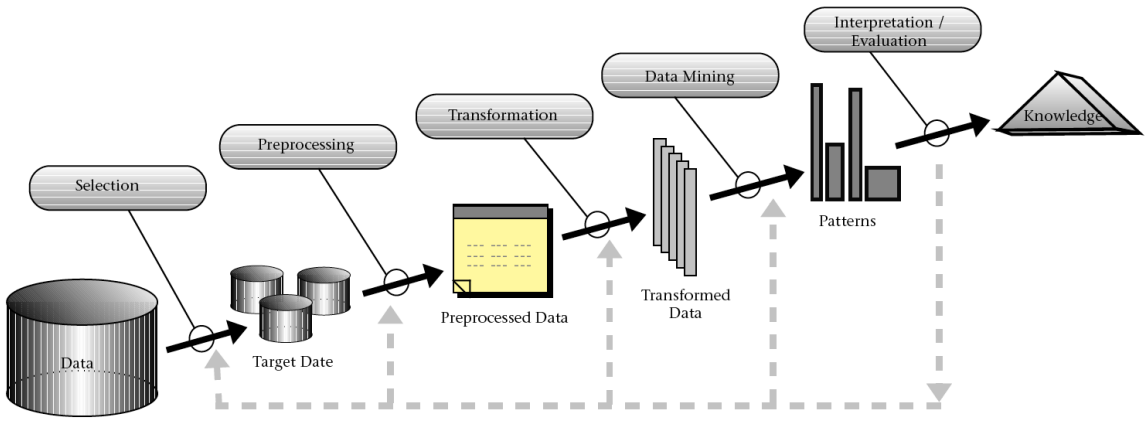
Η ενισχυτική μάθηση (reinforcement learning) αποτελεί ένα παρακλάδι της μηχανικής μάθησης που επικεντρώνεται στην εκμάθηση των στρατηγικών και των αποφάσεων που πρέπει να λαμβάνει ένας πράκτορας για να επιτύχει στόχους σε ένα περιβάλλον. Σε αντίθεση με την επιβλεπόμενη και τη μη επιβλεπόμενη μάθηση, η ενισχυτική μάθηση βασίζεται στην αλληλεπίδραση μεταξύ του πράκτορα και του περιβάλλοντος. Στην ενισχυτική μάθηση, ο πράκτορας λαμβάνει πληροφορίες από το περιβάλλον μέσω καταστάσεων (states) και παίρνει αποφάσεις εκτελώντας ενέργειες (actions). Ως ανταμοιβή, ο πράκτορας λαμβάνει ένα σήμα ανταμοιβής (reward), το οποίο αντανακλά την ποιότητα των εκτελεσμένων ενεργειών. Στόχος είναι ο πράκτορας να βελτιώσει τη στρατηγική του (policy) για να μεγιστοποιήσει τη συνολική ανταμοιβή που λαμβάνει με την πάροδο του χρόνου.

2.3 Διαδικασία Ανακάλυψης Γνώσης σε Βάσεις Δεδομένων

Σε αυτή την παράγραφο θα αναφερθούμε στη διαδικασία ανακάλυψης γνώσης σε βάσεις δεδομένων (Knowledge Discovery in Databases - KDD) και στα επιμέρους βήματα που αυτή περιλαμβάνει. Η ανακάλυψη γνώσης σε βάσεις δεδομένων είναι μια διαδικασία που περιλαμβάνει την εξαγωγή χρήσιμων πληροφοριών, προτύπων ή νέας γνώσης από μεγάλα σύνολα δεδομένων [14]. Η διαδικασία KDD είναι διαμορφωμένη σε πέντε βασικά στάδια, που είναι τα εξής:

1. **Επιλογή δεδομένων (Data Selection)**: Σε αυτό το στάδιο, επιλέγεται το σύνολο δεδομένων που θα χρησιμοποιηθεί για την ανάλυση.
2. **Προεπεξεργασία δεδομένων (Data Preprocessing)**: Σε αυτό το στάδιο, τα δεδομένα καθαρίζονται, προκειμένου να βελτιωθεί η ποιότητα και η αποτελεσματικότητα της ανάλυσης. Η προεπεξεργασία μπορεί να περιλαμβάνει την αντιμετώπιση των απουσιαζουσών τιμών, την απαλοιφή θορύβου, την εξάλειψη διπλοτύπων, και την κανονικοποίηση των δεδομένων.

3. **Μετασχηματισμός δεδομένων (Data Transformation)**: Σε αυτό το στάδιο, τα δεδομένα μετασχηματίζονται περαιτέρω ώστε να ταιριάζουν καλύτερα στις τεχνικές εξόρυξης δεδομένων που θα χρησιμοποιηθούν. Ο μετασχηματισμός μπορεί να περιλαμβάνει την κατασκευή νέων χαρακτηριστικών (feature engineering), την αναπαράσταση των δεδομένων, ή την αγκλίδωση (aggregation) των δεδομένων.
4. **Εξόρυξη δεδομένων (Data Mining)**: Στο κεντρικό στάδιο της διαδικασίας KDD, οι αλγόριθμοι εφαρμόζονται για να ανακαλύψουν κρυμμένα πρότυπα, συσχετίσεις, ή άλλες χρήσιμες πληροφορίες από τα δεδομένα. Οι τεχνικές που χρησιμοποιούνται εδώ περιλαμβάνουν την ομαδοποίηση (clustering), την ταξινόμηση (classification), την παλινδρόμηση (regression), την εξαγωγή κανόνων συσχέτισης (association rule learning) και την ανάλυση συνδυαστικών προτύπων (sequential pattern analysis).
5. **Αξιολόγηση/Ερμηνεία (Evaluation/Interpretation)**: Στο τελικό στάδιο, τα αποτελέσματα της εξόρυξης δεδομένων αξιολογούνται και ερμηνεύονται για να επιβεβαιωθεί ότι είναι χρήσιμα και κατανοητά. Αυτό μπορεί να περιλαμβάνει την επικύρωση των προτύπων που εντοπίστηκαν, τη σύγκριση με γνωστές πληροφορίες ή την επικοινωνία των αποτελεσμάτων σε άλλους εμπειρογνώμονες.



***Εικόνα 7**: Τα βήματα της KDD διαδικασίας.*

2.4 Θεωρία αλγορίθμων μηχανικής μάθησης

Στην ενότητα αυτή θα παρουσιαστεί η θεωρητική ανασκόπηση των αλγορίθμων μηχανικής μάθησης/κατηγοριοποιητών που θα χρησιμοποιηθούν στην συνέχεια. Συγκεκριμένα θα παρουσιαστεί το θεωρητικό υπόβαθρο των αλγορίθμων k - κοντινότερων γειτόνων (k-NN), της Λογιστικής

Παλινδρόμησης (Logistic Regression), των Δέντρων Απόφασης (Decision Trees), των Τυχαίων Δασών (Random Forest), του AdaBooost, του απλοϊκού Bayes (Naïve Bayes), των Μηχανών Διανυσμάτων Υποστήριξης (SVMs) και των Νευρωνικών Δικτύων (Neural Networks).

2.4.1 Κ – Πλησιέστεροι Γείτονες

Ο αλγόριθμος k-πλησιέστερων γειτόνων (k-NN) είναι ένας αλγόριθμος ταξινόμησης που βασίζεται στην αρχή ότι αντικείμενα που είναι κοντά μεταξύ τους στο χώρο των χαρακτηριστικών έχουν παρόμοιες ιδιότητες. Συγκεκριμένα, ο αλγόριθμος k-NN λειτουργεί ως εξής:

1. Δέχεται ως είσοδο ένα σύνολο δεδομένων εκπαίδευσης με ετικέτες.
2. Ορίζει έναν αριθμό k, ο οποίος αντιπροσωπεύει τον αριθμό των πλησιέστερων γειτόνων που θα εξεταστούν.
3. Για κάθε νέο δείγμα που πρέπει να ταξινομηθεί ή να προβλεφθεί:
    A. Υπολογίζει τις αποστάσεις μεταξύ του νέου δείγματος και όλων των δειγμάτων εκπαίδευσης. Η ευκλείδεια απόσταση είναι η πιο συνηθισμένη μετρική απόστασης, αλλά μπορούν να χρησιμοποιηθούν και άλλες μετρικές.
    B. Επιλέγει τα k πλησιέστερα δείγματα (γείτονες) με βάση κάποια από τις μετρικές που θα αναλυθεί στη συνέχεια.
    C. Για προβλήματα ταξινόμησης, καθορίζει την κλάση του νέου δείγματος με βάση την πλειοψηφία των κλάσεων των k πλησιέστερων γειτόνων. Για παράδειγμα, εάν έχουμε k = 3 και οι τρεις πλησιέστεροι γείτονες ανήκουν στις κλάσεις A, A και B, το νέο δείγμα θα ταξινομηθεί ως κλάση A, επειδή η κλάση A έχει την πλειοψηφία των γειτόνων.
    D. Για προβλήματα παλινδρόμησης, καθορίζει την προβλεπόμενη τιμή του νέου δείγματος υπολογίζοντας τον μέσο όρο (ή άλλη κατάλληλη στατιστική) των τιμών των k πλησιέστερων γειτόνων.

Στον αλγόριθμο k-NN, η μέτρηση της απόστασης ανάμεσα στα δείγματα είναι ζωτικής σημασίας για την εύρεση των πλησιέστερων γειτόνων. Υπάρχουν διάφορες μετρικές απόστασης που μπορούν να χρησιμοποιηθούν στον k-NN. Μερικές από τις πιο δημοφιλείς τεχνικές είναι:

1. **Ευκλείδεια απόσταση (Euclidean distance)**: Η πιο κοινή μετρική απόστασης, υπολογίζει την απόσταση μεταξύ δύο σημείων στον ευκλείδειο χώρο. Για δύο διανύσματα x και y στον n-διάστατο χώρο, η ευκλείδεια απόσταση υπολογίζεται ως:

$$d(x,y) = \sqrt{\sum_{i=1}^{n}(x_i+y_i)^2}$$

***Εξίσωση 1**: Ευκλείδεια Απόσταση*

2. **Απόσταση Manhattan (Manhattan distance)**: Η απόσταση Manhattan υπολογίζει την απόσταση μεταξύ δύο σημείων με βάση την κατακόρυφη και την οριζόντια απόσταση. Για δύο διανύσματα x και y στον n-διάστατο χώρο, η απόσταση Μανχάταν υπολογίζεται ως:

$$d(x,y) = \sum_{1}^{n} |x_i - y_i|$$

***Εξίσωση 2**: Απόσταση Manhattan*

3. **Απόσταση Chebyshev (Chebyshev Distance)**: Η απόσταση Chebyshev υπολογίζει τη μέγιστη απόσταση μεταξύ δύο σημείων κατά συντεταγμένη. Για δύο διανύσματα x και y στον n-διάστατο χώρο, η απόσταση Chebyshev υπολογίζεται ως:

$$d(x,y) = max_{i=1,\dots,n}|x_i - y_i|$$

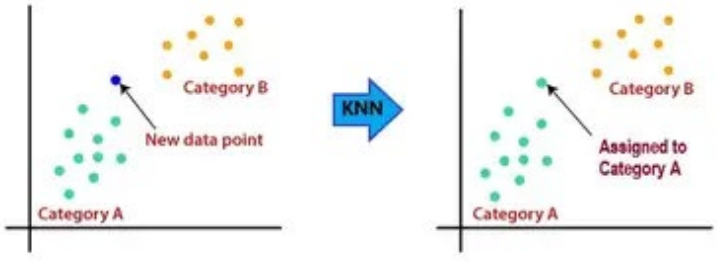
***Εξίσωση 3**: Απόσταση Chebyshev*

4. **Απόσταση Minkowski (Minkowski Distance)**: Η απόσταση Minkowski είναι μια γενίκευση των αποστάσεων Ευκλείδεια, Manhattan και Chebyshev και καλύπτει ένα ευρύ φάσμα αποστάσεων. Χρησιμοποιώντας έναν παράγοντα p, η απόσταση Minkowski υπολογίζεται ως:

$$d(x,y) = (\sum_{i=1}^{n} |x_i - y_i|^p)^{1/p}$$

***Εξίσωση 4**: Απόσταση Minkowski*

Όταν p=2, η απόσταση Minkowski είναι ίση με την Ευκλείδεια απόσταση, ενώ όταν p=1, είναι ίση με την απόσταση Manhattan.



***Εικόνα 8**: Αλγόριθμος KNN*

2.4.2 Λογιστική Παλινδρόμηση

Η λογιστική παλινδρόμηση είναι ένα μοντέλο κατηγοριοποίησης που επινοήθηκε το 1958 από τον στατιστικολόγο David Cox και χρησιμοποιείται ευρέως στην εξόρυξη δεδομένων για να επιλύσει προβλήματα παλινδρόμησης και κατηγοριοποίησης [15]. Ο αλγόριθμος της λογιστικής παλινδρόμησης βασίζεται στο μοντέλο της γραμμικής παλινδρόμησης και εκφράζεται ως εξής:

$$P = a + b_G x_G + b_I x_I + \ldots + b_P x_P$$

***Εξίσωση 5**: Λογιστική Παλινδρόμηση*

Η λογιστική παλινδρόμηση μειώνει το εύρος πρόβλεψης και η τιμή που υπολογίζεται είναι διακριτή και κυμαίνεται μεταξύ 0 και 1, αντίθετα με τη γραμμική παλινδρόμηση όπου η τιμή είναι συνεχής. Για την ανάλυση της λογιστικής παλινδρόμησης χρησιμοποιείται ο λόγος πιθανοτήτων (odds). Έστω p είναι η πιθανότητα επιτυχίας εμφάνισης του γεγονότος και 1-p η πιθανότητα αποτυχίας εμφάνισης του γεγονότος, τότε ο λόγος πιθανοτήτων υπολογίζεται από τον τύπο:

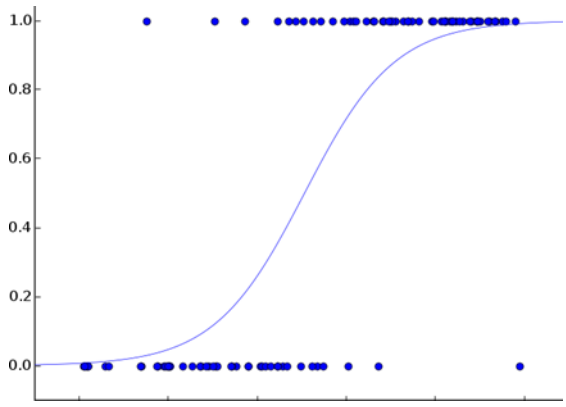
$$odds = \frac{p}{1-p}$$

***Εξίσωση 6**: Λόγος πιθανοτήτων λογιστικής Παλινδρόμησης*

Τέλος ορίζεται και η λειτουργία logit, η οποία είναι ο φυσικός λογάριθμος του λόγου πιθανότητας και μαθηματικά δίνεται από τη σχέση:

$$logit(p) = \ln\left(\frac{p}{1-p}\right)$$

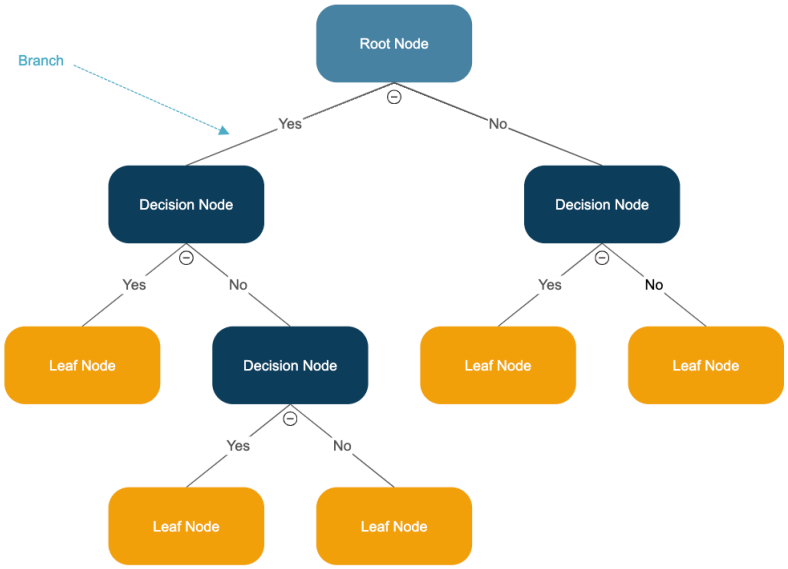
***Εξίσωση 7**: Φυσικός λογάριθμος του λόγου πιθανότητας*



***Εικόνα 9**: Λογιστική Παλινδρόμηση*

2.4.3 Δέντρα Αποφάσεων

Τα δέντρα αποφάσεων είναι μία δημοφιλής τεχνική μηχανικής μάθησης που χρησιμοποιείται για προβλήματα ταξινόμησης. Ένα δέντρο αποφάσεων αποτελείται από κόμβους απόφασης και κόμβους φύλλων. Οι κόμβοι απόφασης αντιστοιχούν σε ένα χαρακτηριστικό (feature) και οι κόμβοι φύλλων αντιπροσωπεύουν την τελική πρόβλεψη ή την κατηγορία. Κατά την εκπαίδευση ενός δέντρου αποφάσεων, ο αλγόριθμος επιλέγει τα χαρακτηριστικά και τα κατώφλια απόφασης που βελτιστοποιούν κάποιο κριτήριο όπως το Gini impurity, το information gain, ή το mean squared error όταν έχουμε να κάνουμε με προβλήματα παλινδρόμησης. Οι αποφάσεις λαμβάνονται κατακόρυφα και ιεραρχικά, ξεκινώντας από τον ριζικό κόμβο και καταλήγοντας σε έναν κόμβο φύλλου. Ένα από τα πλεονεκτήματα των δέντρων αποφάσεων είναι η ευκολία ερμηνείας των αποτελεσμάτων και η οπτικοποίηση των αποφάσεων. Το μειονέκτημα των δέντρων απόφασης είναι ότι ανάλογα με το βάθος του μπορεί να είναι ευάλωτα στην υπερεκπαίδευση (overfitting). Για να αντιμετωπιστεί αυτό το πρόβλημα χρησιμοποιούνται πρακτικές όπως το pruning (κλάδεμα) και η κανονικοποίηση για να περιορίσουν το βάθος του δέντρου και τον αριθμό των κόμβων του. Ευρέως γνωστοί αλγόριθμοι δέντρων απόφασης είναι οι ID3, CHAID, J48 και άλλοι.



***Εικόνα 10**: Δέντρο Απόφασης*

2.4.3.1 ID3

Ο αλγόριθμος ID3 (Iterative Dichotomiser 3) αναπτύχθηκε από τον Ross Quinlan το 1986 και είναι από τους πιο δημοφιλείς αλγορίθμους για τη δημιουργία δέντρων αποφάσεων [16]. Ο αλγόριθμος χρησιμοποιεί πληροφορίες από τα δεδομένα εκπαίδευσης για την δημιουργία του δέντρου. Δημοφιλή

κριτήρια αξιολόγησης για την επιλογή των καλύτερων χαρακτηριστικών που θα διαχωρίσουν τα δεδομένα σε υποσύνολα είναι μερικά από τα εξής:

1. **Εντροπία**: Η εντροπία είναι μια μετρική που αντιπροσωπεύει την αβεβαιότητα ή την ασάφεια σε ένα σύνολο δεδομένων. Υψηλή εντροπία σημαίνει ότι τα δεδομένα είναι πολύ ανόμοια, ενώ χαμηλή εντροπία σημαίνει ότι τα δεδομένα είναι πιο ομοιογενή. Υπολογίζεται από τον τύπο:

$$\mathrm{E(S)} = -\sum_{i=1}^{n} p_i \, log_2 p_i$$

***Εξίσωση 8**: Εντροπία*

Ενώ σε περίπτωση που έχουμε δυαδική κατηγοριοποίηση, αυτή υπολογίζεται από τον τύπο:

$$E(p) = -plog_2 p - (1-p)log_2(1-p)$$

***Εξίσωση 9**: Εντροπία δυαδικής κατηγοριοποίησης*

2. **Δείκτης Gini**: Ο δείκτης Gini είναι μια μετρική ανισότητας που χρησιμοποιείται επίσης στους αλγορίθμους δέντρων αποφάσεων για την επιλογή του κατάλληλου κόμβου διαχωρισμού. Υπολογίζει την πιθανότητα μιας τυχαίας παρατήρησης να κατατάσσεται λανθασμένα. Χαμηλός δείκτης Gini υποδεικνύει ομοιογενή δεδομένα, ενώ υψηλός δείκτης Gini υποδεικνύει ανόμοια δεδομένα. Μαθηματικά υπολογίζεται από τον τύπο:

$$Gini(S) = 1 - \sum_{i=1}^{n} p_i^2$$

***Εξίσωση 10**: Δείκτης Gini*

3. **Information Gain**: Το Information Gain είναι μια μετρική που υπολογίζει τη μείωση της αβεβαιότητας μετά το διαχωρισμό ενός συνόλου δεδομένων, χρησιμοποιώντας συνήθως εντροπία. Κατά τη διαδικασία κατασκευής του δέντρου, επιλέγεται ως κριτήριο διαχωρισμού το χαρακτηριστικό που προσφέρει το μεγαλύτερο Information Gain, δηλαδή τη μεγαλύτερη μείωση της αβεβαιότητας. Με αυτόν τον τρόπο, το δέντρο αποφάσεων μπορεί να βελτιώσει την ικανότητά του να προβλέπει σωστά τις ετικέτες των δεδομένων, διατηρώντας ταυτόχρονα την απλότητά του. Υπολογίζεται από τον τύπο:

$$IG(S) = E(S) - E(S, A)$$

***Εξίσωση 11**: Κέρδος πληροφορίας*

Ο αλγόριθμος επαναλαμβάνεται αναδρομικά στα υποσύνολα, δημιουργώντας έτσι ένα δέντρο αποφάσεων με κόμβους απόφασης και φύλλα. Τα φύλλα του δέντρου αποφάσεων αντιπροσωπεύουν τις κατηγορίες του προβλήματος ταξινόμησης, ενώ οι κόμβοι απόφασης αντιπροσωπεύουν τα χαρακτηριστικά των δεδομένων. Κατά την ταξινόμηση ενός νέου δείγματος, ξεκινάμε από τη ρίζα του δέντρου αποφάσεων και ακολουθούμε το κατάλληλο μονοπάτι, βασισμένο στις τιμές των

χαρακτηριστικών του δείγματος, μέχρι να φτάσουμε σε ένα φύλλο. Η κατηγορία που αντιπροσωπεύει το φύλλο είναι η προβλεπόμενη κατηγορία για το δείγμα.

2.4.3.2 C4.5

Ο αλγόριθμος C4.5 είναι ένας αλγόριθμος κατασκευής δέντρων αποφάσεων που αναπτύχθηκε από τον Ross Quinlan το 1993 ως εξέλιξη του αλγορίθμου ID3 [17]. Ο C4.5 χρησιμοποιεί μια ευριστική προσέγγιση βασισμένη στο πληροφοριακό κέρδος για την επιλογή των χαρακτηριστικών κατά την κατασκευή του δέντρου. Επιπλέον, ο C4.5 μπορεί να χειριστεί απουσιάζουσες τιμές, συνεχείς και διακριτές μεταβλητές, καθώς και περιττά χαρακτηριστικά. Ο αλγόριθμος εφαρμόζει επίσης περικοπή δέντρων για την αποφυγή υπερεκπαίδευσης και βελτιστοποίησης της γενίκευσης.

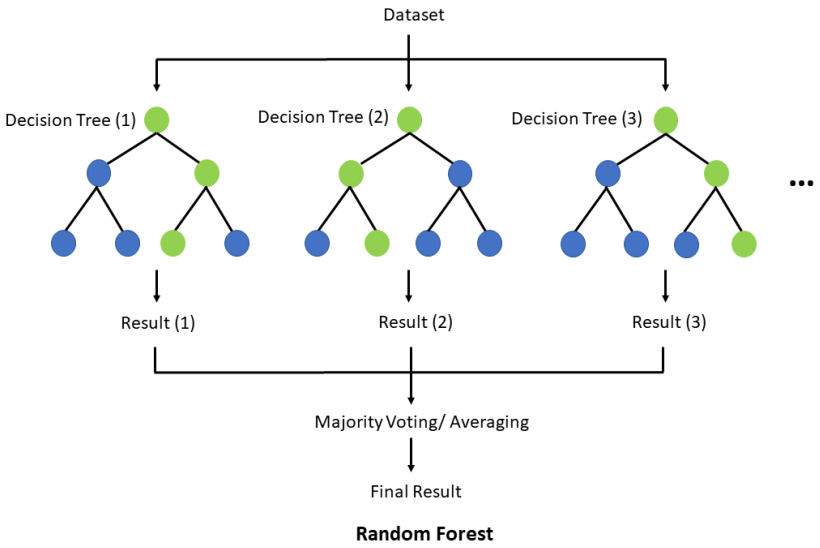
2.4.3.3 J48

Ο ταξινομητής J48 είναι ένας αλγόριθμος κατασκευής δέντρων αποφάσεων, που χρησιμοποιείται για προβλήματα ταξινόμησης. Ο αλγόριθμος J48 κατασκευάζει ένα δέντρο όπου κάθε κόμβος αντιστοιχεί σε ένα χαρακτηριστικό και κάθε κλάδος αντιστοιχεί σε μια τιμή αυτού του χαρακτηριστικού. Η κατασκευή του δέντρου γίνεται με την αναδρομική επιλογή του βέλτιστου χαρακτηριστικού για κάθε κόμβο, με σκοπό την αποτελεσματική διαίρεση του συνόλου δεδομένων σε υποσύνολα με υψηλή ομοιογένεια. Ο J48 χαρακτηρίζεται από την ικανότητά του να αντιμετωπίζει αριθμητικά και κατηγορικά χαρακτηριστικά, να αντιμετωπίζει απουσιάζουσες τιμές, καθώς και να αντιμετωπίζει προβλήματα με πολλαπλές κατηγορίες. Πρόκειται για έναν απλό και αποδοτικό αλγόριθμο, που προσφέρει ευκολία στην ερμηνεία των αποτελεσμάτων και είναι ανθεκτικός στον θόρυβο στα δεδομένα εκπαίδευσης.

2.4.4 Random Forest

Ο αλγόριθμος τυχαίων δασών (Random Forest) είναι μια τεχνική που επινοήθηκε από τον Leo Breiman το 2001 [18]. Η τεχνική αυτή δημιουργεί πολλά δέντρα αποφάσεων και συνδυάζει τις προβλέψεις τους για να παράγει ένα ισχυρότερο, πιο ακριβές μοντέλο. Κάθε δέντρο εκπαιδεύεται ανεξάρτητα, χρησιμοποιώντας ένα τυχαίο υποσύνολο δεδομένων (bootstrap sampling) και τυχαία επιλεγμένα χαρακτηριστικά σε κάθε κόμβο. Αυτή η τυχαιοποίηση μειώνει τη διακύμανση του μοντέλου και προσφέρει προστασία από την υπερεκπαίδευση. Το τελικό μοντέλο προκύπτει από την πλειοψηφία των ψήφων των δέντρων για προβλήματα ταξινόμησης ή το μέσο όρο των προβλέψεων για προβλήματα παλινδρόμησης. Τα τυχαία δάση αποτελούν μια δημοφιλή μέθοδο λόγω της

απλότητας, της αποδοτικότητας και της ικανότητας τους να αντιμετωπίζουν μεγάλο αριθμό χαρακτηριστικών και δειγμάτων.



Εικόνα 11: Random Forest

2.4.5 AdaBoost

Ο αλγόριθμος AdaBoost (Adaptive Boosting) είναι μια τεχνική ensemble learning που λειτουργεί συνδυάζοντας πολλούς απλούς ταξινομητές (ή "αδύναμους" ταξινομητές) για να δημιουργήσει έναν πιο ισχυρό ταξινομητή [19]. Οι "αδύναμοι" ταξινομητές είναι μοντέλα που έχουν ελαφρώς καλύτερη απόδοση από το να ταξινομούν τυχαία. Ο αλγόριθμος AdaBoost λειτουργεί ως εξής:

1. Αρχικά, όλα τα δείγματα εκπαίδευσης έχουν ίσο βάρος.
2. Ένας αδύναμος ταξινομητής εκπαιδεύεται στα δεδομένα.
3. Υπολογίζονται τα λάθη του ταξινομητή και το ακριβές βάρος του ταξινομητή, βάσει της ακρίβειάς του.
4. Τα βάρη των δειγμάτων ενημερώνονται, αυξάνοντας τα βάρη των λανθασμένα ταξινομημένων παραδειγμάτων και μειώνοντας τα βάρη των σωστά ταξινομημένων παραδειγμάτων.
5. Ο επόμενος αδύναμος ταξινομητής εκπαιδεύεται στα ενημερωμένα βάρη των δειγμάτων, με σκοπό να εστιάσει στα πιο δύσκολα ταξινομημένα παραδείγματα.
6. Επαναλαμβάνονται τα βήματα 3-5 για έναν προκαθορισμένο αριθμό εποχών ή μέχρις ότου η απόδοση του ensemble δεν βελτιώνεται άλλο.
7. Τέλος, τα αποτελέσματα των αδύναμων ταξινομητών συνδυάζονται με βάση τα βάρη τους, για να προκύψει ο τελικός, ισχυρός ταξινομητής.

Ο αλγόριθμος AdaBoost είναι αποτελεσματικός για πολλά προβλήματα ταξινόμησης και μπορεί να μειώσει σημαντικά το λάθος ταξινόμησης σε σύγκριση με τη χρήση ενός μόνο αδύναμου ταξινομητή. Επίσης, ο AdaBoost είναι εύκολος στην υλοποίηση και η μόνη παράμετρος που χρειάζεται να προσδιορίσουμε είναι το πλήθος των ταξηνομητών. Ωστόσο, ο αλγόριθμος είναι ευαίσθητος σε θορύβους.

2.4.6 Naïve Bayes

Ο απλοϊκός Bayes (Naïve Bayes) είναι ένας αλγόριθμος κατηγοριοποίησης βασισμένος στο θεώρημα του Bayes. Ο αλγόριθμος χρησιμοποιεί πιθανότητες για να προβλέψει την κατηγορία ενός δείγματος, λαμβάνοντας υπόψη τα χαρακτηριστικά του. Η κύρια ιδέα του απλοϊκού Bayes είναι η υπόθεση ανεξαρτησίας μεταξύ των χαρακτηριστικών, δηλαδή, ότι το επίπεδο επιρροής του κάθε χαρακτηριστικού στην κατηγορία είναι ανεξάρτητο από την παρουσία των υπολοίπων χαρακτηριστικών. Αυτή η υπόθεση καθιστά τον αλγόριθμο "απλοϊκό", καθώς στην πραγματικότητα τα χαρακτηριστικά μπορεί να έχουν κάποια εξάρτηση μεταξύ τους. Έστω λοιπόν ότι έχουμε ένα σύνολο δεδομένων D, το οποίο αντιπροσωπεύεται από ένα διάνυσμα $X=(x1, x2, ..., xn)$ με n διαστάσεις, και θέλουμε να ταξινομήσουμε σε m διαφορετικές κατηγορίες $C1, C2, ..., Cm$ που ανήκουν στην ίδια κλάση C. Για να ταξινομήσουμε ένα δείγμα X σε μια κατηγορία Ci, υπολογίζουμε την πιθανότητα P(Ci | X), δηλαδή την πιθανότητα ότι το δείγμα ανήκει στην κατηγορία Ci, δεδομένου του διανύσματος X. Η υπολογιστική διαδικασία για την εύρεση των πιθανοτήτων P(Ci | X) βασίζεται στο θεώρημα Bayes. Σύμφωνα με αυτό το θεώρημα, η πιθανότητα P(Ci | X) μπορεί να υπολογιστεί ως το γινόμενο των πιθανοτήτων P(X | Ci) και P(Ci) διαιρούμενο με την πιθανότητα P(X), δηλαδή:

$$P(C_i|X) = \frac{P(X|C_i) \cdot P(C_i)}{P(X)}$$

***Εξίσωση 12**: Θεώρημα Baynes*

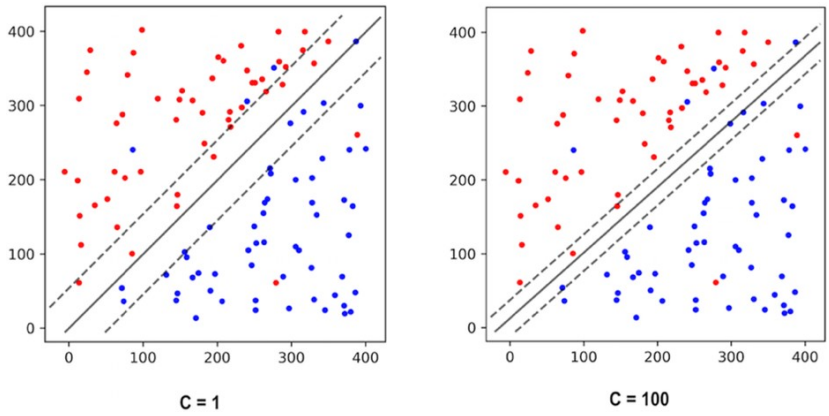
Παρά την απλοϊκή υπόθεση, ο αλγόριθμος Naïve Bayes είναι πολύ αποτελεσματικός σε πολλές περιπτώσεις και ειδικά όταν έχουμε μεγάλο όγκο δεδομένων. Η αποδοτικότητα του αλγορίθμου Naïve Bayes μπορεί μειωθεί από την ύπαρξη στενά συσχετισμένων χαρακτηριστικών, καθώς η υπόθεση ανεξαρτησίας μπορεί να μην είναι πάντα ισχύουσα. Ωστόσο, σε περιπτώσεις όπου η ανεξαρτησία των χαρακτηριστικών είναι μια προσεγγιστικά ορθή υπόθεση, ο αλγόριθμος μπορεί να παρέχει καλές προβλέψεις. Επιπλέον, ο αλγόριθμος Naïve Bayes είναι πολύ ανθεκτικός στον θόρυβο και τις απουσιάζουσες τιμές, καθώς μπορεί να χειριστεί απουσίες τιμών μέσω της πιθανοτικής προσέγγισης. Ακόμη, είναι εύκολος στην υλοποίηση και απαιτεί μικρό χρόνο εκπαίδευσης, καθιστώντας τον ιδιαίτερα κατάλληλο για προβλήματα με πολύ μεγάλο όγκο δεδομένων.

2.4.7 Μηχανές Διανυσμάτων Υποστήριξης (SVMs)

Οι μηχανές διανυσμάτων υποστήριξης (Support Vector Machines) είναι μια μέθοδος μηχανικής μάθησης που επινοήθηκε από τους Hava Siegelmonn και Vladimir Vapnik [20] και χρησιμοποιείται για την κατηγοριοποίηση και πρόβλεψη των κλάσεων ενός συνόλου δεδομένων. Η μέθοδος αποσκοπεί στον εντοπισμό ενός ορίου απόφασης μεταξύ των κλάσεων, που βρίσκεται στη μέγιστη δυνατή απόσταση από οποιοδήποτε σημείο των δεδομένων εκπαίδευσης. Για την επίτευξη του στόχου αυτού, η μέθοδος αναπτύσσει μια συνάρτηση f(x, α) που εξαρτάται από ένα σύνολο παραμέτρων α, βασισμένο σε σημεία που ονομάζονται διανύσματα υποστήριξης και αποτελούνται από το υποσύνολο των δεδομένων όπου ορίζεται η θέση του διαχωριστή. Στόχος των μηχανών διανυσμάτων υποστήριξης είναι η ελαχιστοποίηση του ανώτερου ορίου σφάλματος γενίκευσης, γεγονός που καθιστά το όριο απόφασης της μηχανικής εκπαίδευσης να πρέπει να έχει τη μέγιστη ελάχιστη απόσταση από το πιο κοντινό σημείο εκπαίδευσης. Το αναμενόμενο σφάλμα ελέγχου ορίζεται από τον τύπο:

$$R(\alpha) = \left(\frac{G}{I}\int |y - f(z,a)dP(x,y)|\right)$$

***Εξίσωση 13**: Αναμενόμενο σφάλμα ελέγχου*



***Εικόνα 12**: Μηχανές διανυσμάτων υποστήριξης.*

2.4.8 Νευρωνικά Δίκτυα

Τα Νευρωνικά Δίκτυα (Neural Networks) είναι μια κατηγορία αλγορίθμων μηχανικής μάθησης εμπνευσμένη από τον τρόπο λειτουργίας του ανθρώπινου εγκεφάλου. Στοχεύουν στην εκμάθηση και την αναπαράσταση συνθετών συσχετίσεων και μοτίβων από τα δεδομένα εισόδου. Ένα νευρωνικό δίκτυο αποτελείται από διασυνδεδεμένους νευρώνες που οργανώνονται σε τρία κύρια επίπεδα: το επίπεδο εισόδου (Input Layer), το κρυφό επίπεδο (Hidden Layer) και το επίπεδο εξόδου (Output Layer). Κάθε νευρώνας δέχεται είσοδο από τους νευρώνες του προηγούμενου επιπέδου, την

επεξεργάζεται με βάση μια συνάρτηση ενεργοποίησης και μεταδίδει την έξοδό του στους νευρώνες του επόμενου επιπέδου [21]. Η συνάρτηση ενεργοποίησης δίνεται από τον τύπο:

$$a_{} = g(in_{}) = g(\sum_{\_FM}^{H} w_{} \_a\_)$$

***Εξίσωση 14**: Συνάρτηση Ενεργοποίησης*

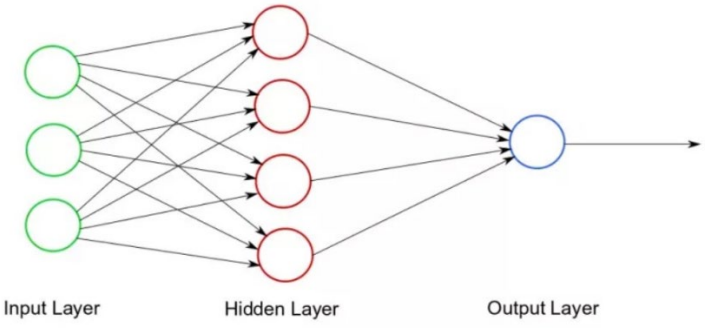
Μερικά από τα πιο δημοφιλή τεχνητά Νευρωνικά Δίκτυα που κατασκευάστηκαν ανά τα χρόνια μαζί με διάφορες πληροφορίες σχετικά με τον κατασκευαστή τους, τη χρονολογία δημιουργίας και τον τρόπο εκπαίδευσης δίνονται στον παρακάτω πίνακα:

***Πίνακας 1**:Διάφορες πληροφορίες σχετικά με τα γνωστότερα νευρωνικά δίκτυα [22].*

ΟΝΟΜΑ	ΚΑΤΑΣΚΕΥΑΣΤΗΣ	ΕΤΟΣ	ΤΡΟΠΟΣ ΕΚΠΑΙΔΕΥΣΗΣ
Perceptron	Rosenblatt	1957 - 1962	Με επίβλεψη
Adaline / Madaline	Widrow	1960 -1962	Με επίβλεψη
Back - propagation	Werbow, Rumelhart etal	1974 - 1986	Με επίβλεψη
Self-organizing map	Kohonen	1981	Χωρίς επίβλεψη
Hopfield net	Hopfield	1982	Με επίβλεψη
Boltzmann machine	Hinton, Hopkins, Szu	1985 - 1986	Με επίβλεψη

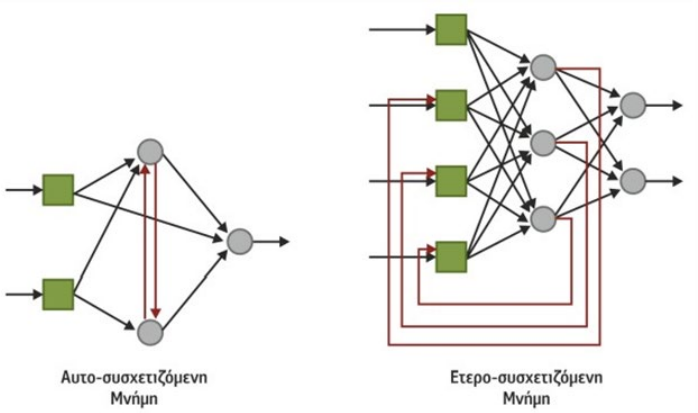
2.4.8.1 Αρχιτεκτονικές Νευρωνικών Δικτύων

Τα νευρωνικά δίκτυα μπορούν να διαχωριστούν σε δύο βασικές αρχιτεκτονικές: της πρόσθιας τροφοδότησης και της οπίσθιας τροφοδότησης. Στα νευρωνικά δίκτυα πρόσθιας τροφοδότησης, η έξοδος μιας μονάδας αποτελεί την είσοδο της επόμενης μονάδας στο επόμενο επίπεδο, δημιουργώντας μια σειρά από επίπεδα μονάδων.



***Εικόνα 13**: Τεχνητό νευρωνικό δίκτυο*

Στα νευρωνικά δίκτυα οπίσθιας τροφοδότησης, υπάρχουν δύο κατηγορίες: οι αυτοσυσχετιζόμενες μνήμες και οι ετεροσυσχετιζόμενες μνήμες. Στις αυτοσυσχετιζόμενες μνήμες, η ανατροφοδότηση γίνεται μόνο στους κόμβους του ίδιου επιπέδου, ενώ στις ετεροσυσχετιζόμενες μνήμες η ανατροφοδότηση γίνεται από ένα επίπεδο σε ένα άλλο επίπεδο του δικτύου [23].



***Εικόνα 14**: Νευρωνικά δίκτυα οπίσθιας τροφοδότης [23]*

2.4.8.2 Νευρωνικά Δίκτυα πολλών επιπέδων

Τα νευρωνικά δίκτυα πολλών επιπέδων ανήκουν στην κατηγορία της πρόσθιας τροφοδότησης. Σε αυτόν τον τύπο νευρωνικών δικτύων έχουμε ένα επίπεδο εισόδου που δέχεται τις εισόδους και στη συνέχεια αυτές περνάνε από ένα ή περισσότερα κρυφά επίπεδα προκειμένου να παραχθούν τα αποτελέσματα. Το τελευταίο επίπεδο (επίπεδο εξόδου) είναι αυτό που παράγει το τελικό αποτέλεσμα. Κάθε επίπεδο μπορεί να έχει πλήρως ή μερικώς συνδεδεμένους κόμβους, οι οποίοι συνδέονται με τους κόμβους του επόμενου επιπέδου για την μετάδοση των δεδομένων. Τα επίπεδα που βρίσκονται πιο κοντά στο επίπεδο εισόδου ονομάζονται κατώτερα επίπεδα, ενώ αυτά που βρίσκονται πιο κοντά στο επίπεδο εξόδου ονομάζονται ανώτερα επίπεδα [23].

2.5 Μέθοδοι αξιολόγησης μοντέλου

Για την αξιολόγηση του μοντέλου δεν μπορούμε να χρησιμοποιήσουμε τα δεδομένα του συνόλου δεδομένων καθώς το μοντέλο έχει τη δυνατότητα να θυμάται ολόκληρο το σύνολο εκπαίδευσης, επομένως θα κατηγοριοποιεί συνεχώς σωστά τα οποιαδήποτε δεδομένα από το σύνολο εκπαίδευσης. Για τον λόγο αυτό απαιτείται τα έχουμε ένα νέο σύνολο ελέγχου χωρίς την ετικέτα κατηγοριοποίησης. Έτσι το σύνολο δεδομένων είναι αναγκαίο να διαχωριστεί σε σύνολο εκπαίδευσης, που θα αφορά την εκπαίδευση του μοντέλου και σε σύνολο ελέγχου όπου θα χρησιμοποιηθεί για να εκτιμηθεί το ποσοστό αποτελεσματικότητας του εκάστοτε μοντέλου [24].

Σε αυτό το σημείο ας εισάγουμε την έννοια του πίνακα σύγχυσης (Confusion Matrix), που χρησιμοποιείται για την αξιολόγηση της απόδοσης ενός μοντέλου ταξινόμησης σε προβλήματα με δύο ή περισσότερες κατηγορίες. Ο πίνακας σύγχυσης παρουσιάζει τις προβλέψεις του μοντέλου σε σχέση με τις πραγματικές κατηγορίες των δεδομένων. Σε ένα δυαδικό πρόβλημα ταξινόμησης, ο πίνακας σύγχυσης είναι ένας πίνακας 2x2, που περιλαμβάνει τέσσερα κελιά:

1. TP (True Positive): Οι περιπτώσεις που το μοντέλο προέβλεψε ως θετικές και ήταν πράγματι θετικές.
2. FP (False Positive): Οι περιπτώσεις που το μοντέλο προέβλεψε ως θετικές, αλλά ήταν αρνητικές.
3. TN (True Negative): Οι περιπτώσεις που το μοντέλο προέβλεψε ως αρνητικές και ήταν πράγματι αρνητικές.
4. FN (False Negative): Οι περιπτώσεις που το μοντέλο προέβλεψε ως αρνητικές, αλλά ήταν θετικές.

		Predicted class	
		P	*N*
Actual Class	*P*	True Positives (TP)	False Negatives (FN)
	N	False Positives (FP)	True Negatives (TN)

***Εικόνα 15**: Πίνακας Σύγχυσης*

Με βάση τον πίνακα σύγχυσης, μπορούμε να υπολογίσουμε διάφορες μετρικές αξιολόγησης, όπως η ορθότητα (accuracy), η ακρίβεια (precision), η ανάκληση (recall) και ο αρμονικός μέσος (F-measure).

2.5.1 Ορθότητα

Πρόκειται για το σημαντικότερο κριτήριο, καθώς αξιολογεί το ποσοστό ακρίβειας του κάθε κατηγοριοποιητή. Η ορθότητα είναι το ποσοστό των σωστά ταξινομημένων παραδειγμάτων προς το συνολικό αριθμό των παραδειγμάτων [11]. Μαθηματικά αυτό εκφράζεται από τη σχέση:

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

***Εξίσωση 15**: Εξίσωση Ορθότητας*

Όπως βλέπουμε η ορθότητα μπορεί να οριστεί ως ο λόγος των σωστών προβλέψεων (θετικών και αρνητικών) προς το συνολικό αριθμό όλων των δειγμάτων.

2.5.2 Ακρίβεια

Η ακρίβεια (precision) είναι μια μετρική αξιολόγησης που χρησιμοποιείται για να καθορίσει το ποσοστό των σωστά ταξινομημένων θετικών περιπτώσεων ανάμεσα σε όλες τις περιπτώσεις που το μοντέλο πρόβλεψε ως θετικές. Μαθηματικά εκφράζεται από τη σχέση:

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

***Εξίσωση 16**: Εξίσωση Ακρίβειας*

2.5.3 Ανάκληση

Η ανάκληση (recall) καθορίζει το ποσοστό των σωστά ταξινομημένων θετικών περιπτώσεων ανάμεσα σε όλες τις πραγματικές θετικές περιπτώσεις [11]. Υπολογίζεται από τη σχέση:

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

***Εξίσωση 17**: Εξίσωση Ανάκλησης*

Η ανάκληση είναι μια σημαντική μετρική σε προβλήματα όπου το κόστος των λανθασμένων αρνητικών προβλέψεων είναι υψηλό.

2.5.4 Αρμονικός Μέσος

Η μετρική f1-score είναι ο αρμονικός μέσος της ακρίβειας (Precision) και της ανάκλησης (Recall). Μαθηματικά υπολογίζεται από τη σχέση:

$$F1\ score = 2 \cdot \frac{Precison \cdot Recall}{Precision + Recall}$$

***Εξίσωση 18**: Εξίσωση Αρμονικού Μέσου*

Το f1-score είναι μια μετρική που προσπαθεί να συνδυάσει την ακρίβεια και την ανάκληση σε έναν μοναδικό αριθμό. Είναι ιδιαίτερα χρήσιμο σε περιπτώσεις όπου έχουμε ανισορροπία κλάσεων, δηλαδή όταν έχουμε πολύ λίγα θετικά δείγματα σε σχέση με τα αρνητικά.

2.6 Αποτελέσματα ερευνών συγκριτικής αξιολόγησης

Στη συνέχεια παρουσιάζονται δυο μελέτες που αξιολογούν συγκριτικά διάφορους αλγορίθμους μηχανικής μάθησης. Η πρώτη έρευνα [25] ασχολείται με τη σύγκριση αλγορίθμων για την ταξινόμηση χαρακτηριστικών φοιτητών, χρησιμοποιώντας ένα πλαίσιο μεθοδολογίας, ενώ η δεύτερη [27] πραγματεύεται την μελέτη αλγορίθμων μηχανικής μάθησης για πρόβλεψη καιρού με αναγνώριση προτύπων από ιστορικά στοιχεία, αντικείμενο με το οποίο ασχολείται και η παρούσα διπλωματική εργασία.

2.6.1 Συγκριτική ανάλυση αλγορίθμων για την ταξινόμηση χαρακτηριστικών μαθητών χρησιμοποιώντας μια μεθοδολογική πλατφόρμα.

Σύμφωνα με την έρευνα [25] το πεδίο της εκπαίδευσης προσφέρει ένα εύφορο έδαφος για εφαρμογές εξόρυξης δεδομένων. Η μοντελοποίηση της απόδοσης των μαθητών μπορεί να αποτελέσει ένα εξαιρετικό εργαλείο τόσο για τους εκπαιδευτικούς όσο και για τους μαθητές, προκειμένου να γίνουν σωστές προσαρμογές στο πρόγραμμα σπουδών και στις μεθόδους διδασκαλίας. Ως βάση δοκιμής για αυτήν τη συγκριτική μελέτη αξιολόγησης, επιλέχθηκαν τέσσερις αλγόριθμοι ταξινόμησης. Συγκεκριμένα επιλέχθηκαν: ο αλγόριθμος k-means, ο αλγόριθμος k-κοντινότερων γειτόνων, οι Μηχανές Διανυσματικής Στήριξης και ο απλοϊκός Bayes. Ο στόχος αυτής της μελέτης είναι να προσδιορίσει τυχόν διαφορές και ομοιότητες που μπορεί να έχουν αυτοί οι αλγόριθμοι στην εξόρυξη δεδομένων των χαρακτηριστικών των μαθητών, το σύνολο των οποίων (δεδομένων των μαθητών) αντλήθηκε από το UCI Machine Learning. Τα χαρακτηριστικά των δεδομένων περιλαμβάνουν βαθμούς μαθητών, δημογραφικά, κοινωνικά και σχολικά χαρακτηριστικά και συλλέχθηκαν μέσω ερωτηματολογίων. Συνολικά περιλαμβάνονται δύο σύνολα δεδομένων σχετικά με την απόδοση σε δύο διακριτά μαθήματα, στα μαθηματικά και στη γλωσσική εκμάθηση. Ο αριθμός των δειγμάτων στο σύνολο δεδομένων είναι 649 και ο αριθμός των γνωρισμάτων είναι 33 και περιλαμβάνουν μεταξύ άλλων το φύλο, την ηλικία, τον αριθμό απουσιών, το επάγγελμα γονέων, συνολικό χρόνο διαβάσματος, ελεύθερο χρόνο και την κατανάλωση αλκοόλ. Ακόμη οι συγγραφείς θεωρούν ως G1: το βαθμό επίδοσης του μαθητή κατά την πρώτη περίοδο και αντίστοιχα ως G2: το βαθμό επίδοσης του μαθητή κατά την δεύτερη περίοδο και ως G3: τον τελικό βαθμό επίδοσης.

Στόχος είναι η πρόβλεψη του τελικού βαθμό επίδοσης των μαθητών (G3) καθώς και η ανάδειξη των βασικών μεταβλητών που επηρεάζουν την εκπαιδευτική διαδικασία. Οι δύο βασικές κατηγορίες (δηλαδή Μαθηματικά και Μάθηση Γλώσσας) θα προσεγγιστούν με τρεις προσεγγίσεις:

- Με δυαδική ταξινόμηση (επιτυχία αν: G3 ≥ 10, αλλιώς αποτυχία)
- Με ταξινόμηση πέντε επίπεδων (από I πολύ καλό ή άριστο έως V - ανεπαρκές)
- Με παλινδρόμηση έχοντας αριθμητική έξοδο που κυμαίνεται από το μηδέν (0%) έως το είκοσι (100%),για την τιμή G3 (αριθμητική έξοδος από 0 έως 20).

Παρακάτω παρατίθενται τα αποτέλεσμα της ερευνάς για την προσέγγιση της δυαδικής ταξινόμησης στα μαθηματικά και την εκμάθηση γλώσσας:

***Πίνακας 2**: Αποτελέσματα δυαδικής ταξινόμησης στα Μαθηματικά*

Αλγόριθμος	Ορθότητα
K-Means	81.57 %
k-NN	78.64 %
SVM	79.14 %
Naïve Bayes	80.94 %

***Πίνακας 3**: Αποτελέσματα δυαδικής ταξινόμησης στη Γλωσσική Εκμάθηση*

Αλγόριθμος	Ορθότητα
K-Means	81.24 %
k-NN	87.24 %
SVM	88.07 %
Naïve Bayes	87.27 %

Οι συγγραφείς με βάση τα ευρήματα των παραπάνω πινάκων συμπεραίνουν ότι ο αλγόριθμος K-means αποδίδει καλύτερα τόσο για τα μαθηματικά όσο και για την γλωσσική εκμάθηση.

Καθώς διαισθητικά ο βαθμός G1 της πρώτης περιόδου και ο βαθμός G2 της δεύτερης περιόδου θα είχαν μεγάλη επίδραση, οι συγγραφείς διεξήγαγαν την ταξινόμηση με πέντε επίπεδα (από I πολύ καλό ή άριστο έως B - ανεπαρκές), τα αποτελέσματα της οποίας φαίνεται στους παρακάτω πίνακες:

***Πίνακας 4**: Αποτελέσματα ταξινόμησης πέντε επιπέδων στα Μαθηματικά*

Αλγόριθμος	Ορθότητα

K-Means	52.87 %
k-NN	46.84 %
SVM	46.17 %
Naïve Bayes	57.3 %

Πίνακας 5:Αποτελέσματα ταξινόμησης πέντε επιπέδων στη Γλωσσική Εκμάθηση

Αλγόριθμος	**Ορθότητα**
K-Means	55.17 %
k-NN	50.27 %
SVM	50.37 %
Naïve Bayes	54.2 %

Τέλος προκειμένου να βρεθούν οι συσχετίσεις μεταξύ των επιλεγμένων ανεξάρτητων μεταβλητών και της εξαρτημένης μεταβλητής απόδοσης στο προηγούμενο στάδιο, οι συγγραφείς σχεδίασαν ένα μοντέλο παλινδρόμησης τα αποτελέσματα του οποίου φαίνονται στους παρακάτω πίνακες:

Πίνακας 6: Αποτελέσματα παλινδρόμησης στα Μαθηματικά

Αλγόριθμος	**Απόδοση**
K-Means	2.71
k-NN	3.1
SVM	3.12
Naïve Bayes	3.14

Πίνακας 7: Αποτελέσματα παλινδρόμησης στη Γλωσσική Εκμάθηση

Αλγόριθμος	**Απόδοση**
K-Means	1.89
k-NN	2.1
SVM	1.66
Naïve Bayes	2.15

Κλείνοντας, οι συγγραφείς επισημαίνουν πως μελέτη των παραγόμενων δεδομένων μπορεί να παρέχει χρήσιμες πληροφορίες. Οι πληροφορίες αυτές μπορούν να χρησιμοποιηθούν για το σχεδιασμό ενός

πιο ελκυστικού προγράμματος σπουδών, για την πρόβλεψη της επίδοσης των φοιτητών και της προόδου των φοιτητών. Δεδομένου ότι η φύση κάθε αλγορίθμου παραμένει διαφορετική και συνεπώς, η απόδοσή τους διαφέρει όσον αφορά τα δεδομένα, μπορεί να συμπεραθεί με ασφάλεια ότι κάθε αλγόριθμος έχει τη δική του δυναμική προς τον λόγο για τον οποίο χρησιμοποιείται.

2.6.2 Μελέτη Αλγορίθμων Πρόβλεψης Καιρικών Συνθηκών Σύμφωνα με Ιστορικά Στοιχεία και Ακολουθίες Προτύπων

Η μελέτη [27] υποθέτει ότι υπάρχει μια αναλογία καιρού των ημερών του τρέχοντος έτους με τον καιρό που έχει καταγραφεί στο παρελθόν στη πόλη Albany της Νέας Υόρκης, επιχειρώντας να αναγνωριστούν πιθανές συσχετίσεις ή πρότυπα καιρού που επαναλαμβάνονται. Η μελέτη εφαρμόζει αλγόριθμους μηχανικής μάθησης τους οποίους εκπαιδεύει σε διάφορα μοντέλα προτείνοντας το πιο αξιόπιστο με σκοπό την πρόβλεψη βροχόπτωσης ή μη βροχόπτωσης. Το σύνολο δεδομένων που χρησιμοποιήθηκε προέρχεται από τη διαδικτυακή εφαρμογή που προσφέρει ίδρυμα ερευνών του πολιτειακού πανεπιστημίου της Utah που προσφέρει δωρεάν πρόσβαση σε κλιματολογικά δεδομένα για διάφορες πόλεις σε ολόκληρο τον κόσμο και βρίσκεται στη διεύθυνση: https://climate.usu.edu/mapServer/mapGUI/index.php. Τα αποτελέσματα φαίνονται παρακάτω:

Πίνακας 8: Αποτελέσματα σύγκρισης ταξινομητών για πρόβλεψη καιρού

Κατηγοριοποιητές	Σωστά Ταξινομημένα	F-Measure Κλάση: None	F-Measure Κλάση: Rain
LMT	93.88%	0.967	0.606
Decision Table	93.88%	0.967	0.533
Simple Logistic	93.64%	0.966	0.540
ClassificationViaRegression	93.51%	0.965	0.566
Naïve Bayes	93.39%	0.965	0.426
J48	93.21%	0.963	0.554
Stacking LogitBoost-J48-IBK	93.08%	0.963	0.539
LogitBoost	92.84%	0.962	0.435
Bagging-REPTree	92.35%	0.958	0.545
REPTree	92.17%	0.957	0.549
PART	91.86%	0.955	0.552
AdaBoostM1-J48	89.96%	0.944	0.494

Random Committee	89.84%	0.944	0.468
IBk	88.80%	0.938	0.440
Random Tree	87.52%	0.930	0.396

Από τα αποτελέσματα της πρόβλεψης ο συγγραφέας διαπιστώνει πως τα καλύτερα αποτελέσματα πρόβλεψης πετυχαίνει το μοντέλο κατηγοριοποίησης *LMT* (LogisticModelTree), το οποίο συνδυάζει παλινδρόμηση (LogitBoost) και δένδρο αποφάσεων (C4.5). Η ακρίβεια πρόβλεψης με το συγκεκριμένο μοντέλο είναι 93.88% για τα 1634 στιγμιότυπα. Το μοντέλο είναι ικανό να προβλέψει τα 1457 από τα 1498 στιγμιότυπα που δεν είχαμε βροχή, δηλαδή το 97.26% και 77 από τα 136 στιγμιότυπα που είχαμε βροχή δηλαδή το 56.61%. Το μοντέλο που κατάφερε να βρει τα περισσότερα στιγμιότυπα με βροχή δημιουργήθηκε με τον κατηγοριοποιητή PART ο οποίος προέβλεψε 88 από τα 136, δηλαδή το 64,70%, όμως κατέταξε λανθασμένα περισσότερα στιγμιότυπα στην κλάση rain ενώ ανήκαν στην κλάση none σε σχέση με τον LMT [27].

Κεφάλαιο 3: Μεθοδολογία

Αυτό το κεφάλαιο θα εξηγήσει τη μεθοδολογία που ακολουθήθηκε στη διπλωματική εργασία. Αρχικά, θα παρουσιαστεί η γλώσσα προγραμματισμού Python, και το εργαλείο WEKA, περιγράφοντας τη γλώσσα προγραμματισμού και το περιβάλλον σε κάθε περίπτωση, καθώς και τις βιβλιοθήκες της Python που χρησιμοποιήθηκαν. Στη συνέχεια, παρουσιάζεται το σύνολο δεδομένων που χρησιμοποιήθηκε, με τα επιμέρους γνωρίσματα του. Επιπλέον, παρουσιάζονται οι αλγόριθμοι μηχανικής μάθησης που χρησιμοποιήθηκαν καθώς και οι παράμετροί τους. Τέλος, περιγράφονται τα βήματα υλοποίησης της διπλωματικής εργασίας.

3.1 Περιβάλλον υλοποίησης

Για την υλοποίηση των αλγορίθμων στην εν λόγω διπλωματική εργασία χρησιμοποιήθηκαν οι βιβλιοθήκες Pandas, Numpy, Matplotlib, Seaborn και Scikit-Learn της Python. Οι βιβλιοθήκες είναι ιδιαίτερα δημοφιλείς και χρησιμοποιούνται για την διαχείριση και την ανάλυση δεδομένων, την οπτικοποίηση των αποτελεσμάτων αλλά και για την εκπαίδευση και τον έλεγχο των αλγορίθμων μηχανικής μάθησης. Το περιβάλλον προγραμματισμού το οποίο χρησιμοποιήθηκε για την Python είναι το Google Colab. Το Google Colab είναι μια δωρεάν πλατφόρμα ανάπτυξης και εκτέλεσης κώδικα στο cloud, βασισμένη στο Jupyter Notebook. Παρέχει δυνατότητες όπως την εγκατάσταση και χρήση βιβλιοθηκών Python για ανάλυση δεδομένων και υλοποίηση αλγορίθμων μηχανικής μάθησης, προσβάσιμο από οποιαδήποτε συσκευή με πρόσβαση στο internet. Οι βιβλιοθήκες Python που χρησιμοποιήθηκαν είναι οι ακόλουθες:

- **Pandas**: Το Pandas είναι μια βιβλιοθήκη ανοιχτού κώδικα για τη γλώσσα προγραμματισμού Python που δημιουργήθηκε από τον Wes McKinney το 2008 [28] με σκοπό τη βελτίωση της αποδοτικότητας και της ευκολίας στην επεξεργασία δεδομένων. Το Pandas είναι ιδιαίτερα χρήσιμο για την επεξεργασία και ανάλυση ταξινομημένων και χρονοσειρικών δεδομένων, καθώς παρέχει εύχρηστες δομές δεδομένων όπως το DataFrame.
- **Matplotlib**: Το Matplotlib είναι μια δημοφιλής βιβλιοθήκη οπτικοποίησης δεδομένων της γλώσσας προγραμματισμού Python που δημιουργήθηκε το 2002 από τον John D. Hunter και αποτελεί ένα σημαντικό εργαλείο για την ανάλυση δεδομένων [30]. Το Matplotlib επιτρέπει τη δημιουργία διαγραμμάτων και γραφημάτων, ιστογραμμάτων και πολλών άλλων.
- **Seaborn**: Το Seaborn είναι μια βιβλιοθήκη οπτικοποίησης δεδομένων στη γλώσσα προγραμματισμού Python που δημιουργήθηκε από τον Michael Waskom το 2012 και βασίζεται στο Matplotlib [30]. Στοχεύει στη διευκόλυνση της δημιουργίας πιο ελκυστικών και

πληροφοριακά πλούσιων γραφημάτων. Το Seaborn προσφέρει υψηλού επιπέδου εντολές για την κατασκευή στατιστικών γραφημάτων, όπως κατανεμημένα διαγράμματα, ιστογράμματα, γραφήματα κατηγοριών και άλλα.

- **NumPy**: Το NumPy είναι μια βασική βιβλιοθήκη της Python για επιστημονικούς υπολογισμούς, που προσφέρει υποστήριξη για μεγάλους, πολυδιάστατους πίνακες και προχωρημένες μαθηματικές λειτουργίες [29].
- **Scikit-learn**: Το Scikit-learn είναι μια βιβλιοθήκη ανοιχτού κώδικα της γλώσσας προγραμματισμού Python, η οποία παρέχει εργαλεία για την επεξεργασία δεδομένων, την εκπαίδευση και την αξιολόγηση μοντέλων μηχανικής μάθησης. Δημιουργήθηκε το 2007 από τον David Cournapeau [31] και έχει εξελιχθεί σε ένα από τα πιο δημοφιλή και ευρέως χρησιμοποιούμενα πακέτα για τη μηχανική μάθηση. Η βιβλιοθήκη Scikit-learn περιλαμβάνει πολλές λειτουργίες, όπως ταξινόμηση, παλινδρόμηση, ομαδοποίηση, μείωση διαστάσεων, επιλογή χαρακτηριστικών και βελτιστοποίηση υπερπαραμέτρων. Παρέχει επίσης εύχρηστες διεπαφές για την προεπεξεργασία δεδομένων, τη διαχείριση απουσιάζουσας τιμών και την κανονικοποίηση.

Ακόμη η συγκριτική αξιολόγηση των αντίστοιχων αλγορίθμων πραγματοποιήθηκε και με το εργαλείο ανοιχτού κώδικα WEKA ώστε να πραγματοποιηθεί και μια συγκριτική αξιολόγηση μεταξύ της βιβλιοθήκης του scikit-learn και του WEKA. Το WEKA (Waikato Environment for Knowledge Analysis) είναι ένα λογισμικό ανοιχτού κώδικα για την εξόρυξη δεδομένων και τη μηχανική μάθηση, αναπτυγμένο από το Πανεπιστήμιο της Waikato στη Νέα Ζηλανδία. Το WEKA παρέχει μια συλλογή από αλγόριθμους μηχανικής μάθησης και εργαλεία για την προεπεξεργασία δεδομένων, την ταξινόμηση, την παλινδρόμηση, την ομαδοποίηση, τη μείωση διαστάσεων και την επιλογή χαρακτηριστικών. Το εργαλείο χαρακτηρίζεται από γραφική διεπαφή χρήστη που επιτρέπει εύκολη πρόσβαση στις λειτουργίες του, καθιστώντας το κατάλληλο για αρχάριους και έμπειρους χρήστες. Τέλος, οι αλγόριθμοί του είναι υλοποιημένοι στη γλώσσα προγραμματισμού Java, προσφέροντας υποστήριξη και επιτρέποντας την ενσωμάτωση και σε άλλες εφαρμογές Java.

3.2 Συλλογή, Προεπεξεργασία και Μετασχηματισμός Συνόλου Δεδομένων

Σε αυτή την ενότητα θα παρουσιάσουμε αναλυτικά τα τρία πρώτα στάδια της εφαρμογής της διαδικασίας ανακάλυψης γνώσης από βάσεις δεδομένων (KDD διαδικασία) που περιλαμβάνει τα βήματα της Συλλογής Δεδομένων, της Προεπεξεργασίας Δεδομένων, του Μετασχηματισμού δεδομένων όπως περιγράφεται στην ενότητα 2.3. Για λόγους σύγκρισης των κατηγοριοποιητών μεταξύ της πλατφόρμας του WEKA και της βιβλιοθήκης του scikit-learn τα στάδια της συλλογής της

προεπεξεργασίας και του μετασχηματισμού δεδομένων που θα πραγματοποιηθούν αποκλειστικά με Python και ύστερα θα μετατρέψουμε το παραγόμενο .csv σε .arff ώστε να το εισάγουμε στο WEKA. Με τον τρόπο αυτό οι κατηγοριοποιητές θα χρησιμοποιηθούν στο ίδιο επεξεργασμένο (και μετασχηματισμένο) σύνολο δεδομένων, πράγμα που κάνει τη σύγκριση μεταξύ του WEKA και του Scikit-learn πιο αντικειμενική.

3.2.1 Συλλογή Δεδομένων: Σύνολο Δεδομένων Australia Weather Data

Το σύνολο δεδομένων αποτελείται από καθημερινές μετεωρολογικές μετρήσεις 10 ετών από διάφορες πόλεις της Αυστραλίας κατά το έτος 2010. Υπενθυμίζουμε ότι το φαινόμενο La Niña κατά το έτος 2010 ήταν ιδιαίτερα καταστροφικό με εκτεταμένες πλημμύρες σε όλη τη χώρα [11]. Οι ημερήσιες προβλέψεις αντλήθηκαν από πολυάριθμους μετεωρολογικούς σταθμούς που είναι προσβάσιμες από τις διευθύνσεις: http://www.bom.gov.au/climate/data και http://www.bom.gov.au/climate/dwo/. Ολόκληρο το σύνολο δεδομένων μπορεί να βρεθεί στην διεύθυνση: https://www.kaggle.com/datasets/arunavakrchakraborty/australia-weather-data.

Αναλυτικότερα το σύνολο δεδομένων αποτελείται από τις εξής στήλες:

***Πίνακας 9**: Γνωρίσματα Συνόλου δεδομένων Australia Weather*

Χαρακτηριστικά	**Περιγραφή**
Location	Το όνομα της πόλης στην Αυστραλία.
MinTemp	Η ελάχιστη θερμοκρασία κατά τη διάρκεια μιας συγκεκριμένης ημέρας σε βαθμούς Κελσίου.
MaxTemp	Η μέγιστη θερμοκρασία κατά τη διάρκεια μιας συγκεκριμένης ημέρας σε βαθμούς Κελσίου.
Rainfall	Η βροχόπτωση κατά τη διάρκεια μιας συγκεκριμένης ημέρας σε χιλιοστά.
Evaporation	Η εξάτμιση κατά τη διάρκεια μιας συγκεκριμένης ημέρας σε χιλιοστά.
Sunshine	Ηλιοφάνεια κατά τη διάρκεια μιας συγκεκριμένης ημέρας σε ώρες.
WindGusDir	Η κατεύθυνση του ισχυρότερου ανέμου κατά τη διάρκεια μιας συγκεκριμένης ημέρας με βάση τους 16 κατευθυντήρες

	ανέμου.
WindGustSpeed	Η ταχύτητα του ισχυρότερου ανέμου κατά τη διάρκεια μιας συγκεκριμένης ημέρας σε χιλιόμετρα ανά ώρα.
WindDir9am	Η κατεύθυνση του ανέμου για 10 λεπτά πριν από τις 9 π.μ. με βάση τους κατευθυντήρες ανέμου.
WindDir3pm	Η κατεύθυνση του ανέμου για τα 10 λεπτά πριν από τις 3 μ.μ.
WindSpeed9am	Η ταχύτητα του ανέμου για τα 10 λεπτά πριν από τις 9 π.μ. (χιλιόμετρα ανά ώρα)
WindSpeed3pm	Η ταχύτητα του ανέμου για τα 10 λεπτά πριν από τις 3 μ.μ. (χιλιόμετρα ανά ώρα)
Humidity9am	Η υγρασία του αέρα στις 9 π.μ. (ποσοστό)
Humidity3pm	Η υγρασία του αέρα στις 3 μ.μ. (ποσοστό)
Pressure9am	Η ατμοσφαιρική πίεση στις 9 π.μ. (εκτοπασκάλ)
Pressure3pm	Η ατμοσφαιρική πίεση στις 3 μ.μ. (εκτοπασκάλ)
Cloud9am	Η απόκρυψη των συννεφιών στις 9 π.μ. (οκτάβες)
Cloud3pm	Η απόκρυψη των συννεφιών στις 3 μ.μ. (οκτάβες)
Temp9am	Η θερμοκρασία στις 9 π.μ. (βαθμοί Κελσίου)
Temp3pm	Η θερμοκρασία στις 3 μ.μ. (βαθμοί Κελσίου)
RainToday	'Yes' αν η σημερινή μέρα είναι βροχερή, διαφορετικά 'No'.
RainTomorrow	'1' αν η σημερινή μέρα είναι βροχερή, διαφορετικά '0'.

Ένα παράδειγμα μετεωρολογικών προβλέψεων για την πόλη της Καμπέρας βρίσκεται στον ακόλουθο σύνδεσμο: http://www.bom.gov.au/climate/dwo/IDCJDW2801.latest.shtml.

3.2.1.1 Ανάλυση Συνόλου Δεδομένων

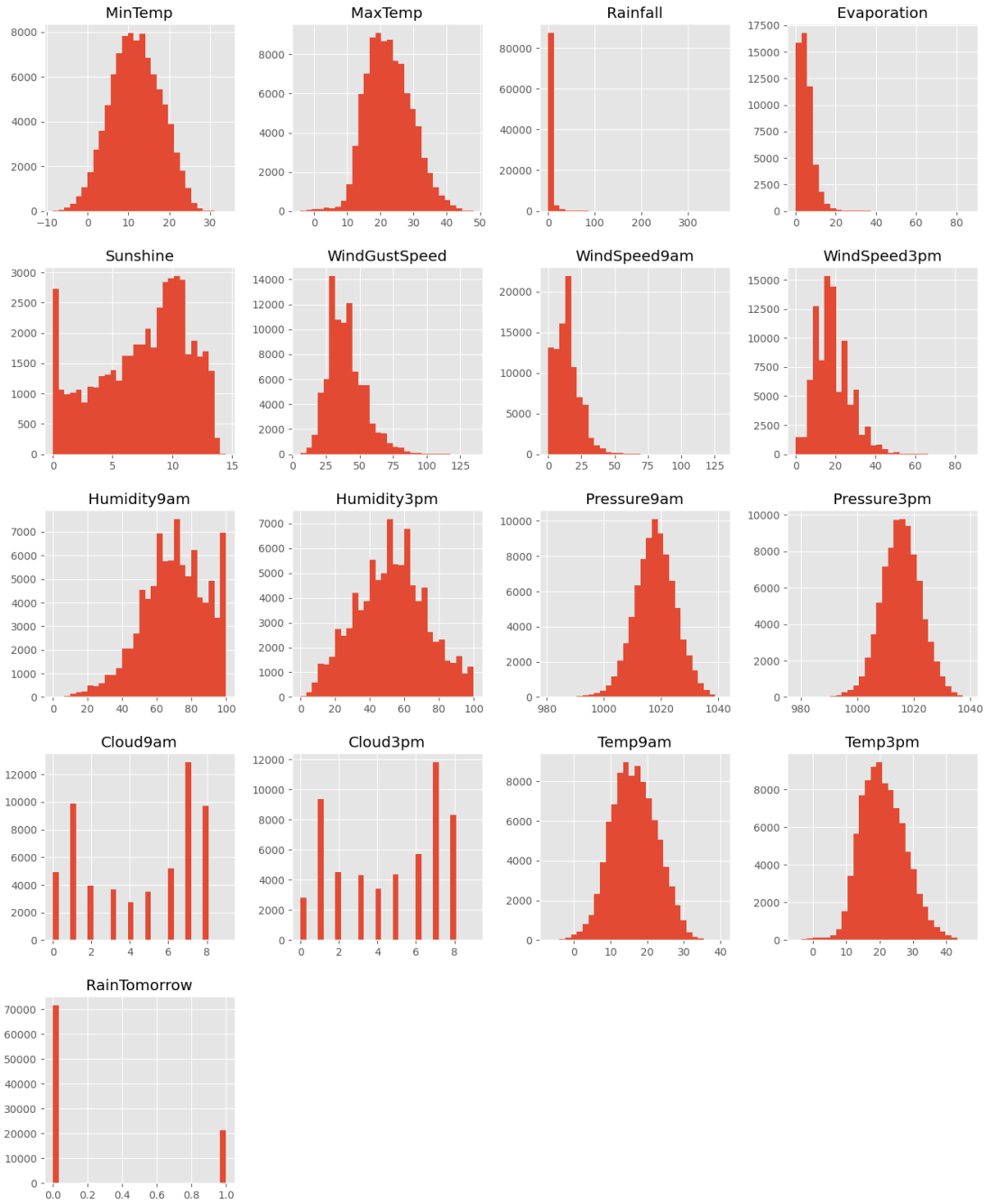
Στον ακόλουθο πίνακα παρατηρούνται μερικά στατιστικά του συνόλου δεδομένων **Australian Weather Data** με χρήση της συνάρτησης describe() της βιβλιοθήκης Pandas. Αυτά τα στατιστικά των 99516 συνολικών εγγραφών είναι το πλήθος κάθε γνωρίσματος, η μέση τιμή, η τυπική απόκλιση, η ελάχιστη τιμή και η μέγιστη τιμή.

Πίνακας 10: Στατιστικά Στοιχεία συνόλου δεδομένων

Χαρακτηριστικά	count	mean	std	min	max
MinTemp	99073.0	12.176266	6.390882	-8.5	33.9
MaxTemp	99286.0	23.218513	7.115072	-4.1	48.1
Rainfall	98537.0	2.353024	8.487866	0.0	371.0
Evaporation	56985.0	5.461320	4.162490	0.0	86.2
Sunshine	52199.0	7.615090	3.783008	0.0	14.5
WindGustSpeed	93036.0	39.976966	13.581524	6.0	135.0
WindSpeed9am	98581.0	14.004849	8.902323	0.0	130.0
WindSpeed3pm	97681.0	18.650464	8.801827	0.0	87.0
Humidity9am	98283.0	68.866376	19.074951	0.0	100.0
Humidity3pm	97010.0	51.433296	20.777616	0.0	100.0
Pressure9am	89768.0	1017.684638	7.110166	980.5	1041.0
Pressure3pm	89780.0	1015.286204	7.045189	978.2	1039.6
Cloud9am	61944.0	4.447985	2.886580	0.0	9.0
Cloud3pm	59514.0	4.519122	2.716618	0.0	9.0
Temp9am	98902.0	16.970041	6.488961	-7.0	40.2
Temp3pm	97612.0	21.681340	6.931681	-5.1	46.7
RainToday	99516.0	0.224677	0.417372	0.0	1.0

3.2.1.2 Ιστογράμματα

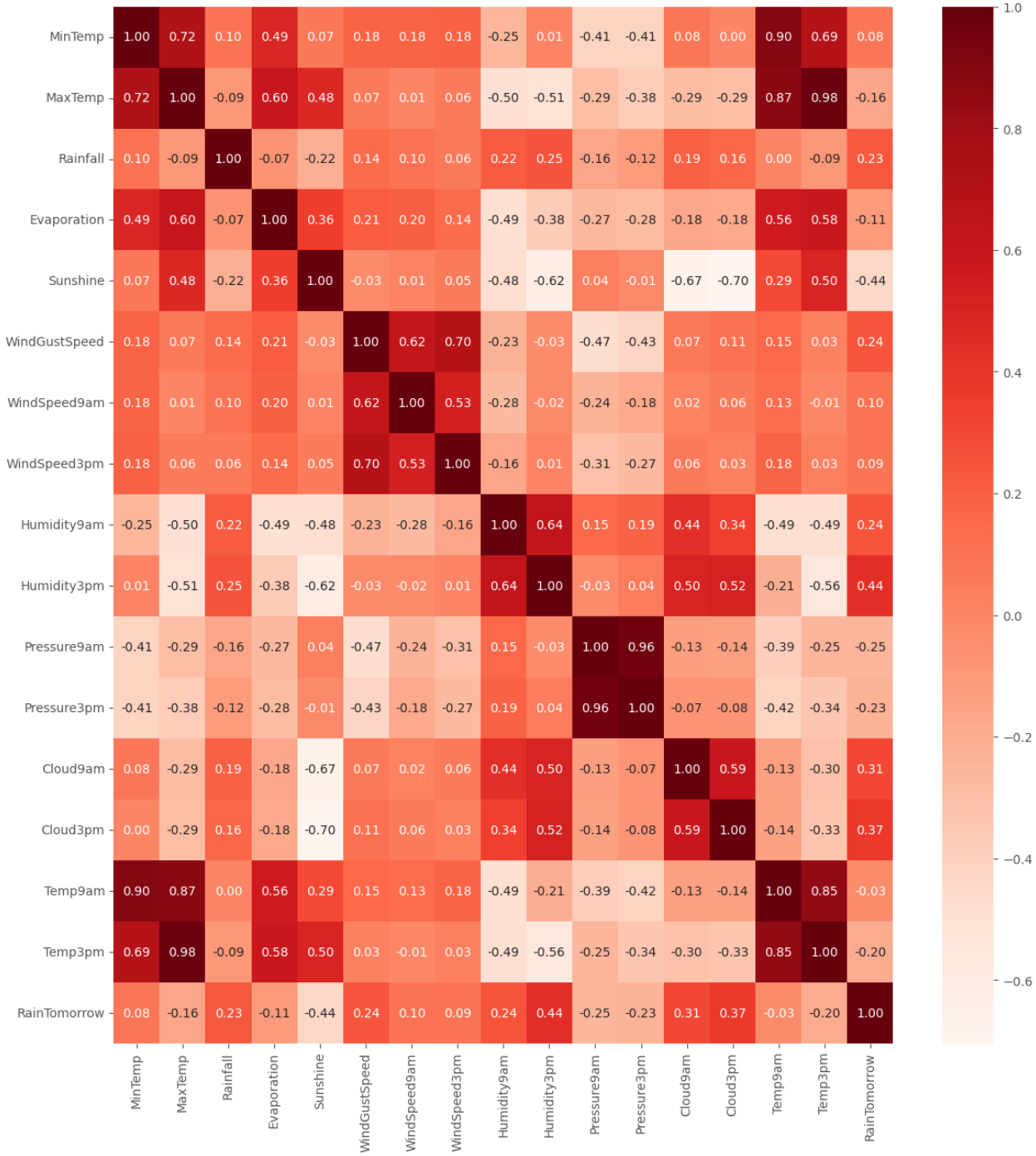
Παρουσιάζουμε το ιστόγραμμα του συνόλου δεδομένων Austalian Weather Data μέσω της συνάρτησης hist() του Matplotlib. Με τη χρήση του ιστογράμματος παρέχεται μια εικόνα των δεδομένων για κάθε χαρακτηριστικό του συνόλου δεδομένων.



***Εικόνα 16**: Ιστογράμματα συνόλου δεδομένων Australia Weather Data*

3.2.1.3 Πίνακας Συσχέτισης

Παρουσιάζεται ο πίνακας συσχέτισης του συνόλου δεδομένων Austalian Weather Data με χρήση της βιβλιοθήκης Seaborn για την οπτικοποίησή του. Στην παρακάτω εικόνα φαίνεται η συσχέτιση που υπάρχει μεταξύ των μεταβλητών του συνόλου δεδομένων.



***Εικόνα 17**: Πίνακας συσχέτισης συνόλου δεδομένων Australia Weather Data*

Όσο πιο σκούρο είναι το χρώμα τόσο πιο έντονη θετική συσχέτιση έχουμε. Για παράδειγμα στον πίνακα βλέπουμε ότι η ελάχιστη θερμοκρασία κατά τη διάρκεια της ημέρας (MinTemp) με την ελάχιστη θερμοκρασία στις 9 π.μ (Temp9am) να έχουν (μεγάλη) συσχέτιση 0.9 που παρουσιάζεται με σκούρο κόκκινο χρώμα. Αντίστοιχα όσο πιο λευκή είναι η απόχρωση του κόκκινου τόσο πιο αρνητική είναι η συσχέτιση. Για παράδειγμα, η υγρασία στις 3 μ.μ (Humidity3pm) έχει (αρνητική) συσχέτιση -0.55 με τη θερμοκρασία (Temp3pm) που παρουσιάζεται με λευκό χρώμα. Αυτό είναι λογικό καθώς όσο μεγαλύτερη θερμοκρασία έχουμε τόσο λιγότερη υγρασία υπάρχει.

3.2.2 Προ-επεξεργασία Δεδομένων

Σε αυτή την ενότητα θα γίνει η προ-επεξεργασία των δεδομένων. Όπως περιεγράφηκε και στην ενότητα 2.3 η διαδικασία αυτή περιλαμβάνει την εξάλειψη όλων των ελλειπουσών/χαμένων τιμών.

Σε πρώτη φάση διαγράφουμε την στήλη row ID από το DataFrame. Η στήλη αυτή απλά αριθμεί όλες τις γραμμές του DataFrame για όλες τις εγγραφές, χωρίς να προσφέρει κάτι ουσιαστικό.

```
data.drop(columns='row ID', inplace=True)
```

Εικόνα 18: Snippet κώδικα για διαγραφή στήλης rowID

Συνεχίζοντας θα θέλαμε να μάθουμε ποιες στήλες περιλαμβάνουν ελλείπεις ή χαμένες τιμές ώστε να αφαιρέσουμε τις αντίστοιχες εγγραφές. Κάτι τέτοιο γίνεται με το ακόλουθο snippet κώδικα:

```
total = data.isnull().sum()
percent = (data.isnull().sum() / data.isnull().count()) * 100
missing_data = pd.concat([total, percent], axis=1, keys=['Total', 'Percent'])
```

Εικόνα 19: Snippet κώδικα για εμφάνιση ποσοστών των ελλείπων τιμών

Ο πίνακας missing_data που δημιουργείται έχει την ακόλουθη μορφή:

Πίνακας 11: Ποσοστά ελλείπων τιμών στο σύνολο δεδομένων

Χαρακτηριστικά	Συνολικά	Επί τοις Εκατό (%)
Location	0	0.000000
MinTemp	443	0.445155
MaxTemp	230	0.231119
Rainfall	979	0.983761
Evaporation	42531	**42.737851**
Sunshine	47317	**47.547128**

WindGustDir	6521	6.552715
WindGustSpeed	6480	6.511516
WindDir9am	7006	7.040074
WindDir3pm	2648	2.660879
WindSpeed9am	935	0.939547
WindSpeed3pm	1835	1.843925
Humidity9am	1233	1.238997
Humidity3pm	2506	2.518188
Pressure9am	9748	9.795410
Pressure3pm	9736	9.783351
Cloud9am	37572	**37.754733**
Cloud3pm	40002	**40.196551**
Temp9am	614	0.616986
Temp3pm	1904	1.913260
RainToday	979	0.983761
RainTommorow	0	0.000000

Στον πίνακα αυτόν που προκύπτει παρατηρούμε ότι τα ποσοστά ελλιπών/χαμένων τιμών για τα γνωρίσματα Evaporation, Sunshine, Cloud9am και Cloud3pm είναι ιδιαίτερα υψηλά. Για το λόγο αυτό θα διαγράψουμε εντελώς τις στήλες αυτές με το παρακάτω snippet κώδικα:

```
data.drop(columns=['Sunshine', 'Evaporation', 'Cloud3pm', 'Cloud9am'], inplace=True)
```

***Εικόνα 20**: Snippet κώδικα για διαγραφή στηλών Sunshine, Evaporation, Cloud3pm, Cloud9am*

Παρατηρούμε ωστόσο πως και οι υπόλοιπες στήλες έχουν ελλιπή δεδομένα. Για το λόγο αυτό σε κάθε γνώρισμα που αντιστοιχεί σε αριθμητική τιμή θα αντικαταστήσουμε τις ελλιπείς/χαμένες τιμές με το μέσο όρο της στήλης του εκάστοτε γνωρίσματος (με ακρίβεια ενός δεκαδικού ψηφίου). Ο ακόλουθος κώδικας υλοποιεί ακριβώς αυτό:

```
data['MinTemp']=data['MinTemp'].fillna(round(data['MinTemp'].mean(), 1))
data['MaxTemp']=data['MaxTemp'].fillna(round(data['MaxTemp'].mean(), 1))
data['Rainfall']=data['Rainfall'].fillna(round(data['Rainfall'].mean(), 1))
data['WindGustSpeed']=data['WindGustSpeed'].fillna(round(data['WindGustSpeed'].mean(), 1))
data['WindSpeed9am']=data['WindSpeed9am'].fillna(round(data['WindSpeed9am'].mean(), 1))
data['WindSpeed3pm']=data['WindSpeed3pm'].fillna(round(data['WindSpeed3pm'].mean(), 1))
data['Humidity9am']=data['Humidity9am'].fillna(round(data['Humidity9am'].mean(), 1))
data['Humidity3pm']=data['Humidity3pm'].fillna(round(data['Humidity3pm'].mean(), 1))
data['Pressure9am']=data['Pressure9am'].fillna(round(data['Pressure9am'].mean(), 1))
data['Pressure3pm']=data['Pressure3pm'].fillna(round(data['Pressure3pm'].mean(), 1))
data['Temp9am']=data['Temp9am'].fillna(round(data['Temp9am'].mean(), 1))
data['Temp3pm']=data['Temp3pm'].fillna(round(data['Temp3pm'].mean(), 1))
```

***Εικόνα 21**: Snippet κώδικα για αντικατάσταση ελλείπων τιμών*

Έτσι οι ελλιπείς τιμές στο DataFrame έχουν διαμορφωθεί ως εξής:

***Πίνακας 12**: Πλήθος ελλείπων τιμών μετά την αντικατάσταση*

Χαρακτηριστικά	**Συνολικά**
Location	0
MinTemp	0
MaxTemp	0
Rainfall	0
WindGustDir	6521
WindGustSpeed	0
WindDir9am	7006
WindDir3pm	2648
WindSpeed9am	0
WindSpeed3pm	0
Humidity9am	0
Humidity3pm	0
Pressure9am	0
Pressure3pm	0
Temp9am	0
Temp3pm	0
RainToday	979
RainTommorow	0

Τα γνωρίσματα WindGustDir, WindDir9am, WindDir3pm λαμβάνουν ονομαστικές τιμές ανάλογα με τις 16 κατευθύνσεις του ανέμου. Ο κώδικας που δίνεται στη συνέχεια συμπληρώνει τις απουσιάζουσες τιμές των στηλών "WindGustDir", "WindDir9am", "WindDir3pm" του DataFrame, χρησιμοποιώντας την πιο συχνή τιμή της στήλης με τη χρήση της συνάρτησης fillna(). Η συνάρτηση value_counts() χρησιμοποιείται για να μετρήσει τον αριθμό εμφανίσεων κάθε μοναδικής τιμής στη στήλη "WindGustDir". Στη συνέχεια, η συνάρτηση idxmax() χρησιμοποιείται για να ανακτήσει το δείκτη (δηλαδή την πιο συχνή τιμή) της μέγιστης μετρήσεως. Τέλος η διαδικασία επαναλαμβάνεται για τις στήλες 'WindDir9am' και 'WindDir3pm'.

```
data['WindGustDir']=data['WindGustDir'].fillna(data['WindGustDir'].value_counts().idxmax())
data['WindDir9am']=data['WindDir9am'].fillna(data['WindDir9am'].value_counts().idxmax())
data['WindDir3pm']=data['WindDir3pm'].fillna(data['WindDir3pm'].value_counts().idxmax())
```

***Εικόνα 22**: Snippet κώδικα αντικατάστασης ελλείπων τιμών κατηγορηματικών μεταβλητών*

Η τεχνική αυτή ονομάζεται "mode imputation" και αποτελεί έναν δημοφιλή τρόπο για τη διαχείριση απουσιαζουσών δεδομένων σε κατηγορικές στήλες.

Τέλος για την στήλη "RainToday" όπως αναφέραμε το σύνολο δεδομένων αφορά καθημερινές μετρήσεις. Συνεπώς οι εγγραφές που έχουν κενές τιμές για το γνώρισμα "RainToday" είναι πάντα ίσες με την τιμή του "RainTomorrow" της ακριβώς από πάνω γραμμής (μια μέρα πριν). Άρα για συμπλήρωσε ελλιπών τιμών της στήλης έχουμε:

```
data['RainToday'] = data['RainToday'].fillna(data['RainTomorrow'].shift())
```

***Εικόνα 23**: Snippet κώδικα αντικατάστασης ελλείπων τιμών στήλης RainToday*

Πλέον το σύνολο δεδομένων μας δεν περιέχει κάποια null τιμή σε κανένα γνώρισμα. Σε αυτό το σημείο ολοκληρώνεται η διαδικασία της προ-επεξεργασίας των δεδομένων μας.

3.2.3 Μετασχηματισμός Δεδομένων

Όπως αναφέρθηκε και στην ενότητα 2.3 σε αυτό το στάδιο, τα δεδομένα μετασχηματίζονται ώστε να ταιριάζουν καλύτερα στις τεχνικές εξόρυξης δεδομένων που θα χρησιμοποιηθούν. Τόσο στη βιβλιοθήκη scikit-learn όσο και στο WEKA, οι αλγόριθμοι μηχανικής μάθησης στηρίζονται σε μαθηματικά μοντέλα που βασίζονται σε αριθμητικές πράξεις και λειτουργίες. Οι κατηγορικές μεταβλητές δεν μπορούν να χρησιμοποιηθούν απευθείας σε αλγόριθμους μηχανικής μάθησης, καθώς οι αλγόριθμοι δεν μπορούν να χειριστούν σύμβολα ή κατηγορίες, κρίνοντας έτσι απαραίτητη τη μετατροπή των τιμών αυτών σε αριθμητικές προκειμένου να προχωρήσουμε με την εξόρυξη δεδομένων.

Στο σύνολο δεδομένων μας τα γνωρίσματα με κατηγορηματικές μεταβλητές είναι Location, WindGustDir, WindDir9am, WindDir3pm. Το γνώρισμα Location περιέχει το όνομα κάποιας από τις 10 πόλεις του συνόλου δεδομένων τις Αυστραλίας. Εντελώς αντίστοιχα, οι μεταβλητές WindGustDir, WindDir9am, WindDir3pm περιέχουν την κατεύθυνση του ανέμου με βάση τους 16 κατευθυντήρες ανέμου. Έτσι, μετασχηματίζουμε αυτές τις στήλες του DataFrame με το παρακάτω snippet κώδικα:

```
location_dict=dict(zip(data['Location'].unique(), range(data['Location'].nunique())))
data = data.replace(location_dict)

#Η unique παίρνει τις μοναδικές τιμές και η zip φτιάχνει τα ζευγάρια

wind_dict=dict(zip(data['WindGustDir'].unique(), range(data['WindGustDir'].nunique())))
data = data.replace(wind_dict)

data.loc[data.RainToday == "Yes", "RainToday"] = 1
data.loc[data.RainToday == "No", "RainToday"] = 0
data['RainToday'] = data['RainToday'].astype(int)
```

***Εικόνα 24**: Snippet κώδικα μετατροπής κατηγορηματικών τιμών σε αριθμητικές*

Με αυτό το snippet κώδικά:

1. Αρχικά δημιουργούμε ένα λεξικό location_dict που αντιστοιχεί κάθε μοναδική τοποθεσία της στήλης 'Location' με έναν αριθμό που με τη βοήθεια των συναρτήσεων zip(), unique() και nunique(). Με αυτό τον τρόπο, δημιουργούμε έναν μοναδικό ακέραιο αριθμό για κάθε μοναδική τοποθεσία, που μπορεί να χρησιμοποιηθεί για αντικατάσταση των (ονομαστικών) τιμών στην στήλη 'Location'. Στη συνέχεια, με τη μέθοδο replace() αντικαθιστούμε τις τιμές της στήλης 'Location' με τους αντίστοιχους αριθμούς που έχουν προσδιοριστεί στο location_dict.
2. Η ίδια διαδικασία επαναλαμβάνεται για τη στήλη 'WindGustDir'. Αφού δημιουργήσουμε ένα λεξικό με όνομα wind_dict, αντιστοιχίζουμε κάθε μοναδική κατεύθυνση ανέμου της στήλης 'WindGustDir' με έναν αριθμό. Έπειτα, καλούμε πάλι τη replace() για να αντικαταστήσει τις τιμές της στήλης 'WindGustDir' με τους αντίστοιχους αριθμούς από το wind_dict.
3. Τέλος χρησιμοποιούμε τη μέθοδο loc για να επιλέξει τις γραμμές όπου η τιμή της στήλης 'RainToday' είναι "Yes" ή "No" και στη συνέχεια, αντικαθιστούμε τις τιμές "Yes" με 1 και τις τιμές "No" με 0. Τέλος, μετατρέπουμε τον τύπο της στήλης 'RainToday' σε ακέραιο, χρησιμοποιώντας την astype(int).

Σε αυτό το σημείο ολοκληρώθηκε και η διαδικασία του μετασχηματισμού δεδομένων. Σε αυτό το σημείο θα πρέπει να αναφέρουμε ότι υπάρχουν πολλοί τρόποι μετατροπής των κατηγορικών δεδομένων σε αριθμητικά που ενδεχομένως να ήταν και πιο αποτελεσματικοί. Για παράδειγμα, θα μπορούσαμε να χρησιμοποιήσουμε τη συνάρτηση get_dummies(), ή την τεχνική One Hot Encoding,

όμως κάτι τέτοιο θα δυσχέραινε την μετατροπή του .csv αρχείου σε .arff που είναι ο υποστηριζόμενος τύπος αρχείου του WEKA όπως θα δούμε στη συνέχεια.

3.2.4 Εξαγωγή του .csv αρχείου και μετατροπή του σε .arff

Η ενότητα αυτή με αποτελεί κάποιο επιμέρους στάδιο της KDD διαδικασίας που ακολουθούμε, απλά στοχεύει στην αναλυτικότερη περιγραφή της δημιουργίας του .arff αρχείου που θα χρησιμοποιήσουμε για να εφαρμόσουμε τους αλγορίθμους μηχανικής μάθησης στο WEKA. Αρχικά στο colab:

```
x = data.to_csv()
with open("weka_data_weather.csv", "w") as file:
  file.write(x)
```

***Εικόνα 25**: Snippet κώδικα εξαγωγής .csv αρχείου*

Με το παραπάνω snippet κώδικα αντιγράφουμε τα δεδομένα του DataFrame **data** στο αρχείο "weka_data_weather.csv" με τη μέθοδο **to_csv()** που μετατρέπει τα δεδομένα σε μορφή κειμένου σε μορφή CSV (Comma Separated Values) και τα επιστρέφει ως αποτέλεσμα. Στη συνέχεια, το csv κείμενο αποθηκεύεται στο αρχείο "weka_data_weather.csv" χρησιμοποιώντας τη μέθοδο **write()** του αρχείου ώστε ύστερα με ένα notepad να τα φέρουμε στη μορφή:

```
@relation weather_data

@attribute Location real
@attribute MinTemp real
@attribute MaxTemp real
@attribute Rainfall real
@attribute WindGustDir real
@attribute WindGustSpeed real
@attribute WindDir9am real
@attribute WindDir3pm real
@attribute WindSpeed9am real
@attribute WindSpeed3pm real
@attribute Humidity9am real
@attribute Humidity3pm real
@attribute Pressure9am real
@attribute Pressure3pm real
@attribute Temp9am real
@attribute Temp3pm real
@attribute RainToday real
@attribute 'class' {0,1}

@data
0,134,229,6,0,440,0,1,200,240,710,220,10077,10071,169,218,0,0
0,74,251,0,1,440,8,9,40,220,440,250,10106,10078,172,243,0,0
0,175,323,10,0,410,5,12,70,200,820,330,10108,10060,178,297,0,0
0,146,297,2,1,560,0,0,190,240,550,230,10092,10054,206,289,0,0
0,77,267,0,0,350,6,0,60,170,480,190,10134,10101,163,255,0,0
0,131,301,14,0,280,10,6,150,110,580,270,10070,10057,201,282,1,0
0,134,304,0,2,300,6,11,170,60,480,220,10118,10087,204,288,0,1
```

***Εικόνα 26**: Αρχείο .arff*

3. 3 Επιλογή αλγορίθμων

Οι αλγόριθμοι που χρησιμοποιήθηκαν στην παρούσα διπλωματική εργασία για την εξόρυξη δεδομένων (που αποτελεί το 4ο στάδιο της διαδικασίας ανακάλυψης γνώσης) είναι υλοποιημένοι σε Python, με χρήση της βιβλιοθήκης Scikit-Learn και σε Java για το παραθυρικό περιβάλλον του εργαλείου WEKA. Η επιλογή των αλγορίθμων βασίστηκε σε πολλούς παράγοντες. Κύριο μέλημα για λόγους σύγκρισης είναι ο κάθε αλγόριθμος που επρόκειτο να χρησιμοποιηθεί ήταν να βρίσκεται υλοποιημένος τόσο σε Scikit-learn όσο και στο WEKA. Παρακάτω παρουσιάζονται μερικοί από τους λόγους που χρησιμοποιήθηκαν οι συγκεκριμένοι αλγόριθμοι:

- Ο αλγόριθμος KNN (k-Nearest Neighbors) είναι ο πρώτος από τους αλγόριθμους που θα μελετήσουμε και αποτελεί έναν από πιο απλούς αλγορίθμους για κατηγοριοποίηση, ο οποίος βασίζεται στη χρήση διαφόρων μέτρων απόστασης μεταξύ των δεδομένων εισόδου και των κατηγοριών εξόδου που αναφέρθηκαν στο κεφάλαιο 2. Ο αλγόριθμος KNN αναζητά τους k πλησιέστερους γείτονες της εισόδου και ταξινομεί την είσοδο στην κατηγορία των περισσότερων γειτόνων. Η απόδοση του αλγορίθμου KNN μπορεί να διαφέρει ανάλογα με το μέτρο απόστασης που χρησιμοποιείται, καθιστώντας σημαντική την επιλογή του κατάλληλου μέτρου για την επίτευξη καλών αποτελεσμάτων.
- Η λογιστική παλινδρόμηση αποτελεί έναν μη γραμμικό κατηγοριοποιητή, που χρησιμοποιεί ένα γραμμικό συνδυασμό παραμέτρων για να αναγνωρίσει τη σχέση μεταξύ των εισόδων και των κατηγοριών εξόδου χωρίς τη χρήση σιγμοειδούς συνάρτησης. Είναι ιδανική για σύνολα δεδομένων που αποτελούνται από δύο κατηγορίες.
- Τα δένδρα απόφασης (Decision Trees) αποτελούν μια καλή λύση για προβλήματα κατηγοριοποίησης. Ο τρόπος λειτουργίας τους είναι αρκετά παρόμοιος με τη διαδικασία λήψης απόφασης από τον άνθρωπο. Χρησιμοποιούνται μόνα τους αλλά και σε συνδυασμό, όπως στα τυχαία δάση (Random Forest) ή στον αλγόριθμο AdaBoost.
- Ο απλοϊκός Bayes (Naïve Bayes) είναι ένας από τους πιο διαδεδομένους αλγορίθμους κατηγοριοποίησης που βασίζεται στο θεώρημα Bayes. Στην παρούσα διπλωματική εργασία χρησιμοποιείται ο απλοϊκός Bayes ακολουθώντας την κανονική κατανομή για την πρόβλεψη δυαδικών αποτελεσμάτων 0 ή 1. Ο αλγόριθμος επεξηγήθηκε στο κεφάλαιο 2.
- Τα νευρωνικά δίκτυα είναι μια πολύ ισχυρή και ευέλικτη μέθοδος μηχανικής μάθησης που μπορεί να αναγνωρίσει και να εξάγει πολύπλοκα μοτίβα από τα δεδομένα. Η λειτουργία τους βασίζεται στην επεξεργασία των δεδομένων μέσω συνελικτικών ή αναδραστικών στρωμάτων, όπως ένα σύνολο νευρώνων στον εγκέφαλο.

- Σε πολλές περιπτώσεις τα δεδομένα δεν είναι γραμμικά διαχωρίσιμα, και έτσι χρειάζεται να χρησιμοποιηθούν αλγόριθμοι που μπορούν να επεκτείνουν τον χώρο χαρακτηριστικών σε έναν υψηλότερης διάστασης χώρο, χρησιμοποιώντας πυρήνες. Στην παρούσα εργασία χρησιμοποιήθηκε ο αλγόριθμος Support Vector Machines (SVMs), που είναι κατάλληλος για την ταξινόμηση δεδομένων μεγάλων διαστάσεων.

3. 4 Επιλογή Παραμέτρων

Σε αυτή την ενότητα θα παρουσιάσουμε τις παραμέτρους των αλγορίθμων μηχανικής μάθησης που θα χρησιμοποιήσουμε και αναφέρθηκαν παραπάνω. Η επιλογή των παραμέτρων για την εφαρμογή των αλγορίθμων μηχανικής μάθησης λαμβάνει χώρα κατά την διαδικασία εξόρυξης δεδομένων που όπως αναφέρθηκε αποτελεί το 4ο στάδιο της διαδικασίας ανακάλυψης γνώσης. Κατά τη διαδικασία επιλογής παραμέτρων έγινε προσπάθεια στο βαθμό που είναι αυτό εφικτό οι αλγόριθμοι μηχανικής μάθησης στο WEKA και στο scikit-learn να έχουν όσο το δυνατόν ίδιες παράμετρούς βάσει των εκάστοτε documentations που βρίσκονται στην διεύθυνση: https://scikit-learn.org/stable/ για το scikit-learn και στη διεύθυνση: https://weka.sourceforge.io/doc.dev/overview-summary.html για το WEKA.

3.4.1 Κ – Πλησιέστεροι Γείτονες (KNN):

Στο περιβάλλον του Scikit-learn:

KNeighborsClassifier(): Θα χρησιμοποιηθεί με 5 παράμετρούς. Η παράμετρος n_neighbors=140 υποδηλώνει τον αριθμό των πλησιέστερων γειτόνων που χρησιμοποιούνται στην ταξινόμηση, που είναι ίδια με την επιλεγμένη τιμή που θα χρησιμοποιηθεί από τον Weka IBk classifier. Η παράμετρος weights='uniform' υποδηλώνει ότι όλα τα σημεία σε κάθε γειτονιά πρέπει να ζυγίζονται ίσα, όπως και στο προεπιλεγμένο σχήμα ζύγισης που χρησιμοποιείται από τον Weka IBk classifier. Η παράμετρος algorithm='ball tree' αφορά τον αλγόριθμο που χρησιμοποιείται για την αναζήτηση των γειτόνων. Τέλος, η παράμετρος metric='minkowski' καθορίζει τη μετρική απόστασης που θα χρησιμοποιηθεί για το δέντρο ενώ η παράμετρος p=2 υποδηλώνει την παράμετρο δύναμης για τη μετρική απόστασης Minkowski.

Πίνακας 13: Παράμετροι KNeighborsClassifier()

Παράμετρος	Τιμή
n_neighbors	140
metric	minkowski

algorithm	ball_tree
weights	uniform
p	2

Αντίστοιχα στο WEKA:

Επιλέγουμε τον ταξινομητή IBk που μοντελοποιεί τον αλγόριθμο KNN και ύστερα τροποποιούμε 2 παραμέτρους. Ο αριθμός των γειτόνων θα πάρει την τιμή 140 όπως και στο scikit-learn. Ο αλγόριθμος που θα χρησιμοποιηθεί για την αναζήτηση των γειτόνων όπως και στο scikit-learn χρησιμοποιήσουμε τον αλγόριθμο BallTree.

***Πίνακας 14**: Παράμετροι IBk*

Παράμετρος	**Τιμή**
KNN	140
nearestNeighbourAlgorithm	BallTree

3.4.2 Λογιστική Παλινδρόμηση

Στο περιβάλλον του Scikit-learn:

LogisticRegression(): Θα χρησιμοποιηθεί με 3 παραμέτρους. Η παράμετρος solver='lbfgs' υποδηλώνει ότι το μοντέλο θα χρησιμοποιήσει τον L-BFGS solver, που είναι ο solver που χρησιμοποιείται και από το Weka logistic classifier. Τέλος, η max_iter=1000 ρυθμίζει το μέγιστο αριθμό επαναλήψεων που θα εκτελεστούν από τον αλγόριθμο κατά τη διαδικασία εκπαίδευσης.

***Πίνακας 15**: Παράμετροι LogisticRegression()*

Παράμετρος	**Τιμή**
solver	lbfgs
max_iter	1000

Αντίστοιχα στο περιβάλλον του Weka:

Επιλέγουμε τον ταξινομητή **Logistic**. Θα χρησιμοποιηθεί με μια παράμετρο, τον αριθμό των επαναλήψεων όπου θα πάρει και εδώ την τιμή 1000:

***Πίνακας 16**: Παράμετροι Logistic*

Παράμετρος	**Τιμή**
maxIts	1000

3.4.3 Δέντρα Απόφασης

Στο περιβάλλον του Scikit-learn:

DecisionTreeClassifier(): Θα χρησιμοποιηθεί με 2 παράμετρους. Η παράμετρος criterion='entropy' καθορίζει ότι το δέντρο απόφασης θα χρησιμοποιεί το κριτήριο κέρδους πληροφορίας, που είναι το κριτήριο που χρησιμοποιείται και από τον J48 στο WEKA. Η παράμετρος ccp_alpha=0.0001 ρυθμίζει τη στρατηγική περικοπής του δέντρου απόφασης, θέτοντας ένα κατώφλι για το κόστος της περικοπής ώστε να αποφύγουμε την υπερεκπαίδευση.

__Πίνακας 17__: Παράμετροι DecisionTreeClassifier()

Παράμετρος	Τιμή
criterion	entropy
ccp_alpha	0.0001

Αντίστοιχα για το περιβάλλον του WEKA:

Επιλέγουμε τον ταξινομητή J48 που μοντελοποιεί ένα δέντρο απόφασης. Θα χρησιμοποιηθεί με τις προεπιλεγμένες παραμέτρους. Ενδεικτικά μερικές από τις παραμέτρους είναι οι παρακάτω:

__Πίνακας 18__:Παράμετροι J48

Παράμετρος	Τιμή
batch_size	100
confidenceFactor	0.25
numFolds	3
unpruned	False

3.4.4 Random Forest

Στο περιβάλλον του Scikit-learn:

RandomForestClassifier(): Οι τιμές των παραμέτρων είναι οι προκαθορισμένες (default) τιμές της βιβλιοθήκης scikit-learn. Ενδεικτικά η παράμετρος n_estimators=100 καθορίζει τον αριθμό των δέντρων απόφασης που θα χρησιμοποιηθούν στο σύνολο δεδομένων. Η παράμετρος criterion='gini' καθορίζει ότι θα χρησιμοποιηθεί η Gini αμφισβήτηση για τη μέτρηση της ποιότητας μιας διαίρεσης, που αποτελεί τη μέθοδο αμφισβήτησης που χρησιμοποιείται από τον αλγόριθμο Random Forest στο Weka. Η παράμετρος max_depth=None καθορίζει ότι ο μέγιστος βάθος των δέντρων απόφασης δεν

περιορίζεται, ενώ η παράμετρος min_samples_split=2 καθορίζει τον ελάχιστο αριθμό δειγμάτων που απαιτούνται για να διαιρεθεί ένα εσωτερικό κόμβο. Η παράμετρος min_samples_leaf=1 καθορίζει το ελάχιστο αριθμό δειγμάτων που απαιτούνται στο φύλλο ενός κόμβου και τέλος η παράμετρος bootstrap=True καθορίζει ότι οι δέντρα απόφασης θα κατασκευάζονται σε bootstrap δείγματα του συνόλου δεδομένων

Πίνακας 19: Παράμετροι RandomForestClassifier

Παράμετρος	Τιμή
n_estimators	100
criterion	gini

Αντίστοιχα στο εργαλείο WEKA:

Επιλέγουμε τον ταξινομητή Random Forest. Οι τιμές των παραμέτρων είναι οι προκαθορισμένες (default) τιμές του εργαλείου WEKA. Ενδεικτικά:

Πίνακας 20: Παράμετροι RandomForest (WEKA)

Παράμετρος	Τιμή
bagSizePercent	100
batch_size	100
max_depth	0
numIterations	100

3.4.5 AdaBoost

Στο περιβάλλον του Scikit-learn:

AdaBoostClassifier(): Οι τιμές των παραμέτρων είναι οι προκαθορισμένες (default) τιμές της βιβλιοθήκης scikit-learn. Η παράμετρος base_estimator=DecisionTreeClassifier(max_depth=1) δηλώνει ότι ο βασικός εκτιμητής για το μοντέλο AdaBoost θα πρέπει να είναι ένας ταξινομητής δέντρου απόφασης με μέγιστο βάθος 1, όπως και στον αλγόριθμο AdaBoostM1 στο Weka. Η παράμετρος algorithm='SAMME' δηλώνει ότι το μοντέλο θα χρησιμοποιεί τον αλγόριθμο SAMME για την ενημέρωση των βαρών των δειγμάτων:

Πίνακας 21: Παράμετροι AdaBoostClassifier()

Παράμετρος	Τιμή
base_estimator	DecisionTreeClassifier(max_depth=1)
algorithm	SAMME

Αντίστοιχα στο περιβάλλον του WEKA οι τιμές των παραμέτρων είναι οι προκαθορισμένες (default) τιμές του εργαλείου WEKA. Ενδεικτικά:

Πίνακας 22: Παράμετροι AdaBoostM1

Παράμετρος	Τιμή
classifier	J48
numIterations	10
weightThreshold	100

3.3.6 Naïve Bayes

Στο περιβάλλον του Scikit-learn:

GaussianNB(): Οι τιμές των παραμέτρων είναι οι προκαθορισμένες (default) τιμές της βιβλιοθήκης scikit-learn. Ενδεικτικά:

Πίνακας 23: Παράμετροι GaussianNB()

Παράμετρος	Τιμή
priors	None
var_smoothing	1e-9

Αντίστοιχα στο περιβάλλον του WEKA πάλι θα χρησιμοποιήσουμε τις προεπιλεγμένες παραμέτρους του κατηγοριοποιητή. Ενδεικτικά:

Πίνακας 24: Παράμετροι NaiveBayes()

Παράμετρος	Τιμή
batchSize	100
useKernelEstimator	False
useSupervisedDiscretization	False

3.4.7 Support Vector Machines

Στο περιβάλλον του Scikit-learn:

SVC(): Θα χρησιμοποιηθεί με μια παράμετρο. Η παράμετρος kernel='poly' χρησιμοποιείται για να ορίσει τον πυρήνα που καθορίζει τον τρόπο με τον οποίο τα δεδομένα θα αντιστοιχηθούν σε ένα χώρο υψηλότερης διάστασης ώστε να μπορούν να διαχωριστούν με μια επίπεδη επιφάνεια (hyperplane). Η

παράμετρος C=1.0 καθορίζει την ποινή για το σφάλμα, η οποία μπορεί να ρυθμιστεί για να ελέγξει την ισορροπία μεταξύ του μεγιστοποίηση του περιθωρίου και της ελαχιστοποίησης του σφάλματος κατηγοριοποίησης.

Πίνακας 25: Παράμετροι SVC()

Παράμετρος	Τιμή
kernel	poly
C	1.0

Αντίστοιχα στο εργαλείο WEKA ορίζουμε τον πύρινα και την παράμετρο C με τις ίδιες τιμές του scikit-learn:

Πίνακας 26: Παράμετροι SMO

Παράμετρος	Τιμή
kernel	Polykernel
C	1.0

3.4.8 Multi-Layer Perceptron

Στο περιβάλλον του Scikit-learn:

MLP(): Θα χρησιμοποιηθεί με δυο παραμέτρους. Η παράμετρος solver που ορίζει τον αλγόριθμο βελτιστοποίησης που θα χρησιμοποιηθεί για την εκπαίδευση του MLP θα πάρει την τιμή sgd. Η παράμετρος activation='logistic' καθορίζει ότι η συνάρτηση ενεργοποίησης για τους κρυφούς νευρώνες πρέπει να είναι η λογιστική σιγμοειδής συνάρτηση, η οποία είναι η ίδια συνάρτηση που χρησιμοποιείται από τον αλγόριθμο MLP του WEKA. Η παράμετρος learning_rate_init=0.001 καθορίζει τον αρχικό ρυθμό μάθησης και μπορεί να προσαρμοστεί ανάλογα με την πολυπλοκότητα του συνόλου δεδομένων.

Πίνακας 27: Παράμετροι MLPClassifier()

Παράμετρος	Τιμή
solver	'sgd'
activation	'logistic'
learning_rate_init	0.0001

Αντίστοιχα στο εργαλείο WEKA:

Πίνακας 28: Παράμετροι MultiLayer Perceptron

Παράμετρος	Τιμή
batchSize	200
hiddenLayers	a
learningRate	0.0001
momentum	0.9

Κεφάλαιο 4: Αποτελέσματα

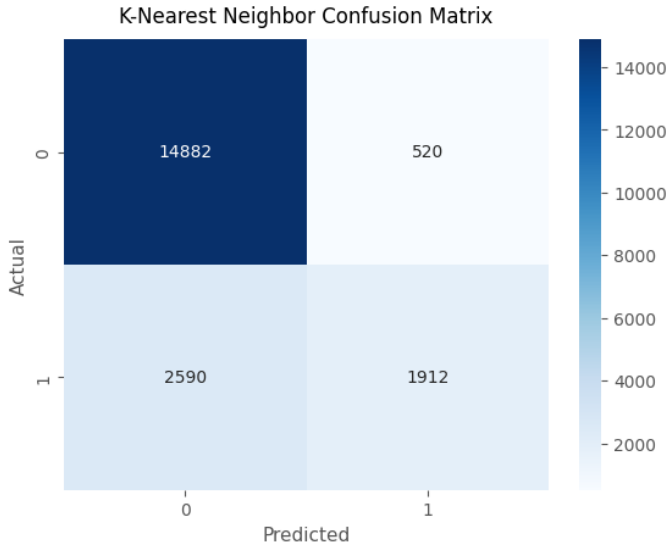
Σε αυτό το κεφάλαιο της διπλωματικής θα παρουσιαστούν τα αποτελέσματα της κατηγοριοποίησης των ταξινομητών. Για κάθε αλγόριθμο ταξινόμησης θα παρουσιαστούν οι αντίστοιχοι πίνακες σύγχυσης της βιβλιοθήκης scikit-learn και του WEKA, καθώς και οι μετρικές της ορθότητας, της ακρίβειας, της ανάκλησης και του αρμονικού μέσου μέσω ενός συγκριτικού διαγράμματος των μετρικών μεταξύ του WEKA και του scikit-learn. Τέλος αφού γίνει συγκριτική αξιολόγηση θα επιλεχθεί ο βέλτιστος αλγόριθμος μηχανικής μάθησης με τα υψηλότερα ποσοστά για την πρόβλεψη των καιρικών συνθηκών του συνόλου δεδομένων Australia Weather Data.

4.1 Προβλέψεις αλγόριθμων μηχανικής μάθησης

Τώρα θα παρουσιαστούν οι προβλέψεις των ταξινομητών (classifiers). Οι αλγόριθμοι αυτοί είναι οι Κ-κοντινότεροι γείτονες (K-nearest-neighboors - kNN), , η λογιστική παλινδρόμηση (logistic regression), τα δέντρα απόφασης (Decision Trees), τα τυχαία δάση (Random Forest), ο AdaBoost, ο απλοϊκός Bayes, οι μηχανές διανυσμάτων υποστήριξης και τα νευρωνικά δίκτυα πολλών επιπέδων.

4.1.1 Κ – Πλησιέστεροι Γείτονες (KNN)

Παρακάτω παρουσιάζονται τα αποτελέσματα του 1ου κατηγοριοποιητή KNeighborsClassifier που είναι υλοποιημένος στη βιβλιοθήκη scikit-learn.



***Εικόνα 27**: Πίνακας Σύγχυσης KNN (scikit-learn)*

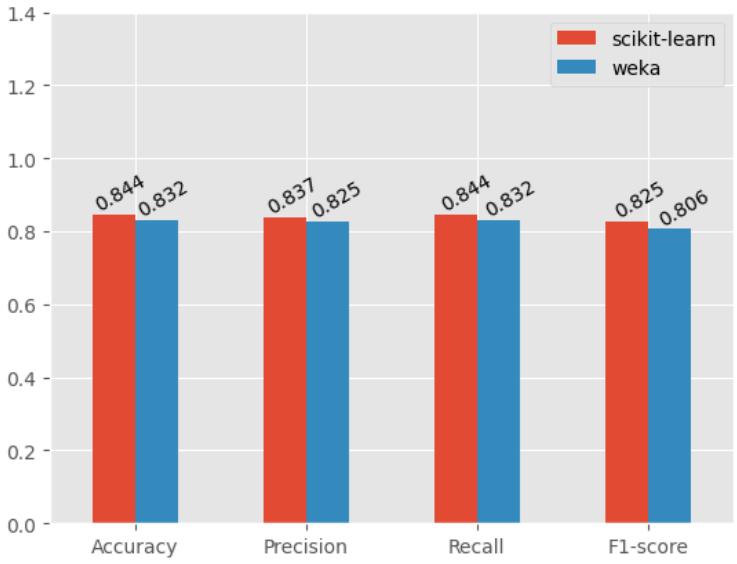
Στον παραπάνω πίνακα σύγχυσης (confusion matrix) του κατηγοριοποιητή KNeighborsClassifier παρατηρούμε ότι το μοντέλο πρόβλεψε 1912 αληθινές βροχερές μέρες (True Positive), 14882 αληθινές μη βροχερές μέρες (True Negative), 520 ψευδείς θετικές βροχερές μέρες (False Positive) και 2590 ψευδώς αρνητικές μη βροχερές μέρες (False Negative).

Παρακάτω παρουσιάζονται τα αποτελέσματα του κατηγοριοποιητή IBk στο WEKA:

```
Correctly Classified Instances       16567               83.2387 %
Incorrectly Classified Instances      3336               16.7613 %
Kappa statistic                          0.3976
Mean absolute error                      0.2469
Root mean squared error                  0.3481
Relative absolute error                 70.9814 %
Root relative squared error             83.5706 %
Total Number of Instances            19903

=== Detailed Accuracy By Class ===

               TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
               0,972    0,654    0,838      0,972    0,900      0,445    0,847     0,945     0
               0,346    0,028    0,782      0,346    0,480      0,445    0,847     0,660     1
Weighted Avg.  0,832    0,514    0,825      0,832    0,806      0,445    0,847     0,881

=== Confusion Matrix ===

     a     b   <-- classified as
 15026   429 |     a = 0
  2907  1541 |     b = 1
```

***Εικόνα 28**: Αποτελέσματα KNN (WEKA)*

Στον πίνακα σύγχυσης (confusion matrix) του κατηγοριοποιητή IBk στο WEKA παρατηρούμε ότι το μοντέλο πρόβλεψε 1541 αληθινές βροχερές μέρες (True Positive), 15026 αληθινές μη βροχερές μέρες (True Negative), 429 ψευδείς θετικές βροχερές μέρες (False Positive) και 2907 ψευδώς αρνητικές μη βροχερές μέρες (False Negative).

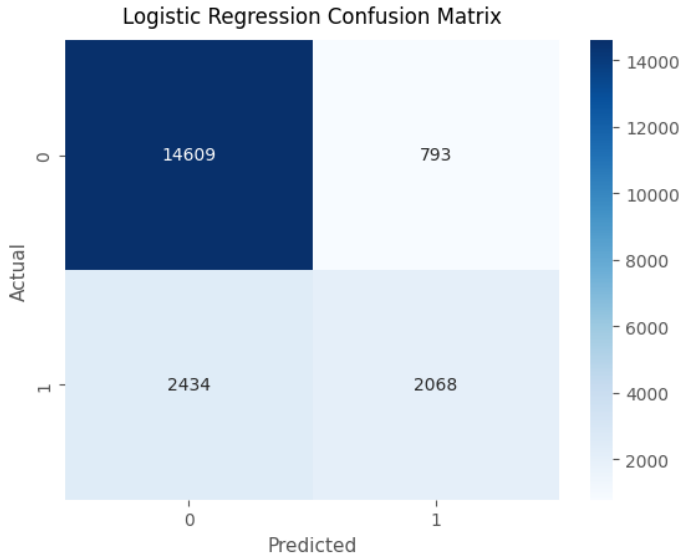


***Γραφική Παράσταση 1**: Συγκριτικό διάγραμμα αποτελεσμάτων k-NN*

Στο παραπάνω διάγραμμα παρατηρούμε ότι ο κατηγοριοποιητής KNeighborsClassifier στο scikit-learn (κόκκινο χρώμα) έχει ορθότητα (accuracy) 0.844, έχει ακρίβεια (precision) 0.837, έχει ανάκληση (recall) 0.844 και έχει αρμονικό μέσο (f1 score) 0.825. Αντίστοιχα ο κατηγοριοποιητής IBk στο WEKA (μπλε χρώμα) έχει ορθότητα (accuracy) 0.832, έχει ακρίβεια (precision) 0.825, έχει ανάκληση (recall) 0.832 και έχει αρμονικό μέσο (f1 score) 0.806.

4.1.2 Λογιστική Παλινδρόμηση

Παρακάτω παρουσιάζονται τα αποτελέσματα του 2ου κατηγοριοποιητή LogisticRegression που είναι υλοποιημένος στη βιβλιοθήκη scikit-learn.



***Εικόνα 29**: Πίνακας Σύγχυσης Λογιστικής Παλινδρόμησης (scikit-learn)*

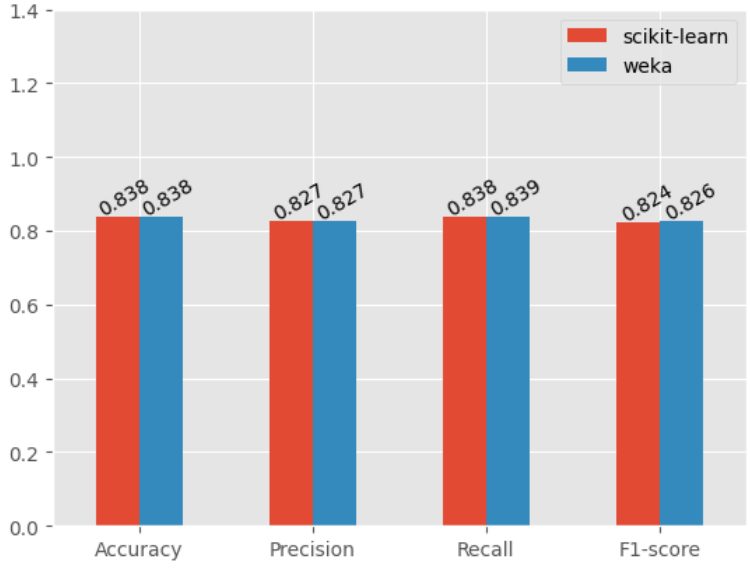
Στον παραπάνω πίνακα σύγχυσης (confusion matrix) του κατηγοριοποιητή LogisticRegression παρατηρούμε ότι το μοντέλο πρόβλεψε 2068 αληθινές βροχερές μέρες, (True Positive), 14609 αληθινές μη βροχερές μέρες (True Negative), 793 ψευδείς θετικές βροχερές μέρες (False Positive) και 2434 ψευδώς αρνητικές μη βροχερές μέρες (False Negative).

Παρακάτω παρουσιάζονται τα αποτελέσματα του 2ου κατηγοριοποιητή Logistic στο WEKA:

```
Correctly Classified Instances       16695               83.8818 %
Incorrectly Classified Instances      3208               16.1182 %
Kappa statistic                          0.4716
Mean absolute error                      0.2319
Root mean squared error                  0.3418
Relative absolute error                 66.642  %
Root relative squared error             82.0415 %
Total Number of Instances            19903

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure  MCC      ROC Area  PRC Area  Class
               0,945    0,531    0,861      0,945    0,901      0,487    0,851     0,947     0
               0,469    0,055    0,711      0,469    0,566      0,487    0,851     0,661     1
Weighted Avg.  0,839    0,424    0,827      0,839    0,826      0,487    0,851     0,883

=== Confusion Matrix ===

     a     b   <-- classified as
 14607   848 |     a = 0
  2360  2088 |     b = 1
```

***Εικόνα 30**: Αποτελέσματα Λογιστικής Παλινδρόμησης (WEKA)*

Στον πίνακα σύγχυσης (confusion matrix) του κατηγοριοποιητή Logistic στο WEKA παρατηρούμε ότι το μοντέλο πρόβλεψε 2088 αληθινές βροχερές μέρες (True Positive), 14607 αληθινές μη βροχερές μέρες (True Negative), 848 ψευδείς θετικές βροχερές μέρες (False Positive) και 2360 ψευδώς αρνητικές μη βροχερές μέρες (False Negative).

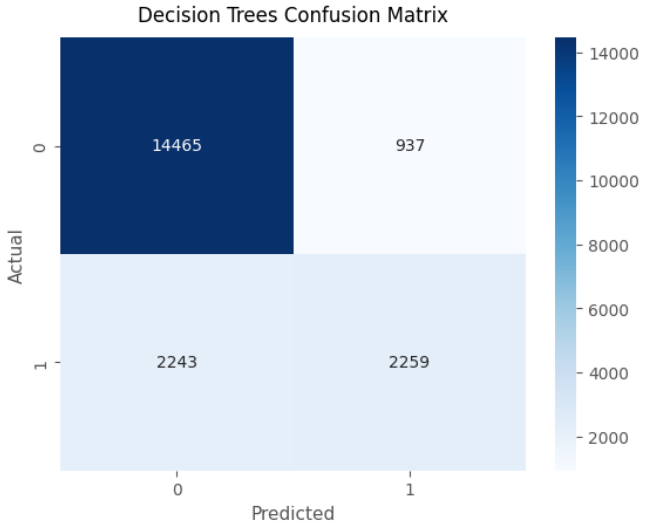


***Γραφική Παράσταση 2**: Συγκριτικό διάγραμμα αποτελεσμάτων Λογιστικής Παλινδρόμησης*

Στο παραπάνω διάγραμμα παρατηρούμε ότι ο 2ος κατηγοριοποιητής LogisticRegression στο scikit-learn (κόκκινο χρώμα) έχει ορθότητα (accuracy) 0.838, έχει ακρίβεια (precision) 0.827, έχει ανάκληση (recall) 0.838 και έχει αρμονικό μέσο (f1 score) 0.824. Αντίστοιχα ο κατηγοριοποιητής Logistic στο WEKA (μπλέ χρώμα) έχει ορθότητα (accuracy) 0.838, έχει ακρίβεια (precision) 0.827, έχει ανάκληση (recall) 0.839 και έχει αρμονικό μέσο (f1 score) 0.826.

4.1.3 Δέντρα Απόφασης

Παρακάτω παρουσιάζονται τα αποτελέσματα του 3ου κατηγοριοποιητή DecisionTreeClassifier που είναι υλοποιημένος στη βιβλιοθήκη scikit-learn.



***Εικόνα 31**: Πίνακας Σύγχυσης Δέντρων Απόφασης (scikit-learn)*

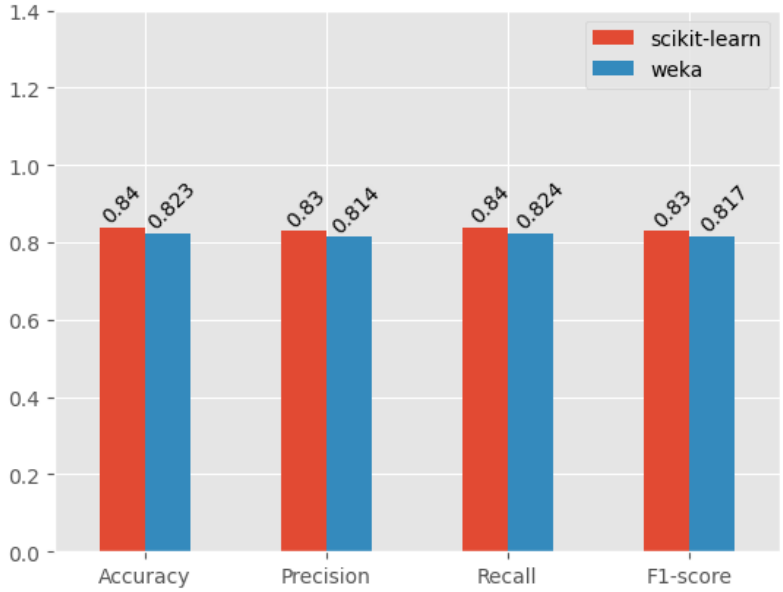
Στον παραπάνω πίνακα σύγχυσης (confusion matrix) του κατηγοριοποιητή DecisionTreeClassifier παρατηρούμε ότι το μοντέλο πρόβλεψε 2259 αληθινές βροχερές μέρες (True Positive), 14465 αληθινές μη βροχερές μέρες (True Negative), 937 ψευδείς θετικές βροχερές μέρες (False Positive) και 24243 ψευδώς αρνητικές μη βροχερές μέρες (False Negative).

Παρακάτω παρουσιάζονται τα αποτελέσματα του 3ου κατηγοριοποιητή J48 που μοντελοποιεί ένα δέντρο απόφασης στο WEKA:

```
Correctly Classified Instances       16391               82.3544 %
Incorrectly Classified Instances      3512               17.6456 %
Kappa statistic                          0.4549
Mean absolute error                      0.2199
Root mean squared error                  0.3822
Relative absolute error                 63.2124 %
Root relative squared error             91.7387 %
Total Number of Instances            19903

=== Detailed Accuracy By Class ===

               TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
               0,914    0,489    0,866      0,914    0,889      0,459    0,760     0,873     0
               0,511    0,086    0,630      0,511    0,564      0,459    0,760     0,513     1
Weighted Avg.  0,824    0,399    0,814      0,824    0,817      0,459    0,760     0,793

=== Confusion Matrix ===

     a     b   <-- classified as
 14119  1336 |     a = 0
  2176  2272 |     b = 1
```

***Εικόνα 32**: Αποτελέσματα J48 (WEKA)*

Στον πίνακα σύγχυσης (confusion matrix) του κατηγοριοποιητή J48 στο WEKA παρατηρούμε ότι το μοντέλο πρόβλεψε 2272 αληθινές βροχερές μέρες (True Positive), 14119 αληθινές μη βροχερές μέρες (True Negative), 1336 ψευδείς θετικές βροχερές μέρες (False Positive) και 2176 ψευδώς αρνητικές μη βροχερές μέρες (False Negative).

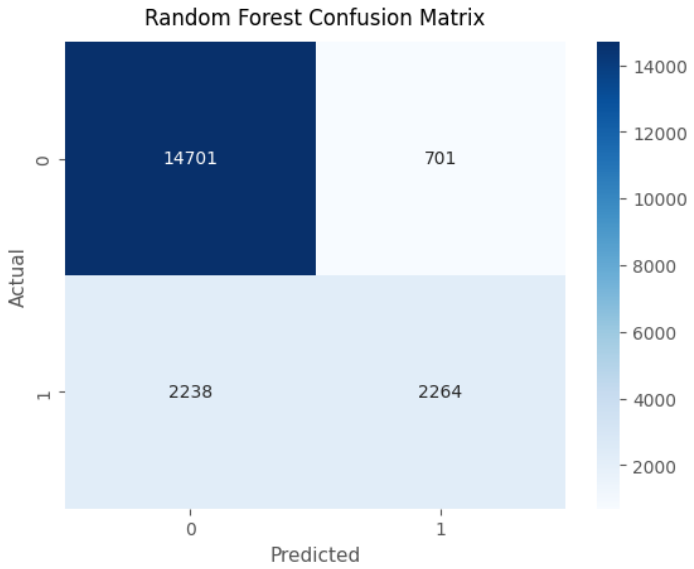


***Γραφική Παράσταση 3**: Συγκριτικό διάγραμμα Δέντρων Απόφασης*

Στο παραπάνω διάγραμμα παρατηρούμε ότι ο 3ος κατηγοριοποιητής DecisionTreesClassifier στο scikit-learn (κόκκινο χρώμα) έχει ορθότητα (accuracy) 0.84, έχει ακρίβεια (precision) 0.83, έχει ανάκληση (recall) 0.84 και έχει αρμονικό μέσο (f1 score) 0.83. Αντίστοιχα ο κατηγοριοποιητής Logistic στο WEKA (μπλέ χρώμα) έχει ορθότητα (accuracy) 0.823, έχει ακρίβεια (precision) 0.814, έχει ανάκληση (recall) 0.824 και έχει αρμονικό μέσο (f1 score) 0.817.

4.1.4 Random Forest

Παρακάτω παρουσιάζονται τα αποτελέσματα του 4ου κατηγοριοποιητή RandomForestClassifier που είναι υλοποιημένος στη βιβλιοθήκη scikit-learn.



***Εικόνα 33**: Πίνακας Σύγχυσης RandomForest (scikit-learn)*

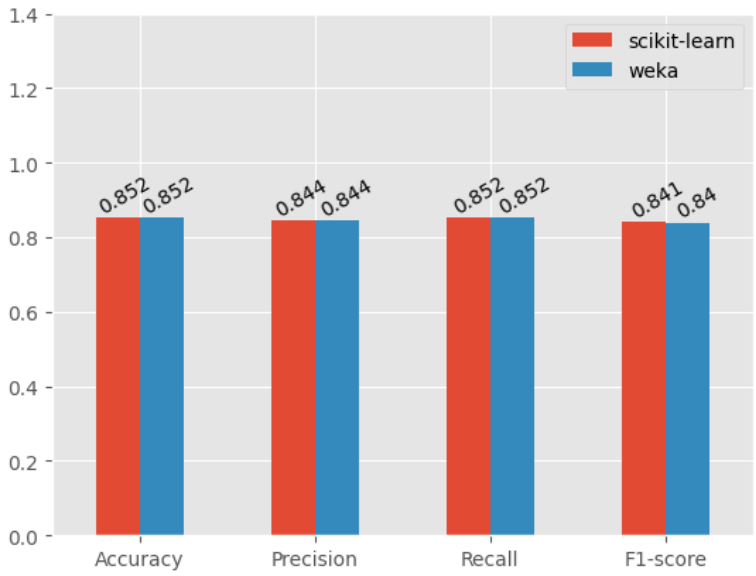
Στον παραπάνω πίνακα σύγχυσης (confusion matrix) του κατηγοριοποιητή RandomForestClassifier παρατηρούμε ότι το μοντέλο πρόβλεψε 2264 αληθινές βροχερές μέρες (True Positive), 14701 αληθινές μη βροχερές μέρες (True Negative), 701 ψευδείς θετικές βροχερές μέρες (False Positive) και 2238 ψευδώς αρνητικές μη βροχερές μέρες (False Negative).

Παρακάτω παρουσιάζονται τα αποτελέσματα του κατηγοριοποιητή Random Forest στο WEKA:

```
Correctly Classified Instances       16963               85.2284 %
Incorrectly Classified Instances      2940               14.7716 %
Kappa statistic                          0.5149
Mean absolute error                      0.2201
Root mean squared error                  0.3263
Relative absolute error                 63.251  %
Root relative squared error             78.3331 %
Total Number of Instances            19903

=== Detailed Accuracy By Class ===

               TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
               0,954    0,503    0,868      0,954    0,909      0,532    0,878     0,956     0
               0,497    0,046    0,759      0,497    0,601      0,532    0,878     0,720     1
Weighted Avg.  0,852    0,401    0,844      0,852    0,840      0,532    0,878     0,903

=== Confusion Matrix ===

     a     b   <-- classified as
 14751   704 |     a = 0
  2236  2212 |     b = 1
```

***Εικόνα 34**: Αποτελέσματα Random Forest (WEKA)*

Στον πίνακα σύγχυσης (confusion matrix) του κατηγοριοποιητή Random Forest στο WEKA παρατηρούμε ότι το μοντέλο πρόβλεψε 2212 αληθινές βροχερές μέρες (True Positive), 14751 αληθινές μη βροχερές μέρες (True Negative), 704 ψευδείς θετικές βροχερές μέρες (False Positive) και 2236 ψευδώς αρνητικές μη βροχερές μέρες (False Negative).

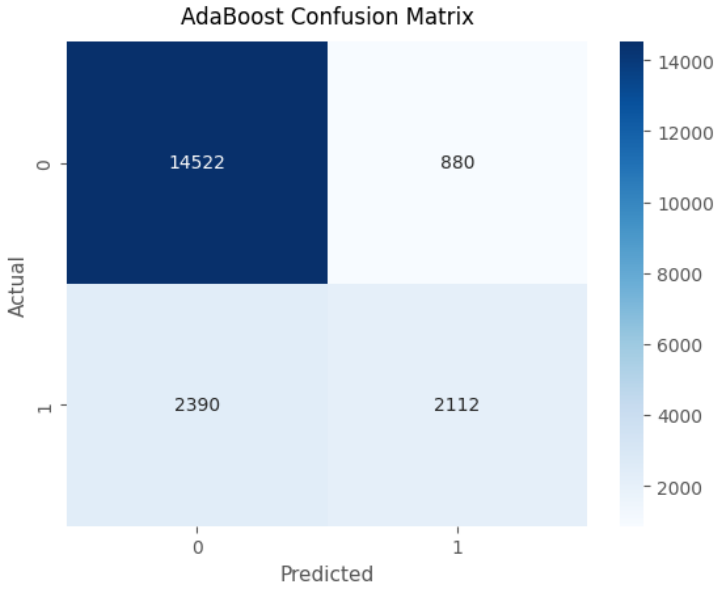


***Γραφική Παράσταση 4**: Συγκριτικό διάγραμμα αποτελεσμάτων Random Forest*

Στο παραπάνω διάγραμμα παρατηρούμε ότι ο κατηγοριοποιητής RandomForestClassifier στο scikit-learn (κόκκινο χρώμα) έχει ορθότητα (accuracy) 0.852, έχει ακρίβεια (precision) 0.844, έχει ανάκληση (recall) 0.852 και έχει αρμονικό μέσο (f1 score) 0.841. Αντίστοιχα ο κατηγοριοποιητής Random Forest στο WEKA (μπλέ χρώμα) έχει ορθότητα (accuracy) 0.852, έχει ακρίβεια (precision) 0.844, έχει ανάκληση (recall) 0.852 και έχει αρμονικό μέσο (f1 score) 0.84.

4.1.5 AdaBoost

Παρακάτω παρουσιάζονται τα αποτελέσματα του 5ου κατηγοριοποιητή AdaBoostClassifier που είναι υλοποιημένος στη βιβλιοθήκη scikit-learn.



***Εικόνα 35**: Πίνακας Σύγχυσης AdaBoost (scikit-learn)*

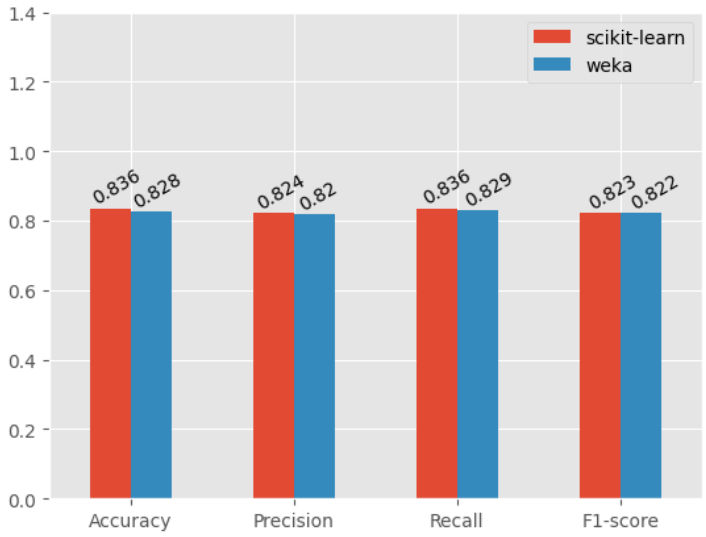
Στον παραπάνω πίνακα σύγχυσης (confusion matrix) του κατηγοριοποιητή AdaBoostClassifier παρατηρούμε ότι το μοντέλο πρόβλεψε 2112 αληθινές βροχερές μέρες (True Positive), 14522 αληθινές μη βροχερές μέρες (True Negative), 880 ψευδείς θετικές βροχερές μέρες (False Positive) και 2390 ψευδώς αρνητικές μη βροχερές μέρες (False Negative).

Παρακάτω παρουσιάζονται τα αποτελέσματα του κατηγοριοποιητή AdaBoostM1 στο WEKA:

```
=== Summary ===

Correctly Classified Instances       16496               82.882  %
Incorrectly Classified Instances      3407               17.118  %
Kappa statistic                          0.4724
Mean absolute error                      0.1704
Root mean squared error                  0.3976
Relative absolute error                 48.984  %
Root relative squared error             95.4432 %
Total Number of Instances            19903

=== Detailed Accuracy By Class ===

               TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
               0,916    0,475    0,870      0,916    0,893      0,476    0,837     0,936     0
               0,525    0,084    0,643      0,525    0,578      0,476    0,837     0,653     1
Weighted Avg.  0,829    0,387    0,820      0,829    0,822      0,476    0,837     0,873

=== Confusion Matrix ===

     a     b   <-- classified as
 14159  1296 |     a = 0
  2111  2337 |     b = 1
```

***Εικόνα 36**: Αποτελέσματα AdaBoost (WEKA)*

Στον πίνακα σύγχυσης (confusion matrix) του κατηγοριοποιητή AdaBoostM1 στο WEKA παρατηρούμε ότι το μοντέλο πρόβλεψε 2337 αληθινές βροχερές μέρες (True Positive), 14159 αληθινές μη βροχερές μέρες (True Negative), 1296 ψευδείς θετικές βροχερές μέρες (False Positive) και 2111 ψευδώς αρνητικές μη βροχερές μέρες (False Negative).

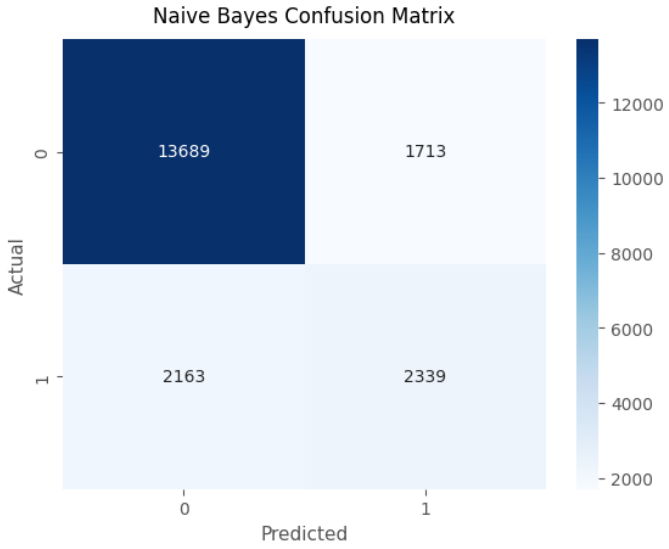


***Γραφική Παράσταση 5**: Συγκριτικό διάγραμμα αποτελεσμάτων AdaBoost*

Στο παραπάνω διάγραμμα παρατηρούμε ότι ο κατηγοριοποιητής AdaBoostClassifier στο scikit-learn (κόκκινο χρώμα) έχει ορθότητα (accuracy) 0.836, έχει ακρίβεια (precision) 0.824, έχει ανάκληση (recall) 0.836 και έχει αρμονικό μέσο (f1 score) 0.823. Αντίστοιχα ο κατηγοριοποιητής AdaBoostM1 στο WEKA (μπλέ χρώμα) έχει ορθότητα (accuracy) 0.828, έχει ακρίβεια (precision) 0.82, έχει ανάκληση (recall) 0.829 και έχει αρμονικό μέσο (f1 score) 0.822.

4.1.6 Naïve Bayes

Παρακάτω παρουσιάζονται τα αποτελέσματα του 6ου κατηγοριοποιητή Naïve Bayes που είναι υλοποιημένος στη βιβλιοθήκη scikit-learn.



***Εικόνα 37**: Πίνακας Σύγχυσης GaussianNB (scikit-learn)*

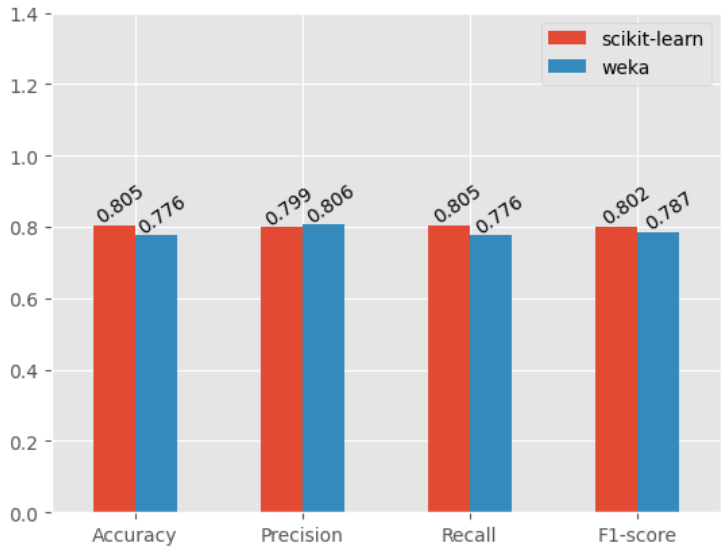
Στον παραπάνω πίνακα σύγχυσης (confusion matrix) του 6ου κατηγοριοποιητή GaussianNB παρατηρούμε ότι το μοντέλο πρόβλεψε 2339 αληθινές βροχερές μέρες (True Positive), 13689 αληθινές μη βροχερές μέρες (True Negative), 1713 ψευδείς θετικές βροχερές μέρες (False Positive) και 2163 ψευδώς αρνητικές μη βροχερές μέρες (False Negative).

Παρακάτω παρουσιάζονται τα αποτελέσματα του 6ου κατηγοριοποιητή NaiveBayes στο WEKA:

```
Correctly Classified Instances       15452               77.6365 %
Incorrectly Classified Instances      4451               22.3635 %
Kappa statistic                          0.4237
Mean absolute error                      0.26
Root mean squared error                  0.413
Relative absolute error                 74.7414 %
Root relative squared error             99.1324 %
Total Number of Instances            19903

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure  MCC      ROC Area  PRC Area  Class
               0,808     0,335     0,894       0,808   0,849      0,432    0,812     0,929     0
               0,665     0,192     0,500       0,665   0,571      0,432    0,812     0,562     1
Weighted Avg.  0,776     0,303     0,806       0,776   0,787      0,432    0,812     0,847

=== Confusion Matrix ===

     a     b   <-- classified as
 12492  2963 |     a = 0
  1488  2960 |     b = 1
```

***Εικόνα 38**: Αποτελέσματα Naïve Bayes (WEKA)*

Στον πίνακα σύγχυσης (confusion matrix) του 6ου κατηγοριοποιητή Naïve Bayes στο WEKA παρατηρούμε ότι το μοντέλο πρόβλεψε 2960 αληθινές βροχερές μέρες (True Positive), 12492 αληθινές μη βροχερές μέρες (True Negative), 2963 ψευδείς θετικές βροχερές μέρες (False Positive) και 1488 ψευδώς αρνητικές μη βροχερές μέρες (False Negative).

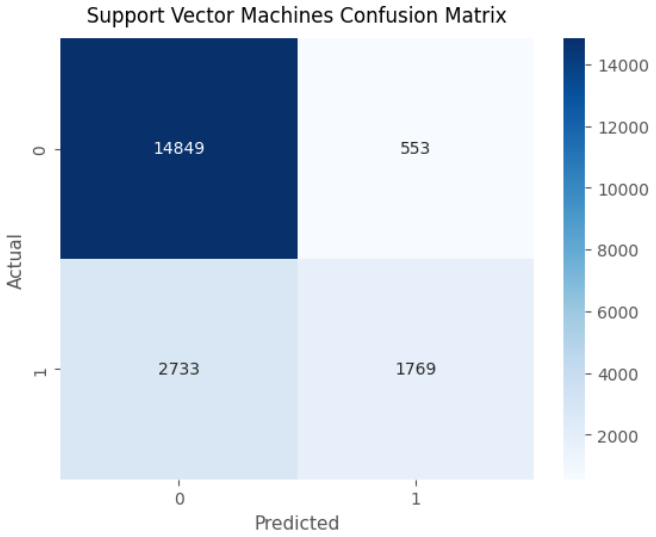


***Γραφική Παράσταση 6**: Συγκριτικό διάγραμμα αποτελεσμάτων Naïve Bayes*

Στο παραπάνω διάγραμμα παρατηρούμε ότι ο 6ος κατηγοριοποιητής GaussianNB στο scikit-learn (κόκκινο χρώμα) έχει ορθότητα (accuracy) 0.805, έχει ακρίβεια (precision) 0.799, έχει ανάκληση (recall) 0.805 και έχει αρμονικό μέσο (f1 score) 0.802. Αντίστοιχα ο κατηγοριοποιητής NaiveBayes στο WEKA (μπλέ χρώμα) έχει ορθότητα (accuracy) 0.776, έχει ακρίβεια (precision) 0.806, έχει ανάκληση (recall) 0.776 και έχει αρμονικό μέσο (f1 score) 0.787.

4.1.7 Support Vector Machines

Παρακάτω παρουσιάζονται τα αποτελέσματα του 7ου κατηγοριοποιητή SVC που είναι υλοποιημένος στη βιβλιοθήκη scikit-learn.



Εικόνα 39: Πίνακας Σύγχυσης SVMs

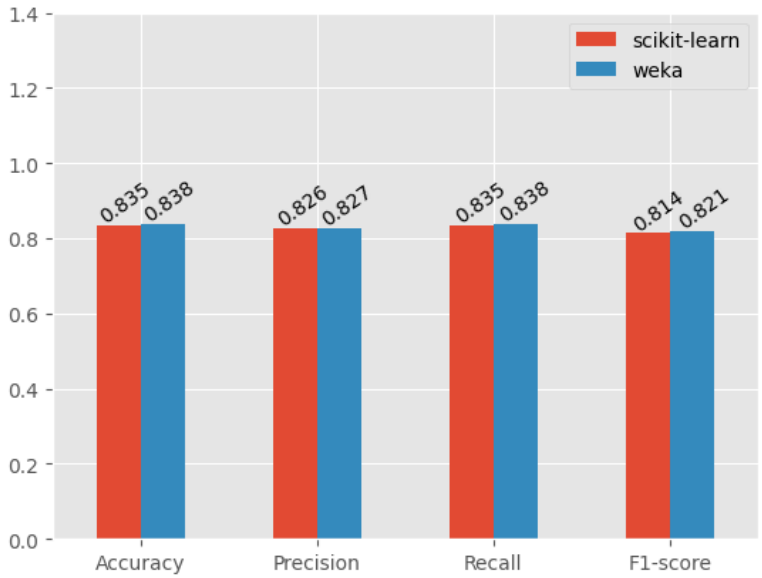
Στον παραπάνω πίνακα σύγχυσης (confusion matrix) του 7ου κατηγοριοποιητή SVC παρατηρούμε ότι το μοντέλο πρόβλεψε 1769 αληθινές βροχερές μέρες (True Positive), 14849 αληθινές μη βροχερές μέρες (True Negative), 553 ψευδείς θετικές βροχερές μέρες (False Positive) και 2733 ψευδώς αρνητικές μη βροχερές μέρες (False Negative).

Παρακάτω παρουσιάζονται τα αποτελέσματα του 7ου κατηγοριοποιητή SMO στο WEKA:

```
Correctly Classified Instances       16683               83.8215 %
Incorrectly Classified Instances      3220               16.1785 %
Kappa statistic                          0.4519
Mean absolute error                      0.1618
Root mean squared error                  0.4022
Relative absolute error                 46.5023 %
Root relative squared error             96.5534 %
Total Number of Instances            19903

=== Detailed Accuracy By Class ===

               TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
               0,956    0,572    0,853      0,956    0,902      0,477    0,692     0,850     0
               0,428    0,044    0,738      0,428    0,542      0,477    0,692     0,444     1
Weighted Avg.  0,838    0,454    0,827      0,838    0,821      0,477    0,692     0,759

=== Confusion Matrix ===

     a     b   <-- classified as
 14779   676 |     a = 0
  2544  1904 |     b = 1
```

Εικόνα 40: Αποτελέσματα SVMs

Στον πίνακα σύγχυσης (confusion matrix) του 7ου κατηγοριοποιητή SMO στο WEKA παρατηρούμε ότι το μοντέλο πρόβλεψε 1904 αληθινές βροχερές μέρες (True Positive), 14779 αληθινές μη βροχερές μέρες (True Negative), 676 ψευδείς θετικές βροχερές μέρες (False Positive) και 2544 ψευδώς αρνητικές μη βροχερές μέρες (False Negative).

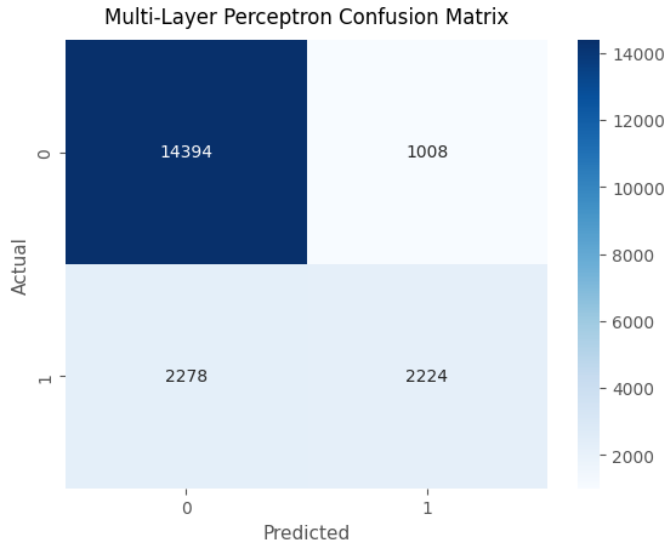


***Γραφική Παράσταση** 7: Συγκριτικό διάγραμμα αποτελεσμάτων SVMs*

Στο παραπάνω διάγραμμα παρατηρούμε ότι ο 7ος κατηγοριοποιητής SVC στο scikit-learn (κόκκινο χρώμα) έχει ορθότητα (accuracy) 0.835, έχει ακρίβεια (precision) 0.826, έχει ανάκληση (recall) 0.835 και έχει αρμονικό μέσο (f1 score) 0.814. Αντίστοιχα ο κατηγοριοποιητής SMO στο WEKA (μπλέ χρώμα) έχει ορθότητα (accuracy) 0.838, έχει ακρίβεια (precision) 0.827, έχει ανάκληση (recall) 0.838 και έχει αρμονικό μέσο (f1 score) 0.821.

4.1.8 Multi – Layer Perceptron

Παρακάτω παρουσιάζονται τα αποτελέσματα του 8ου κατηγοριοποιητή MLPClassifier που είναι υλοποιημένος στη βιβλιοθήκη scikit-learn.



***Εικόνα 41**: Πίνακας Σύγχυσης MLP (scikit-learn)*

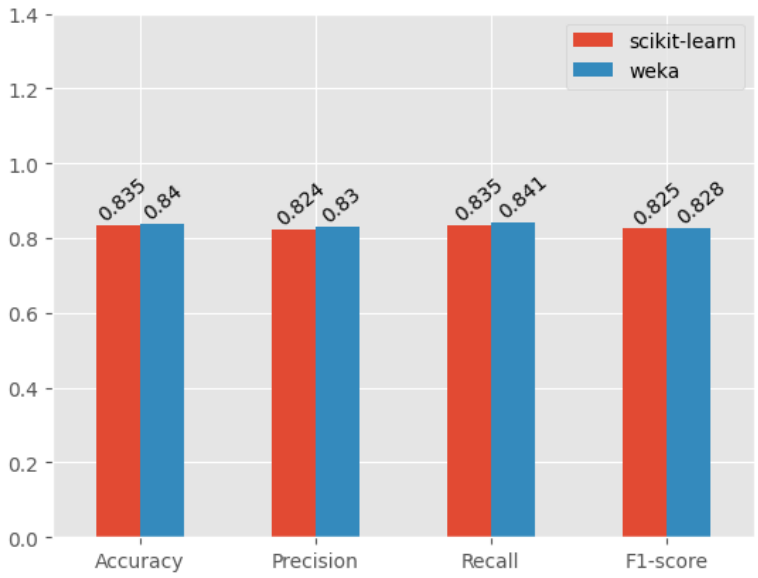
Στον παραπάνω πίνακα σύγχυσης (confusion matrix) του κατηγοριοποιητή MLPClassifier παρατηρούμε ότι το μοντέλο πρόβλεψε 2224 αληθινές βροχερές μέρες (True Positive), 14394 αληθινές μη βροχερές μέρες (True Negative), 1008 ψευδείς θετικές βροχερές μέρες (False Positive) και 2278 ψευδώς αρνητικές μη βροχερές μέρες (False Negative).

Παρακάτω παρουσιάζονται τα αποτελέσματα του κατηγοριοποιητή MLP στο WEKA:

```
Correctly Classified Instances       16730               84.0577 %
Incorrectly Classified Instances      3173               15.9423 %
Kappa statistic                          0.4794
Mean absolute error                      0.2313
Root mean squared error                  0.3387
Relative absolute error                 66.4807 %
Root relative squared error             81.3097 %
Total Number of Instances            19903

=== Detailed Accuracy By Class ===

               TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
               0,945    0,522    0,863      0,945    0,902      0,494    0,855     0,947     0
               0,478    0,055    0,714      0,478    0,573      0,494    0,855     0,682     1
Weighted Avg.  0,841    0,418    0,830      0,841    0,828      0,494    0,855     0,888

=== Confusion Matrix ===

     a     b   <-- classified as
 14604   851 |     a = 0
  2322  2126 |     b = 1
```

***Εικόνα 42**: Αποτελέσματα MultiLayerPerceptron (WEKA)*

Στον παραπάνω πίνακα σύγχυσης (confusion matrix) του κατηγοριοποιητή MLP παρατηρούμε ότι το μοντέλο πρόβλεψε 2126 αληθινές βροχερές μέρες (True Positive), 14604 αληθινές μη βροχερές μέρες (True Negative), 851 ψευδείς θετικές βροχερές μέρες (False Positive) και 2322 ψευδώς αρνητικές μη βροχερές μέρες (False Negative).



***Γραφική Παράσταση 8**: Συγκριτικό διάγραμμα αποτελεσμάτων MLP*

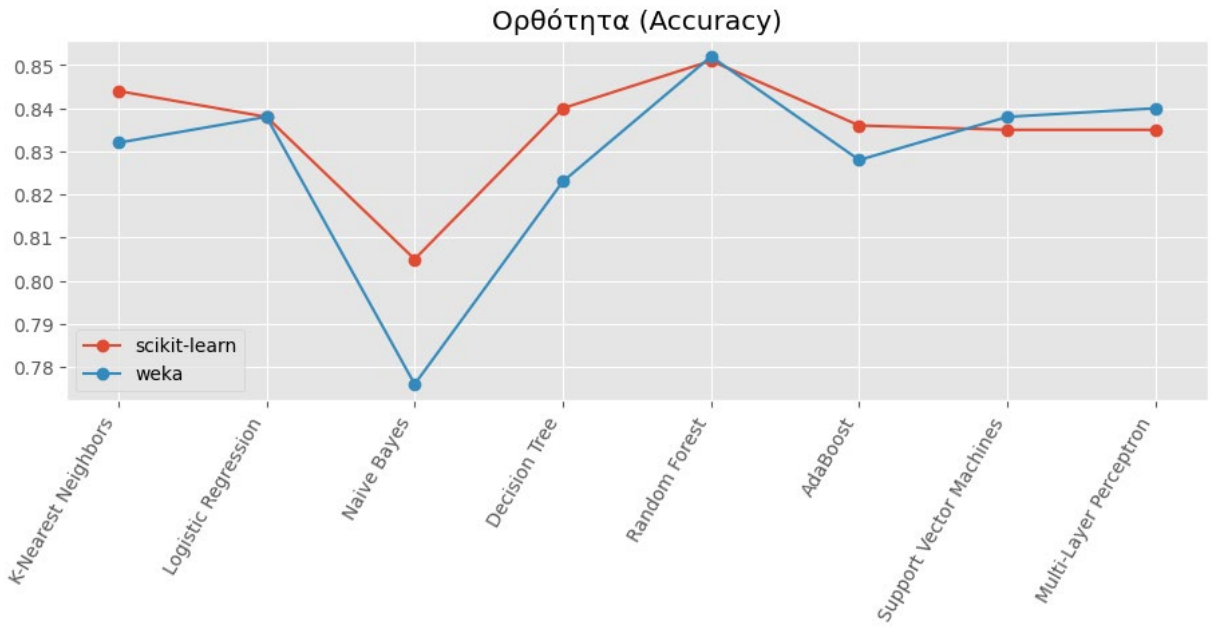
Στο παραπάνω διάγραμμα παρατηρούμε ότι ο κατηγοριοποιητής MLP Classifier στο scikit-learn (κόκκινο χρώμα) έχει ορθότητα (accuracy) 0.835, έχει ακρίβεια (precision) 0.824, έχει ανάκληση (recall) 0.835 και έχει αρμονικό μέσο (f1 score) 0.825. Αντίστοιχα ο κατηγοριοποιητής MLP στο WEKA (μπλέ χρώμα) έχει ορθότητα (accuracy) 0.84, έχει ακρίβεια (precision) 0.83, έχει ανάκληση (recall) 0.841 και έχει αρμονικό μέσο (f1 score) 0.828.

4.2 Συγκριτική αξιολόγηση αλγόριθμων μηχανικής μάθησης

Σε αυτή την ενότητα θα πραγματοποιηθεί η συγκριτική αξιολόγηση των κατηγοριοποιητών που εφαρμόστηκαν επί του συνόλου δεδομένων, που αποτελεί και το τελικό στάδιο (Evaluation) της διαδικασίας ανακάλυψης γνώσης. Η σύγκριση αυτή περιλαμβάνει την αξιολόγηση μετρικών όπως η ορθότητα (accuracy), η ακρίβεια (precision), η ανάκληση (recall) και το F1-score για κάθε κατηγοριοποιητή. Μετα το πέρας της σύγκρισης όλων των αποτελεσμάτων θα επιλεγεί ο βέλτιστος κατηγοριοποιητής.

4.2.1 Συγκριτική αξιολόγηση ορθότητας

Στο παρακάτω διάγραμμα παρουσιάζεται η ορθότητα όλων των ταξινομητών (classifiers):

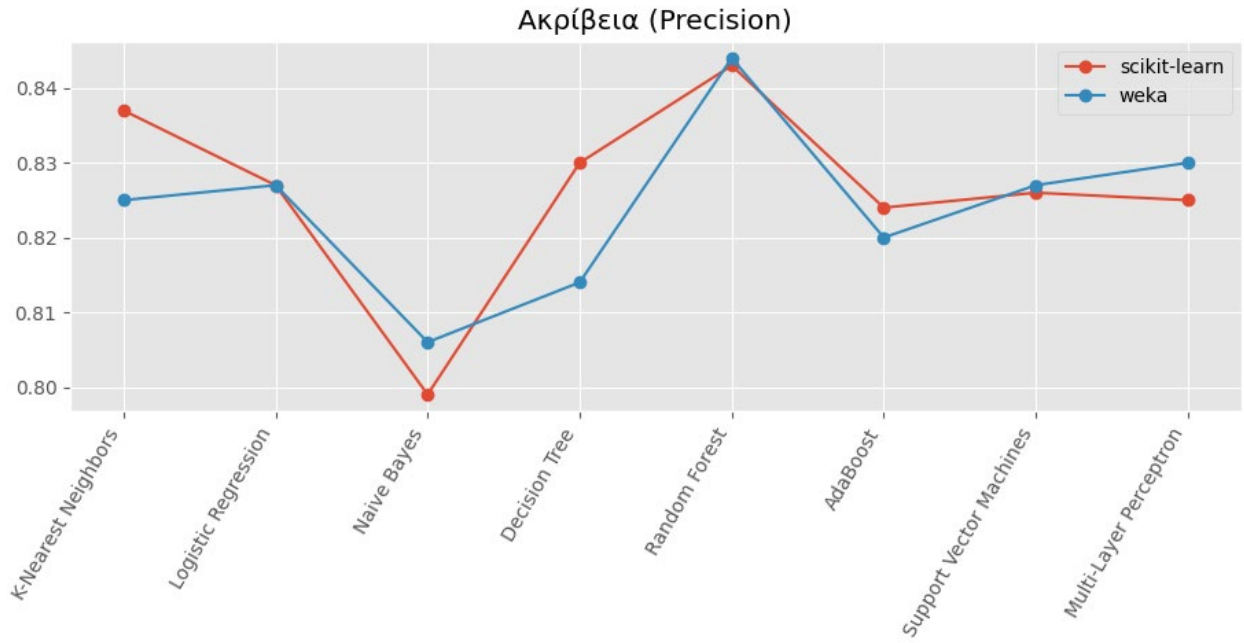


***Γραφική Παράσταση 9**: Συγκριτικό διάγραμμα αξιολόγησης ορθότητας*

Από το διάγραμμα παρατηρούμε ότι όλοι οι ταξινομητές (classifiers) έχουν σχετικά υψηλή ορθότητα (accuracy). Ο κατηγοριοποιητής RandomForest έχει την υψηλότερη ορθότητα (accuracy) σε σχέση με τους υπόλοιπους τόσο στο Scikit-learn όσο και στο λογισμικό WEKA, γεγονός που οφείλεται στις λιγότερες συνολικές ψευδείς προβλέψεις σε σύγκριση με τους υπόλοιπους. Υπενθυμίζεται ότι στον τύπο της ορθότητας (εξίσωση 15) ο συνολικός αριθμός προβλέψεων διαιρείται με τις συνολικές ψευδείς προβλέψεις, επομένως όσο πιο χαμηλός είναι αυτός ο αριθμός τόσο πιο υψηλή θα είναι η ορθότητα.

4.2.2 Συγκριτική αξιολόγηση ακρίβειας

Στο παρακάτω διάγραμμα παρουσιάζεται η ακρίβεια όλων των ταξινομητών (classifiers):

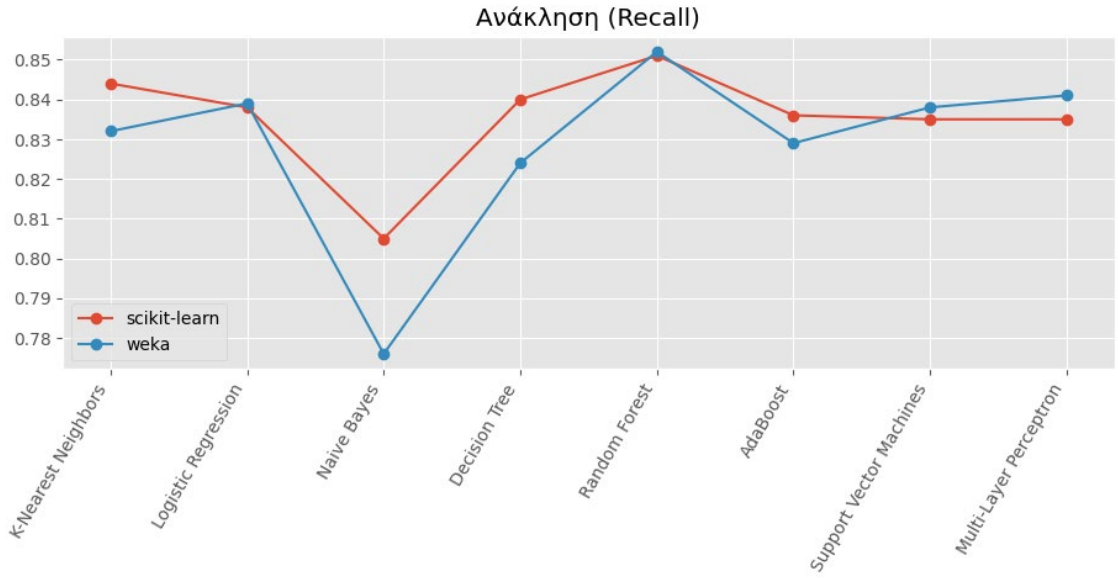


***Γραφική Παράσταση 10**: Συγκριτικό διάγραμμα αξιολόγησης ακρίβειας*

Από το διάγραμμα παρατηρούμε ότι όλοι οι ταξινομητές (classifiers) έχουν σχετικά υψηλή ακρίβεια (precision). Ο κατηγοριοποιητής RandomForest έχει την υψηλότερη ορθότητα (accuracy) σε σχέση με τους υπόλοιπους τόσο στο Scikit-learn όσο και στο λογισμικό WEKA, γεγονός που οφείλεται στις λιγότερες συνολικές ψευδείς θετικές προβλέψεις σε σύγκριση με τους υπόλοιπους. Υπενθυμίζεται ότι από τον τύπο της ακρίβειας (εξίσωση 16) όσο μικρότερος είναι ο αριθμός των ψευδών θετικών προβλέψεων (που είναι στο παρονομαστή του κλάσματος της εξίσωσης) τόσο πιο υψηλή θα είναι η ακρίβεια.

4.2.3 Συγκριτική αξιολόγηση ανάκλησης

Στο παρακάτω διάγραμμα παρουσιάζεται η ανάκληση όλων των ταξινομητών (classifiers):

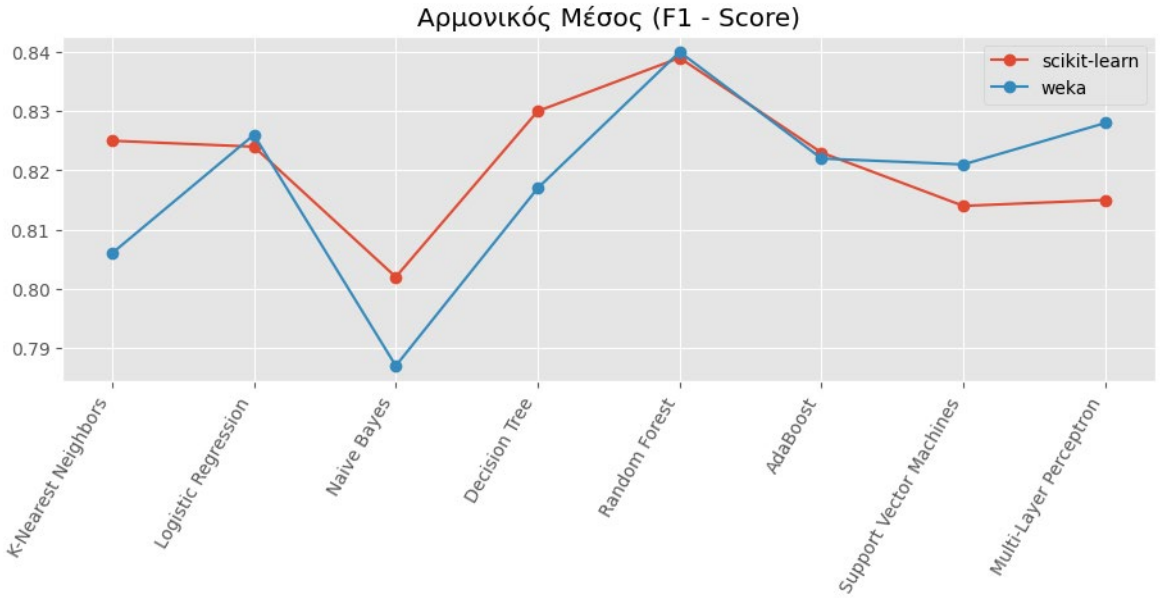


***Γραφική Παράσταση 11**: Συγκριτικό διάγραμμα αξιολόγησης ανάκλησης*

Από το διάγραμμα παρατηρούμε ότι όλοι οι ταξινομητές (classifiers) έχουν σχετικά υψηλή ανάκληση (recall). Ο κατηγοριοποιητής RandomForest έχει την υψηλότερη ανάκληση (recall) σε σχέση με τους υπόλοιπους τόσο στο Scikit-learn όσο και στο λογισμικό WEKA, γεγονός που οφείλεται στις λιγότερες συνολικές ψευδείς αρνητικές προβλέψεις σε σύγκριση με τους υπόλοιπους. Υπενθυμίζεται ότι από τον τύπο της ανάκλησης (εξίσωση 17) όσο μικρότερος είναι ο αριθμός των ψευδών αρνητικών προβλέψεων (που είναι στο παρονομαστή του κλάσματος της εξίσωσης) τόσο πιο υψηλή θα είναι η ανάκληση.

4.2.4 Συγκριτική αξιολόγηση αρμονικού μέσου

Στο παρακάτω διάγραμμα παρουσιάζεται η ο Αρμονικός Μέσος όλων των ταξινομητών (classifiers):



***Γραφική Παράσταση 12**: Συγκριτικό διάγραμμα αξιολόγησης αρμονικού μέσου*

Στο παραπάνω διάγραμμα παρατηρούμε πως όλοι οι ταξινομητές (classifiers) έχουν υψηλό αρμονικό μέσο (f1 score). Ο κατηγοριοποιητής RandomForest έχει την υψηλότερη ανάκληση (recall) σε σχέση με τους υπόλοιπους τόσο στο Scikit-learn όσο και στο λογισμικό WEKA γεγονός που οφείλεται στο ότι ο τύπος του αρμονικού μέσου χρησιμοποιεί την ανάκληση και την ακρίβεια, μετρικές που ο κατηγοριοποιητής όπως είδαμε πέτυχε τα υψηλότερα ποσοστά.

4.3 Επιλογή βέλτιστου αλγόριθμου μηχανικής μάθησης

Όλοι οι κατηγοριοποιητές/ταξινομητές (classifiers) έχουν υψηλά αποτελέσματα όπως παρατηρείται στο κεφάλαιο και έχουν μικρές διαφορές μεταξύ τους. Ακόμη παρατηρούμε πως σε γενικές γραμμές τα αποτελέσματα της ταξινόμησης στο WEKA και στο scikit-learn είναι παραπλήσια σχεδόν σε όλους τους κατηγοριοποιητές, με μόνη εξαίρεση τον κατηγοριοποιητή Naïve Bayes, στον οποίο η παραμετροποίηση κρίθηκε περιορισμένη, γι' αυτό και χρησιμοποιήθηκε και με τις προεπιλεγμένες παραμέτρους. Και στις τέσσερις μετρικές (Ορθότητα, Ακρίβεια, Ανάκληση και Αρμονικό Μέσο) ο κατηγοριοποιητής Random Forest πέτυχε τα υψηλότερα αποτελέσματα τόσο στο scikit-learn όσο και στο WEKA. Συγκεκριμένα τα αποτελέσματα που αλγορίθμου Random Forest συνοψίζονται στο παρακάτω πίνακα:

***Πίνακας 29**:Συγκριτικά Αποτελέσματα μετρικών Random Forest σε Scikit-learn και WEKA*

Random Forest	Scikit-learn	WEKA
Ορθότητα (Accuracy)	0.852	0.852
Ακρίβεια (Precision)	0.844	0.844

Ανάκληση (Recall)	0.852	0.852
Αρμονικός Μέσος (F1-Score)	0.841	0.84

Όπως βλέπουμε και στα δυο περιβάλλοντα τα αποτελέσματα όλων των μετρικών είναι αρκετά παραπλήσια (ή ακόμα και ίδια). Τέλος, παρουσιάζουμε τις προβλέψεις από τον πίνακα σύγχυσης του κατηγοριοποιητή:

***Πίνακας 30**: Αποτελέσματα Πίνακα Σύγχυσης Random Forest σε Scikit-learn και WEKA*

Random Forest	**Scikit-learn**	**WEKA**
True Positive (TP)	2264	2212
False Positive (FP)	701	704
True Negative (TN)	14701	14751
False Negative (FN)	2238	2236

Κεφάλαιο 5: Συμπεράσματα και προτάσεις για μελλοντικές κατευθύνσεις

Σε αυτό το τελευταίο κεφάλαιο θα παρουσιαστούν τα συμπεράσματα της παρούσας διπλωματικής εργασίας καθώς και προτάσεις για μελλοντικές κατευθύνσεις. Αρχικά, θα δοθεί μια σύντομη ανασκόπηση του περιεχομένου της διπλωματικής εργασίας και θα παρουσιαστούν τα κύρια συμπεράσματα που προκύπτουν από τη μελέτη. Τέλος, θα δοθούν ορισμένες προτάσεις για το πώς μπορεί να εξελιχθεί η μελέτη πρόγνωσης καιρικών συνθήκων με χρήση τεχνικών μηχανικής μάθησης στο μέλλον.

5.1 Συμπεράσματα

Στην παρούσα διπλωματική εργασία ακολουθήθηκαν τα βήματα της διαδικασίας ανακάλυψης γνώσης από βάσεις δεδομένων με σκοπό τη σύγκριση διαφόρων αλγορίθμων μηχανικής μάθησης για πρόβλεψη καιρικών συνθηκών και συγκεκριμένα για την πρόβλεψη βροχόπτωσης. Η μηχανική μάθηση αντιπροσωπεύει ένα αναπόσπαστο κομμάτι της τεχνητής νοημοσύνης που μπορεί να χρησιμοποιηθεί για προβλήματα κατηγοριοποίησης. Με την άνοδο της τεχνολογίας τα τελευταία χρόνια, η μηχανική μάθηση χρησιμοποιείται όλο και περισσότερο και αναμένεται να παρέχει λύσεις για πολλά προβλήματα στο μέλλον [32-48]. Στην καθημερινή ζωή, οι ανθρώπινες δραστηριότητες, εξαρτώνται σε μεγάλο βαθμό από τις καιρικές συνθήκες, καθώς σωστές προβλέψεις βοηθούν τους ανθρώπους να προετοιμαστούν και να προσαρμοστούν ανάλογα. Ακόμη η πρόβλεψη του καιρού είναι καίρια για την αντιμετώπιση καιρικών φαινόμενων όπως καταιγίδες, πλημμύρες που μπορούν να έχουν καταστροφικές επιπτώσεις στο περιβάλλον ιδιαίτερα κατά την περίοδο και κλιματικής κρίσης που διανύουμε. Για τη πρόβλεψη της βροχόπτωσης χρησιμοποιήθηκαν μετεωρολογικά δεδομένα από διάφορες πόλεις της Αυστραλίας κατά το έτος 2010, χρονιά που συμπίπτει με εκδήλωση του φαινομένου La Lina που μπορεί να προκαλέσει σφοδρές βροχοπτώσεις και πλημμύρες. Η ποιότητα των μετεωρολογικών οργάνων παίζει πολύ σημαντικό ρόλο στην διαθεσιμότητα αξιόπιστων δεδομένων στα οποία μπορούμε να βασίσουμε την διαδικασία της εκπαίδευσης κάθε μοντέλου. Πρέπει επομένως τα δεδομένα μας να προέρχονται από σύγχρονους μετεωρολογικούς σταθμούς που ελέγχονται και συντηρούνται διαρκώς από εξειδικευμένο προσωπικό. Ακολουθώντας τη διαδικασία ανακάλυψης γνώσης από βάσεις δεδομένων, στο στάδιο της εξόρυξης δεδομένων με την κατηγοριοποίηση που εφαρμόστηκε οι κατηγοριοποιητές πρόβλεψαν την πιθανότητα βροχόπτωσης και στη συνέχεια αξιολογήθηκαν. Με αυτό το τρόπο, όπως μπορεί να συνειδητοποιήσει κανείς η μηχανική μάθηση μπορεί να δώσει λύσεις σε προβλήματα του πραγματικού κόσμου, διευκολύνοντας τις ζωές των ανθρώπων.

5.2 Προτάσεις για μελλοντικές κατευθύνσεις

Για την πρόβλεψη και την αξιολόγηση των αποτελεσμάτων της παρούσας διπλωματικής εργασίας, ο διαχωρισμός των δεδομένων σε σύνολο ελέγχου και σύνολο εκπαίδευσης έγινε με την μέθοδο test_train_split της βιβλιοθήκης scikit-learn και με το Option Percentage Split στο WEKA. Κατ' αυτόν τον τρόπο υπενθυμίζουμε ότι χρησιμοποιήσαμε το 80% του συνόλου δεδομένων για εκπαίδευση και το υπόλοιπο 20% για έλεγχο. Μια διαφορετική τεχνική που θα μπορούσε να χρησιμοποιηθεί είναι η k-Fold Cross-Validation. Κύρια ιδέα της τεχνικής αυτής αποτελεί ο διαχωρισμός των δεδομένων μας σε k "διασταυρούμενα" fold, όπου κάθε fold θα χρησιμοποιηθεί τόσο για την εκπαίδευση όσο και για τον έλεγχο του μοντέλου. Η διαδικασία αυτή πρέπει να επαναληφθεί k φορές, όπου κάθε φορά ένα από τα μπλοκ θα χρησιμοποιείται ως fold ελέγχου, ενώ τα υπόλοιπα μπλοκ θα χρησιμοποιούνται ως fold εκπαίδευσης. Ακόμη, μια άλλη προέκταση της μελέτης θα περιλάμβανε την συγκριτική μελέτη περισσότερων αλγορίθμων μηχανικής μάθησης (όπως για παράδειγμα οι XGBoost και GradientTreeBoosting) που είναι υλοποιημένοι στο scikit-learn αλλά δεν υπάρχουν στο WEKA, ή και το ανάποδο. Τέλος θα ήταν ενδιαφέρον, να δινόταν η δυνατότητα τα μετεωρολογικά δεδομένα να αντλούνταν από κάποιο API σε πραγματικό χρόνο έτσι ώστε με χρήση μηχανικής μάθησης να προβλέψουμε τη βροχόπτωσης ανά περιοχή για μια δεδομένη στιγμή. Κάτι τέτοιο όμως θα δημιουργούσε δυο ζητήματα. Αρχικά, ενδεχομένως να έπρεπε να καταφύγουμε στην επί-πληρωμή χρήση κάποιου API από το οποίο θα αντλούσαμε τα δεδομένα, καθώς όσα ήταν ελεύθερα προς χρήση παρατηρήθηκε ότι παρουσίαζαν τεράστιο αριθμό ελλείπων τιμών. Ακόμη στην εκδοχή αυτή, χαρακτηριστική θα είναι η απουσία της μεταβλητής στόχου για την βροχόπτωση, γεγονός που θα μας έκανε να πρέπει να εξετάσουμε με μεγάλη προσοχή το γνώρισμα στόχο βάσει του οποίου πρέπει να γίνει η κατηγοριοποίηση.

Βιβλιογραφία

1. Kontellis, E., Troussas, C., Krouska, A., & Sgouropoulou, C. (2021). Real-time face mask detector using convolutional neural networks amidst COVID-19 pandemic. In Novelties in Intelligent Digital Systems: Proceedings of the 1st International Conference (NIDS 2021), Athens, Greece, September 30-October 1, 2021 (Vol. 338, p. 247-255). IOS Press. doi:10.3233/FAIA210102.
2. Christos Troussas, Akrivi Krouska και Maria Virvou. «A multicriteria framework for assessing sentiment analysis in social and digital learning: software review». Στο: 2018 9th International Conference on Information, Intelligence, Systems and Applications (IISA). IEEE. 2018, σσ. 1–7.
3. Christos Troussas, Akrivi Krouska και Maria Virvou. «A multicriteria framework for assessing sentiment analysis in social and digital learning: Software».
4. Christos Troussas, Akrivi Krouska και Maria Virvou. «Evaluation of ensemblebased sentiment classifiers for Twitter data». Στο: 2016 7th International Conference on Information, Intelligence, Systems Applications (IISA). 2016, σσ. 1–6. DOI: 10.1109/IISA.2016.7785380.
5. Christos Troussas, Maria Virvou, Kurt Junshean Espinosa, Kevin Llaguno και Jaime Caro. «Sentiment analysis of Facebook statuses using Naive Bayes classifier for language learning». Στο: IISA 2013. 2013, σσ. 1–6. DOI: 10.1109/IISA.2013. 6623713.
6. Christos Troussas, Akrivi Krouska και Maria Virvou. «Trends on sentiment analysis over social networks: pre-processing ramifications, stand-alone classifiers and ensemble averaging». Στο: Machine Learning Paradigms. Springer, 2019, σσ. 161–186.
7. Christos Troussas, Akrivi Krouska, Cleo Sgouropoulou: *Collaboration and fuzzy-modeled personalization for mobile game-based learning in higher education*, Computers & Education, Volume 144, 2020, 103698, https://doi.org/10.1016/j.compedu.2019.103698.
8. Kanetaki, Z., Stergiou, C., Bekas, G., Troussas, C., & Sgouropoulou, C. (2021). Data Mining for Improving Online Higher Education Amidst COVID-19 Pandemic: A Case Study in the Assessment of Engineering Students. Novelties in Intelligent Digital Systems: Proceedings of the 1st International Conference (NIDS 2021), Athens, Greece, September 30-October 1, 2021 (Vol. 338, p. 157-165). doi:10.3233/FAIA210088.
9. Kapetanaki, A., Krouska, A., Troussas, C., & Sgouropoulou, C. (2021). A Novel Framework Incorporating Augmented Reality and Pedagogy for Improving Reading Comprehension in Special Education. In Novelties in Intelligent Digital Systems: Proceedings of the 1st International Conference (NIDS 2021), Athens, Greece, September 30-October 1, 2021 (Vol. 338, p. 105-110). IOS Press. doi:10.3233/FAIA210081.
10. Kuleshov, Y.; Qi, L.; Fawcett, R.; Jones, D. (2008). «On tropical cyclone activity in the Southern Hemisphere: Trends and the ENSO connection» (στα αγγλικά). Geophysical Research Letters 35 (14): S08. doi:10.1029/2007GL032983. ISSN 1944-8007.
11. Record-breaking La Niña events: An analysis of the La Niña life cycle and the impacts and significance of the 2010–11 and 2011–12 La Niña events in Australia. Available at: http://www.bom.gov.au/climate/enso/history/La-Nina-2010-12.pdf
12. Mitchell, T. M. (1997). Machine Learning. McGraw-Hill.
13. Bishop, C. M. (2006). Pattern Recognition and Machine Learning. Springer.
14. Fayyad, U., Piatetsky-Shapiro, G., & Smyth, P. (1996). From Data Mining to Knowledge Discovery in Databases. AI Magazine, 17(3), 37-54.
15. Cox, D. R. (1958). The regression analysis of binary sequences. Journal of the Royal Statistical Society, Series B (Methodological), 20(2), 215-242.

16. Quinlan, J. R. (1986). Induction of decision trees. Machine Learning, 1(1), 81-106.
17. Quinlan, J. R. (1993). C4.5: Programs for machine learning. San Francisco, CA: Morgan Kaufmann Publishers.
18. Breiman, L. (2001). Random forests. Machine Learning, 45(1), 5-32.
19. Freund, Y., & Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. Journal of computer and system sciences, 55(1), 119-139.
20. Cortes, C., & Vapnik, V. (1995). Support vector networks. Machine Learning, 20(3), 273-297.
21. LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature, 521(7553), 436-444.
22. Τηλλυρος, Χ. (2017). Συγκριτική αξιολόγηση αλγορίθμων μηχανικής μάθησης σε δεδομένα ασθενών με διαβήτη [Comparative evaluation of machine learning algorithms on diabetes patient data]. Πανεπιστήμιο Πειραιά.
23. Georgouli, A. (2015). Τεχνητή νοημοσύνη. Athens: Kallipos, Open Academic Editions. http://hdl.handle.net/11419/3381
24. Muller, A. C., & Guido, S. (2017). Introduction to Machine Learning with Python: A Guide for Data Scientists. O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472.
25. C. Troussas, M. Virvou and S. Mesaretzidis, "Comparative analysis of algorithms for student characteristics classification using a methodological framework," 2015 6th International Conference on Information, Intelligence, Systems and Applications (IISA), 2015, pp. 1-5, doi: 10.1109/IISA.2015.7388038.
26. Τιμούδας, Β. Π. (2022). Συγκριτική αξιολόγηση αλγορίθμων μηχανικής μάθησης: Η περίπτωση της πρόβλεψης δασικών πυρκαγιών. Πανεπιστήμιο Δυτικής Αττικής.
27. Κουβάτσος, Κ. (2017). Μελέτη Αλγορίθμων Πρόβλεψης Καιρικών Συνθηκών Σύμφωνα με Ιστορικά Στοιχεία και Ακολουθίες Προτύπων. Τεχνολογικό Εκπαιδευτικό Ίδρυμα Πελοποννήσου.
28. McKinney, W. (2011). Python for Data Analysis: Data Wrangling with Pandas. Sebastopol, CA: O'Reilly Media.
29. Olliphant, T., & Jones, E. (2006). NumPy: A Guide to NumPy. Trelgol Publishing.
30. McKinney, W. (2010). Data Structures for Statistical Computing in Python. Proceedings of the 9th Python in Science Conference.
31. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, É. (2011). Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research, 12, 2825-2830.
32. Krouska, A., Troussas, C., Sgouropoulou, C. (2020). Applying Genetic Algorithms for Recommending Adequate Competitors in Mobile Game-Based Learning Environments. In: Kumar, V., Troussas, C. (eds) Intelligent Tutoring Systems. ITS 2020. Lecture Notes in Computer Science(), vol 12149. Springer, Cham. https://doi.org/10.1007/978-3-030- 49663-0_23.
33. C. Troussas, M. Virvou, and K. J. Espinosa, "Using visualization algorithms for discovering patterns in groups of users for tutoring multiple languages through Social Networking", Journal of Networks, vol. 10, no. 12, pp. 668-674, 2015.
34. Troussas C., Krouska A., Sgouropoulou C. Improving Learner-Computer Interaction through Intelligent Learning Material Delivery Using Instructional Design Modeling. Entropy. 2021; 23(6):668. https://doi.org/10.3390/e23060668
35. Giannakas, F., Troussas, C., Krouska, A. et al. Multi-technique comparative analysis of machine learning algorithms for improving the prediction of teams' performance. Educ Inf Technol (2022). https://doi.org/10.1007/s10639-022-10900-4.

36. C. Troussas, A. Krouska, F. Giannakas, C. Sgouropoulou, and I. Voyiatzis. Automated reasoning of learners' cognitive states using classification analysis. In 24th Pan-Hellenic Conference on Informatics, pp. 103–106, 2020.
37. Krouska, A., Troussas, C., Sgouropoulou, C. (2020). A Personalized Brain-Based Quiz Game for Improving Students' Cognitive Functions. In: Frasson, C., Bamidis, P., Vlamos, P. (eds) Brain Function Assessment in Learning. BFAL 2020. Lecture Notes in Computer Science(), vol 12462. Springer, Cham. https://doi.org/10.1007/978-3-030- 60735-7_11.
38. Kanetaki, Z., Stergiou, C., Bekas, G., Troussas, C., & Sgouropoulou, C. (2022). A Hybrid Machine Learning Model for Grade Prediction in Online Engineering Education. International Journal of Engineering Pedagogy (iJEP), 12(3), pp. 4–24. https://doi.org/10.3991/ijep.v12i3.23873.
39. Troussas, C., Krouska, A., Sgouropoulou, C. (2020). Dynamic Detection of Learning Modalities Using Fuzzy Logic in Students' Interaction Activities. In: Kumar, V., Troussas, C. (eds) Intelligent Tutoring Systems. ITS 2020. Lecture Notes in Computer Science(), vol 12149. Springer, Cham. https://doi.org/10.1007/978-3-030-49663-0_24.
40. Virvou, M., Troussas, C., Caro, J., Espinosa, K.J. (2012). User Modeling for Language Learning in Facebook. In: Sojka, P., Horák, A., Kopeček, I., Pala, K. (eds) Text, Speech and Dialogue. TSD 2012. Lecture Notes in Computer Science(), vol 7499. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-32790-2_42
41. Krouska, A., Troussas, C., Virvou, M. (2019). Computerized Adaptive Assessment Using Accumulative Learning Activities Based on Revised Bloom's Taxonomy. In: Virvou, M., Kumeno, F., Oikonomou, K. (eds) Knowledge-Based Software Engineering: 2018. JCKBSE 2018. Smart Innovation, Systems and Technologies, vol 108. Springer, Cham. https://doi.org/10.1007/978-3-319-97679-2_26
42. Troussas, C., Virvou, M. & Alepis, E. Comulang: towards a collaborative e-learning system that supports student group modeling. SpringerPlus 2, 387 (2013). https://doi.org/10.1186/2193-1801-2-387
43. A. Krouska, C. Troussas, A. Voulodimos, C. Sgouropoulou, A 2-tier fuzzy control system for grade adjustment based on students' social interactions, Expert Systems with Applications, Volume 203, 2022, 117503, https://doi.org/10.1016/j.eswa.2022.117503.
44. K. Chrysafiadi, C. Troussas and M. Virvou, "A Framework for Creating Automated Online Adaptive Tests Using Multiple-Criteria Decision Analysis," 2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC), 2018, pp. 226-231, doi: 10.1109/SMC.2018.00049.
45. Troussas, C., Virvou, M., Caro, J., & Espinosa, K. J. (2013). Language Learning Assisted by Group Profiling in Social Networks. International Journal of Emerging Technologies in Learning (iJET), 8(3), pp. 35–38. https://doi.org/10.3991/ijet.v8i3.2684.
46. C. Troussas, A. Krouska, E. Alepis & M. Virvou (2020) Intelligent and adaptive tutoring through a social network for higher education, New Review of Hypermedia and Multimedia, 26:3-4, 138-167, DOI: 10.1080/13614568.2021.1908436
47. Papakostas C., Troussas C., Krouska A., Sgouropoulou C. Measuring User Experience, Usability and Interactivity of a Personalized Mobile Augmented Reality Training System. Sensors. 2021; 21(11):3888. https://doi.org/10.3390/s21113888
48. Krouska, A., Troussas, C. and Sgouropoulou, C. 2019. Fuzzy Logic for Refining the Evaluation of Learners' Performance in Online Engineering Education. European Journal of Engineering and Technology Research. 4, 6 (Jun. 2019), 50–56. DOI: https://doi.org/10.24018/ejeng.2019.4.6.1369.