ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

«Εφαρμογή ιστού για κατηγοριοποίηση δεδομένων μέσω δέντρων απόφασης»

Του φοιτητή
Ζωγράφου Ματθαίου
Αρ. Μητρώου: 185179

Επιβλέπων
Ουγιάρογλου Στέφανος
Επίκουρος Καθηγητής

31 Ιανουαρίου 2024

Τίτλος Π.Ε.: Εφαρμογή ιστού για κατηγοριοποίηση δεδομένων μέσω δέντρων απόφασης
Κωδικός Π.Ε.: 23231
Ονοματεπώνυμο φοιτητή: Ζωγράφος Ματθαίος
Ονοματεπώνυμο εισηγητή: Ουγιάρογλου Στέφανος
Ημερομηνία ανάληψης Π.Ε.: 14-07-2023
Ημερομηνία περάτωσης Π.Ε.: 31-01-2024

Περίληψη

Η έννοια της κατηγοριοποίησης (classification), αναφέρεται στη διαδικασία ταξινόμησης κάθε στοιχείου ενός συνόλου δεδομένων σε προκαθορισμένες ομάδες - κατηγορίες, με βάση τα χαρακτηριστικά τους. Για αυτόν το σκοπό, αξιοποιούνται διάφοροι αλγόριθμοι, με τα Δέντρα Απόφασης να αποτελούν έναν εκ των δημοφιλέστερων. Αυτό οφείλεται στην εύκολα κατανοητή δομή τους και στον απλό τρόπο λειτουργίας τους, χαρακτηριστικά που συνέβαλαν στην υιοθέτησή τους από μία πληθώρα εφαρμογών. Κατόπιν σχετικής έρευνας, διαπιστώθηκε ότι η χρήση του αλγορίθμου, συχνά καθίσταται δύσκολη. Οι χρήστες θα πρέπει να είναι εξοικειωμένοι με τα αντικείμενα της Μηχανικής Μάθησης και του προγραμματισμού, με αποτέλεσμα να καταφεύγουν στην αναζήτηση λύσεων λογισμικού. Αφενός, παρατηρείται έλλειψη αυτού και αφετέρου, οι υπάρχουσες λύσεις έχουν μια σειρά από μειονεκτήματα και περιορισμούς. Αυτοί περιλαμβάνουν την απαίτηση αγοράς συνδρομής ή την εγκατάσταση λογισμικού και βιβλιοθηκών, περιπλέκοντας τη χρήση του και καταναλώνοντας πόρους από τον υπολογιστή του χρήστη. Η παρούσα πτυχιακή εργασία στοχεύει να αντιμετωπίσει αυτά τα ζητήματα, μέσα από την ανάπτυξη της εφαρμογής «AutoDTrees». Πρόκειται για μία διαδικτυακή εφαρμογή, που επιτρέπει σε οποιονδήποτε ενδιαφερόμενο να επιλέξει τα επιθυμητά σύνολα δεδομένων και να καθορίσει τις παραμέτρους του κατηγοριοποιητή, ώστε να προχωρήσει στη δημιουργία μοντέλων Δέντρων Απόφασης. Στη συνέχεια, η αποτελεσματικότητα των μοντέλων αξιολογείται μέσω της τεχνικής k-fold cross-validation και παρουσιάζονται εκτενώς τα αποτελέσματα των μετρικών απόδοσης. Ακολούθως, ο χρήστης μπορεί να αποθηκεύσει το προεκπαιδευμένο μοντέλο και να το χρησιμοποιήσει για την πρόβλεψη μη κατηγοριοποιημένων στιγμιοτύπων ή για την οπτικοποίηση του Δέντρου Απόφασης. Η εφαρμογή προσφέρει τη δυνατότητα αξιοποίησης του αλγορίθμου με έναν απλό και γρήγορο τρόπο. Αυτό γίνεται μέσω μίας γραφικής διεπαφής, φιλικής προς τον χρήστη, ενώ παράλληλα διατίθεται και διαδικτυακό API για την ελεύθερη χρήση της εφαρμογής από πλευράς προγραμματιστών.

«Web application for decision tree classification»

«Matthaios Zografos»

Abstract

The concept of classification refers to the process of categorizing each element of a dataset into predefined groups, based on their characteristics. Various algorithms are utilized for this purpose, with Decision Trees being one of the most popular. This is due to their easily understandable structure and simple way of operation, which led to their adoption in a variety of applications. However, after relevant research, it has been found that the usage of this algorithm often becomes challenging. The fact that users need to be familiar with Machine Learning and programming fields, leads them to search for software solutions. On the one hand, there is a lack of that kind of software, and on the other hand, the existing solutions have some disadvantages and limitations. These include the requirement for purchasing a subscription or installing software and external libraries, making the algorithm's usage complicated and resource consuming for a personal computer. The current thesis, aims to address these issues through the development of an application, called "AutoDTrees". This is a web-based application that allows any user to both select the preferred datasets and define classifier parameters, in order to build Decision Tree models. Afterwards, the model effectiveness can be evaluated by using the k-fold cross-validation method. Also, detailed evaluation metrics are presented. Users are then able to save the pretrained model and reuse it for predicting unclassified instances or visualizing the Decision Tree graph. The application offers the ability to utilize the algorithm in a simple and fast way, through a user-friendly graphical interface, while at the same time an open-source Web API is provided for interested developers.

Ευχαριστίες

Αρχικά, θα ήθελα να ευχαριστήσω την οικογένειά μου για την κάθε είδους στήριξη που μου παρείχε καθ' όλη τη διάρκεια εκπόνησης της εργασίας, αλλά και συνολικά σε όλη τη διάρκεια των σπουδών μου. Αισθάνομαι επίσης την ανάγκη, να ευχαριστήσω ιδιαίτερα τον επιβλέποντα της πτυχιακής μου εργασίας, τον Επίκουρο Καθηγητή κ. Στέφανο Ουγιάρογλου. Η συνεισφορά του, μέσα από τις συμβουλές και την καθοδήγησή του, υπήρξε σημαντική για την ολοκλήρωση της εργασίας.

Περιεχόμενα

Κατάλογος Σχημάτων

Κατάλογος Πινάκων

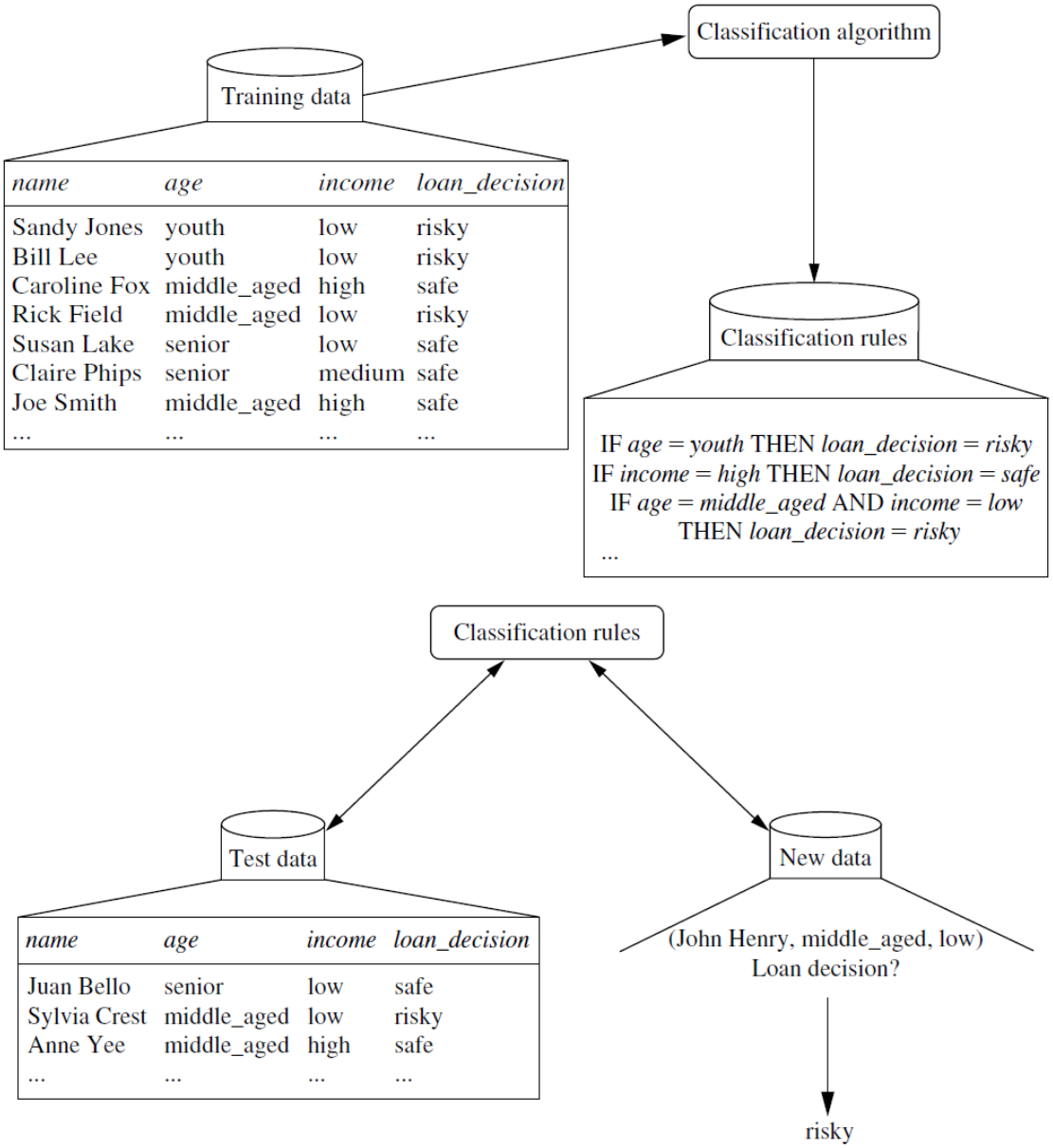
Κεφάλαιο 1ο: Εισαγωγή

1.1 Κατηγοριοποίηση δεδομένων

Η έννοια της κατηγοριοποίησης (classification) στους τομείς της Εξόρυξης Δεδομένων και της Μηχανικής Μάθησης, αναφέρεται στην διαδικασία της ταξινόμησης κάθε στοιχείου ενός συνόλου δεδομένων σε προκαθορισμένες ομάδες - κατηγορίες, λαμβάνοντας υπόψη τα χαρακτηριστικά τους [1]. Ο στόχος είναι να επιλεγούν τα κατάλληλα γνωρίσματα, τα οποία έχουν κρίσιμο ρόλο για την τελική κατάταξη των δεδομένων σε κατηγορίες. Για αυτό το λόγο, αξιοποιούνται κατάλληλοι αλγόριθμοι που εξετάζουν σύνολα δεδομένων (datasets) που έχουν ήδη κατηγοριοποιηθεί, με τελικό σκοπό τη δημιουργία προεκπαιδευμένων μοντέλων. Αυτά τα μοντέλα μπορούν στη συνέχεια να χρησιμοποιηθούν τόσο για τη μελέτη και εξαγωγή συμπερασμάτων για τα υπάρχοντα δεδομένα, όσο και για την κατηγοριοποίηση νέων στιγμιοτύπων [2].

Η κατηγοριοποίηση δεδομένων αποτελεί ένα πεδίο με σημαντικό επιστημονικό ενδιαφέρον, γεγονός που αποδεικνύεται από τον μεγάλο αριθμό μελετών που εστιάζουν σε τομείς, όπως η ανάλυση τεχνικών και αλγορίθμων κατηγοριοποίησης, η αξιολόγηση της απόδοσής τους, καθώς και οι συγκρίσεις μεταξύ αυτών των αλγορίθμων και τεχνικών. Παράλληλα, το ενδιαφέρον αυτό έγκειται και στο γεγονός, ότι το εν λόγω πεδίο έχει εφαρμογές σε πρακτικά ζητήματα της καθημερινότητας και συμβάλλει στην επίλυση προβλημάτων σε διάφορους τομείς. Παραδείγματα αποτελούν η ιατρική, για την αποτελεσματικότερη διάγνωση ασθενειών, όπως και το εμπόριο, για τη βελτιστοποίηση της διαδικασίας προώθησης προϊόντων και υπηρεσιών, μέσα από τη στόχευση σε συγκεκριμένες ομάδες πελατών [1]. Επιπλέον, σημαντική είναι η συνεισφορά και στο πεδίο των οικονομικών, για ζητήματα όπως η έγκριση δανείων, η πρόβλεψη χρεοκοπίας και η αναγνώριση απάτης, στην επιχειρηματικότητα, διευκολύνοντας τις διεργασίες λήψης αποφάσεων [3], αλλά και σε τομείς όπως η σεισμολογία, για την ταχύτερη πρόβλεψη επικίνδυνων φαινομένων [4].

Στο Σχήμα 1.1 [5] απεικονίζεται η διαδικασία κατηγοριοποίησης δεδομένων, χρησιμοποιώντας ως παράδειγμα τα βήματα που ακολουθούνται για την έγκριση ενός δανείου.



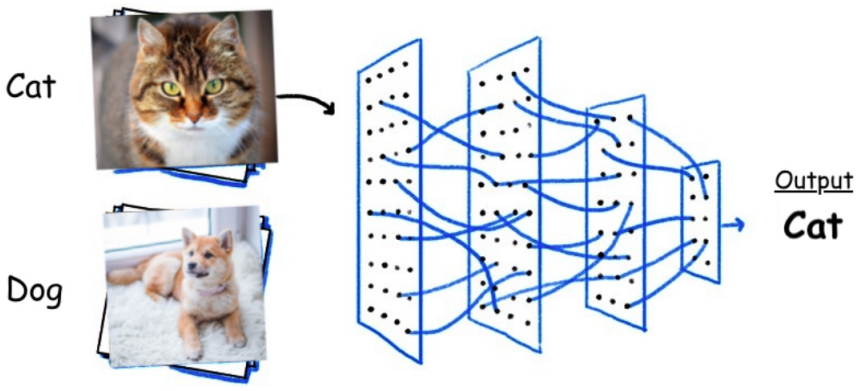
Σχήμα 1.1: Παράδειγμα κατηγοριοποίησης: Η έγκριση δανείων

Η κατηγοριοποίηση δεδομένων εντάσσεται σε μία ευρύτερη ομάδα, αυτή των μεθόδων μάθησης με επίβλεψη (supervised learning). Κύριο χαρακτηριστικό αυτής είναι η εκπαίδευση μοντέλων με χρήση συνόλων δεδομένων, στα οποία είναι εκ των προτέρων γνωστή η ομάδα - κατηγορία στην οποία ανήκει η κάθε πρόβλεψη. Με αυτόν τον τρόπο καθίσταται δυνατή και η εκτίμηση της απόδοσης του μοντέλου που δημιουργείται, γεγονός που λαμβάνεται υπόψιν για την ενδεχόμενη μελλοντική αξιοποίησή του [6]. Στη συγκεκριμένη κατηγορία, εκτός από την κατηγοριοποίηση, ανήκει και μία ακόμη μέθοδος, γνωστή ως παλινδρόμηση (regression). Παρόλο που οι δύο αυτές μέθοδοι ανήκουν στην ίδια κατηγορία, παρουσιάζουν μία σημαντική διαφορά. Αυτή έχει να κάνει με τον τύπο των προβλέψεων που πραγματοποιούν, καθώς στην περίπτωση της κατηγοριοποίησης προκύπτουν διακριτές τιμές, ενώ στην παλινδρόμηση οι τιμές είναι συνεχόμενες [3].

Μία ακόμη κατηγορία μεθόδων με μεγάλο εύρος εφαρμογών στα πεδία της Εξόρυξης Δεδομένων και της Μηχανικής Μάθησης, είναι και αυτή της μάθησης χωρίς επίβλεψη (unsupervised learning). Οι κυριότερες μέθοδοι που ανήκουν σε αυτή είναι η συσταδοποίηση (clustering), η συσχέτιση (association) και η μείωση διαστάσεων (dimensionality reduction). Παρόλο που ο στόχος και των δύο κατηγοριών που αναφέρθηκαν, είναι η αναζήτηση σχέσεων μεταξύ των δεδομένων, θα πρέπει να σημειωθεί ότι παρουσιάζουν μία θεμελιώδη διαφορά. Αυτή έγκειται στο γεγονός ότι στις μεθόδους χωρίς επίβλεψη, η εκπαίδευση του αλγορίθμου πραγματοποιείται χωρίς να είναι εκ των προτέρων γνωστό το μοντέλο πρόβλεψης ή οποιαδήποτε πληροφορία σχετικά με την ομαδοποίηση των δεδομένων [6].

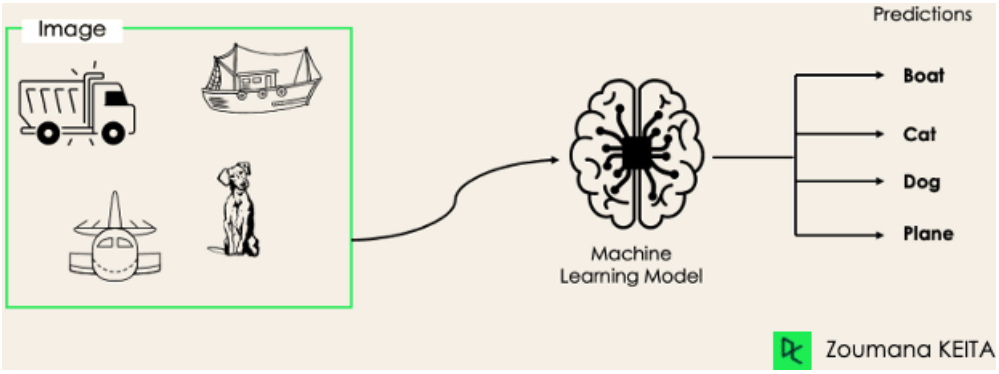
1.2 Τύποι κατηγοριοποίησης και Είδη κατηγοριοποιητών

Η κατηγοριοποίηση δεδομένων κατατάσσεται σε διάφορους τύπους, με βάση τον αριθμό των κατηγοριών - κλάσεων που είναι διαθέσιμες. Οι κυριότεροι από αυτούς τους τύπους είναι η δυαδική κατηγοριοποίηση (binary classification), η κατηγοριοποίηση πολλαπλών κατηγοριών (multi-class classification) και η κατηγοριοποίηση πολλαπλών ετικετών (multi-label classification). Πιο συγκεκριμένα, κατά τη δυαδική κατηγοριοποίηση ο αλγόριθμος καλείται να κατατάξει τα δεδομένα μεταξύ των δύο διαθέσιμων κατηγοριών - κλάσεων. Κάθε στιγμιότυπο δηλαδή, θα πρέπει να ανήκει υποχρεωτικά σε μία μόνο από αυτές τις κατηγορίες, οι τιμές των οποίων είναι συνήθως της μορφής «Αληθής» - «Ψευδής», «Ναι» - «Όχι» ή «0» - «1» [7]. Στο Σχήμα 1.2 [8] φαίνεται ένα παράδειγμα δυαδικής κατηγοριοποίησης, όπου ο αλγόριθμος επιλέγει μεταξύ των κλάσεων «Σκύλος» - «Γάτα».



Σχήμα 1.2: Παράδειγμα δυαδικής κατηγοριοποίησης

Στην περίπτωση της κατηγοριοποίησης πολλαπλών κατηγοριών, ο αλγόριθμος επιλέγει ανάμεσα σε κατηγορίες των οποίων το πλήθος μπορεί να ξεπερνάει τις δύο. Όπως συμβαίνει στην δυαδική κατηγοριοποίηση, έτσι και σε αυτή των πολλαπλών κατηγοριών, κάθε στιγμιότυπο θα πρέπει να ανήκει υποχρεωτικά σε μία μόνο κλάση. Σε ό,τι αφορά τον τελευταίο τύπο, αυτόν της κατηγοριοποίησης πολλαπλών ετικετών, οι διαθέσιμες κατηγορίες μπορούν και σε αυτή την περίπτωση να είναι περισσότερες των δύο. Ωστόσο, η ειδοποιός διαφορά σε σχέση με τον δεύτερο τύπο είναι ότι ένα στιγμιότυπο μπορεί να ανήκει σε πολλές κατηγορίες ταυτόχρονα [7]. Ένα τέτοιο παράδειγμα φαίνεται στο Σχήμα 1.3 [7], σύμφωνα με το οποίο μία εικόνα μπορεί να ανήκει σε πολλές κατηγορίες ανάλογα με τα αντικείμενα που αυτή περιέχει.



Σχήμα 1.3: Παράδειγμα κατηγοριοποίησης πολλαπλών ετικετών

Όσον αφορά τις μεθόδους και αλγορίθμους κατηγοριοποίησης ή απλούστερα κατηγοριοποιητές (classifiers), αυτοί κατατάσσονται σε δύο κύριες κατηγορίες, τους πρόθυμους (eager) και τους οκνηρούς (lazy). Η διάκρισή τους βασίζεται στον τρόπο με τον οποίο κατασκευάζουν μοντέλα. Πιο συγκεκριμένα, οι πρόθυμοι κατηγοριοποιητές δημιουργούν αρχικά ένα μοντέλο, αξιοποιώντας τα σύνολα δεδομένων εκπαίδευσης. Στη συνέχεια, το εν λόγω προεκπαιδευμένο μοντέλο μπορεί να χρησιμοποιηθεί για την κατηγοριοποίηση νέων στιγμιοτύπων [9]. Αντιθέτως, οι οκνηροί κατηγοριοποιητές, αφότου λάβουν τα δεδομένα εκπαίδευσης, δεν προχωρούν στη δημιουργία μοντέλου. Εφόσον κληθούν να κατηγοριοποιήσουν νέα στιγμιότυπα, χρησιμοποιούν τα δεδομένα εκπαίδευσης αντί μοντέλου για την εξαγωγή του τελικού αποτελέσματος [3].

Χαρακτηριστικά παραδείγματα της πρώτης κατηγορίας, αποτελούν αλγόριθμοι όπως τα Δέντρα Αποφάσεων (Decision Trees) [5], ο Naive Bayes [5], τα Νευρωνικά Δίκτυα (Neural Networks) [10], καθώς και οι Μηχανές Διανυσμάτων Υποστήριξης (Support Vector Machines) [10] και η Λογιστική Παλινδρόμηση (Logistic Regression) [3]. Η χρήση των παραπάνω αλγορίθμων είναι προτιμότερη σε περιπτώσεις όπου τα δεδομένα εκπαίδευσης είναι εκ των προτέρων γνωστά και διαθέσιμα στο σύνολό τους, καθώς κάτι τέτοιο είναι απαραίτητο για την δημιουργία του μοντέλου. Συνεπώς, ένα από τα πλεονεκτήματά τους είναι ότι υπερτερούν σε ταχύτητα πρόβλεψης, μιας και η ύπαρξη προεκπαιδευμένου μοντέλου επιτρέπει την άμεση ολοκλήρωση της διαδικασίας στα νέα στιγμιότυπα. Παράλληλα εξοικονομείται μνήμη και επεξεργαστική ισχύς, καθώς σε σχέση με τους οκνηρούς κατηγοριοποιητές δεν απαιτείται η διατήρηση των δεδομένων εκπαίδευσης. Ωστόσο, από την χρήση των πρόθυμων κατηγοριοποιητών προκύπτουν και ορισμένα αρνητικά στοιχεία. Ειδικότερα, παρά την υψηλή ταχύτητα κατά τη διαδικασία πρόβλεψης, οι πρόθυμοι κατηγοριοποιητές παρουσιάζουν χαμηλότερες επιδόσεις κατά την εκπαίδευση του μοντέλου. Το γεγονός αυτό επιδεινώνεται με την αύξηση του όγκου των δεδομένων εκπαίδευσης, ενώ στην περί-

πτωση που προκύψουν αλλαγές σε αυτά, απαιτείται η εκ νέου κατασκευή μοντέλου για την ενσωμάτωσή τους [11].

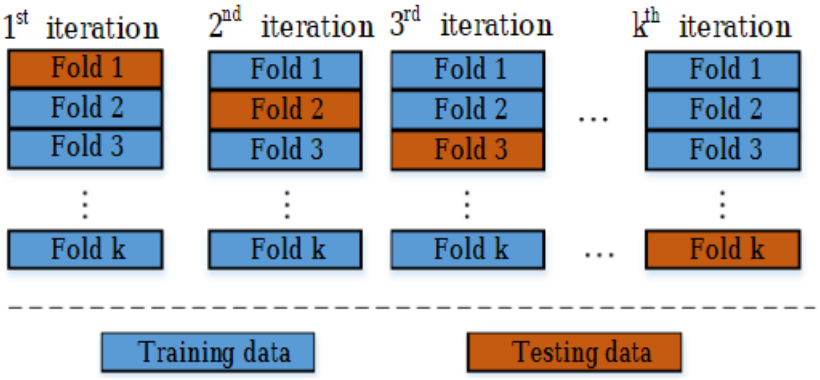
Σχετικά με την δεύτερη κατηγορία, αυτή των οκνηρών κατηγοριοποιητών, κύριο εκπρόσωπό της αποτελεί ο αλγόριθμος των κ - Πλησιέστερων Γειτόνων (k-Nearest Neighbors, k-NN). Ο k-NN προκειμένου να επιτύχει την κατηγοριοποίηση ενός συνόλου δεδομένων, λαμβάνει υπόψιν τις τιμές των προβλέψεων - κλάσεων του συνόλου εκπαίδευσης, που βρίσκονται πλησίον του νέου στιγμιοτύπου. Η χρήση τέτοιου είδους αλγορίθμων προτιμάται σε περιπτώσεις, όπου τα δεδομένα εκπαίδευσης έχουν περίπλοκη δομή και δεν είναι εκ των προτέρων γνωστά στο σύνολό τους, λόγου χάρη όταν υπάρχει ανάγκη συνεχούς ενημέρωσής τους. Μία τέτοια περίπτωση είναι τα συστήματα συστάσεων (recommender systems), όπου οι οκνηροί αλγόριθμοι πραγματοποιούν συγκρίσεις μεταξύ των προτιμήσεων διάφορων χρηστών μίας υπηρεσίας, προκειμένου να παρέχουν σε αυτούς εξειδικευμένες προτάσεις. Το πλεονέκτημα είναι ότι, αφού ληφθούν νέα δείγματα, η διαδικασία ενημέρωσης των δεδομένων εκπαίδευσης ολοκληρώνεται άμεσα υπερτερώντας έναντι των πρόθυμων κατηγοριοποιητών, καθώς δεν απαιτείται να κατασκευαστεί από την αρχή κάποιο μοντέλο. Παρόλα αυτά, δεν παρουσιάζουν ανάλογες επιδόσεις κατά τη διαδικασία πρόβλεψης. Αυτό οφείλεται στο γεγονός ότι ο αλγόριθμος καλείται να αναζητήσει τις όποιες συσχετίσεις, εξετάζοντας ολόκληρο το σύνολο δεδομένων εκπαίδευσης. Όπως γίνεται αντιληπτό, η συγκεκριμένη διεργασία απαιτεί περισσότερο χρόνο και υστερεί σημαντικά σε ταχύτητα, σε σχέση με τους πρόθυμους κατηγοριοποιητές, ενώ αυξημένες είναι και οι απαιτήσεις σε αποθήκευση [12].

1.3 Αξιολόγηση της απόδοσης

Όπως έχει ήδη σημειωθεί, ο στόχος της κατηγοριοποίησης είναι, με την ολοκλήρωση του σταδίου της μάθησης, να δημιουργηθεί ένα προεκπαιδευμένο μοντέλο το οποίο είναι ικανό να πραγματοποιεί προβλέψεις σε νέα, μη κατηγοριοποιημένα στιγμιότυπα. Ένα κρίσιμο λοιπόν ζήτημα που προκύπτει, είναι αυτό που σχετίζεται με την αξιολόγηση της απόδοσης των μοντέλων κατηγοριοποίησης, δηλαδή το κατά πόσο είναι ικανά να εκτελούν σωστές προβλέψεις. Για αυτόν το λόγο υιοθετούνται διάφορες μέθοδοι και τεχνικές, όπως είναι η διασταυρούμενη επικύρωση κ τμημάτων (k-fold cross-validation), η μέθοδος «leave-one-out», η μέθοδος holdout, καθώς και η τυχαία δειγματοληψία (random subsampling) [3, 5].

Σε ό,τι αφορά τη μέθοδο k-fold cross-validation, η διαδικασία που ακολουθείται είναι η εξής: Αρχικά, το δοθέν σύνολο δεδομένων διασπάται με τυχαίο τρόπο σε k διαφορετικά υποσύνολα, τα οποία με τη σειρά τους περιέχουν διαφορετικά στιγμιότυπα. Στη συνέχεια, πραγματοποιείται εκπαίδευση και έλεγχος του μοντέλου, γεγονότα που επαναλαμβάνονται k φορές. Σε κάθε μία από αυτές τις επαναλήψεις, ένα από τα υποσύνολα επιλέγεται για να χρησιμοποιηθεί ως σύνολο ελέγχου (test set) του μοντέλου. Πιο συγκεκριμένα, αν «i» ο δείκτης επανάληψης, στην i-οστή επανάληψη, το i-οστό υποσύνολο θα είναι το σύνολο ελέγχου. Αντίστοιχα, τα υπόλοιπα k-1 υποσύνολα θα αποτελούν το σύνολο εκπαίδευσης (training set), με την απόδοση τελικά να εκτιμάται από τον μέσο όρο των μετρήσεων της κάθε επανάληψης. Το πλεονέκτημα αυτής της μεθόδου είναι το γεγονός, ότι όλα τα υποσύνολα αξιοποιούνται στο σύνολο εκπαίδευσης για τον ίδιο αριθμό επαναλήψεων, ενώ κάθε ένα χρησιμοποιείται ακριβώς μία φορά ως σύνολο ελέγχου [3, 5]. Αξίζει επίσης να σημειωθεί, ότι οι υλοποιήσεις της μεθόδου για k = 5 και k = 10 προτείνονται ως μερικές από τις πιο αξιόπιστες [13].

Στο Σχήμα 1.4 [13] απεικονίζεται η διαδικασία που ακολουθεί η μέθοδος k-fold cross-validation.



Σχήμα 1.4: Η μέθοδος k-fold cross-validation

Μία ειδική περίπτωση της τεχνικής k-fold cross-validation, αποτελεί η μέθοδος «leave-one-out». Ειδικότερα, η τελευταία διαφοροποιείται ως προς το γεγονός, ότι το k ισούται με το πλήθος των δειγμάτων που περιέχει το δοθέν σύνολο δεδομένων. Συνεπώς, ακολουθούνται τα προαναφερθέντα βήματα με το σύνολο ωστόσο να χωρίζεται σε τόσα τμήματα, όσα είναι και τα δείγματα. Ακολούθως, επιλέγεται για κάθε επανάληψη το δείγμα που θα χρησιμοποιηθεί ως σύνολο ελέγχου, με τα υπόλοιπα k-1 να αποτελούν το σύνολο εκπαίδευσης [3, 5]. Ωστόσο, το κύριο μειονέκτημα της «leave-one-out» είναι ότι πρόκειται για μέθοδο με μεγάλο υπολογιστικό κόστος. Αυτό έχει ως αποτέλεσμα να αποφεύγεται η χρήση της σε μεγάλα σύνολα δεδομένων [14].

Σχετικά με τη μέθοδο holdout, ακολουθείται η εξής μεθοδολογία: Αρχικά, το δοθέν σύνολο δεδομένων διασπάται με τυχαίο τρόπο σε δύο διαφορετικά υποσύνολα που περιέχουν διαφορετικά στιγμιότυπα. Η συνήθης πρακτική που ακολουθείται κατά τη δημιουργία τους, είναι η εκχώρηση των δύο τρίτων του αρχικού συνόλου δεδομένων στο πρώτο υποσύνολο, με το υπόλοιπο ένα τρίτο να ανήκει στο δεύτερο υποσύνολο. Αυτά με τη σειρά τους αντιπροσωπεύουν το σύνολο εκπαίδευσης και το σύνολο ελέγχου, αντίστοιχα. Αφότου ολοκληρωθεί το στάδιο της εκπαίδευσης χρησιμοποιώντας το πρώτο σύνολο, προχωράει η διαδικασία ελέγχου, όπου το μοντέλο προβλέπει την κλάση των στιγμιοτύπων του δεύτερου συνόλου. Εν συνεχεία, πραγματοποιεί συγκρίσεις με τις πραγματικές παρατηρήσεις, για την εξαγωγή των τελικών συμπερασμάτων που σχετίζονται με την απόδοση [3, 5]. Τα κύρια μειονεκτήματα αυτής της μεθόδου, είναι ότι χρησιμοποιεί μόνο ένα μέρος του αρχικού συνόλου για τη δημιουργία του μοντέλου, ενώ προκύπτουν και εξαρτήσεις από την διάταξη των δεδομένων. Αποτέλεσμα αυτών είναι η μέθοδος να παρουσιάζει χαμηλές επιδόσεις και χαμηλή αξιοπιστία [10].

Μία παραλλαγή της τεχνικής holdout, είναι η μέθοδος της τυχαίας δειγματοληψίας. Ο τρόπος με τον οποίο λειτουργεί η τελευταία, βασίζεται στην επαναληπτική εκτέλεση της τεχνικής holdout για k φορές. Χάρη σε αυτή τη μεθοδολογία, δημιουργούνται αρκετά σύνολα εκπαίδευσης και ελέγχου και εκτελείται ικανός αριθμός από συγκρίσεις, αντισταθμίζοντας έτσι τα μειονεκτήματα της τεχνικής holdout και παρέχοντας καλύτερη εκτίμηση της απόδοσης ενός μοντέλου. Η τελική αξιολόγηση προκύπτει από τον μέσο όρο, συνυπολογίζοντας τις τιμές των μετρήσεων της κάθε επανάληψης [3, 5, 10].

Κοινό χαρακτηριστικό των μεθόδων που αναφέρθηκαν παραπάνω είναι ότι, προκειμένου να εξάγουν τα τελικά συμπεράσματα για τις εκτιμήσεις τους, αξιοποιούν ορισμένες μετρικές που εκφράζουν την απόδοση των μοντέλων. Μερικές από τις πιο δημοφιλείς είναι η ακρίβεια (accuracy), η ορθότητα (precision), η

ανάκληση (recall), καθώς και το μέτρο f-score. Αξίζει να αναφερθεί ότι τα συγκεκριμένα μέτρα εκφράζονται συναρτήσει ορισμένων εννοιών, που σχετίζονται με τις προβλέψεις. Πιο συγκεκριμένα, ο όρος «True Positive (TP)» χρησιμοποιείται όταν ένα στιγμιότυπο ανήκει σε μία κλάση και ο αλγόριθμος έκανε τη σωστή πρόβλεψη, ενώ καλείται «True Negative (TN)» όταν ένα στιγμιότυπο δεν ανήκει σε μία κλάση και ο αλγόριθμος έκανε τη σωστή πρόβλεψη. Επιπλέον, ο όρος «False Positive (FP)» αναφέρεται όταν ένα στιγμιότυπο δεν ανήκει σε μία κλάση και ο αλγόριθμος προέβλεψε λανθασμένα ότι ανήκει, ενώ ως «False Negative (FN)» καλείται όταν ένα στιγμιότυπο ανήκει σε μία κλάση, αλλά ο αλγόριθμος προέβλεψε λανθασμένα ότι δεν ανήκει. Οι παραπάνω όροι μπορούν να γίνουν καλύτερα κατανοητοί χρησιμοποιώντας τον πίνακα σύγχυσης (confusion matrix). Πρόκειται για μία τεχνική, η οποία συμβάλλει στην καλύτερη παρουσίαση των επιδόσεων των κατηγοριοποιητών, μέσα από την οπτικοποίησή τους με τη μορφή δισδιάστατου πίνακα [3, 5]. Όπως φαίνεται και στον Πίνακα 1.1 [3], οι γραμμές αντιστοιχούν στις πραγματικές τιμές και οι στήλες στις προβλέψεις του αλγορίθμου, ενώ στα κελιά καταγράφεται το πλήθος των «True Positive (TP)», «True Negative (TN)», «False Positive (FP)» και «False Negative (FN)» αντίστοιχα.

Πίνακας 1.1: Ο πίνακας σύγχυσης (confusion matrix)

	Πρόβλεψη Αρνητικής Κλάσης	Πρόβλεψη Θετικής Κλάσης
Πραγματική Αρνητική Κλάση	TN	FP
Πραγματική Θετική Κλάση	FN	TP

Μετά την απαραίτητη επισήμανση των προηγούμενων εννοιών, μπορεί πλέον να γίνει μία εκτενέστερη αναφορά στις προαναφερθείσες μετρικές και στις μαθηματικές σχέσεις με τις οποίες ορίζονται. Πιο συγκεκριμένα, η ακρίβεια (accuracy) εκφράζει το ποσοστό των δειγμάτων που έχουν κατηγοριοποιηθεί σωστά και προκύπτει από τη διαίρεση του πλήθους των σωστών προβλέψεων, με το συνολικό πλήθος των δειγμάτων [3, 5], κάτι που φαίνεται και από τη Σχέση (1.1):

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{1.1}$$

Όσον αφορά την ορθότητα (precision), αυτή προκύπτει από τη διαίρεση του πλήθους των «True Positive», με το άθροισμα του πλήθους των «True Positive» και «False Positive». Επιπλέον, η ανάκληση (recall) προκύπτει από τη διαίρεση του πλήθους των «True Positive», με το άθροισμα του πλήθους των «True Positive» και «False Negative» [5]. Οι παραπάνω τύποι εκφράζονται αντίστοιχα από τη Σχέση (1.2) και τη Σχέση (1.3) που ακολουθούν:

$$Precision = \frac{TP}{TP + FP} \tag{1.2}$$

$$Recall = \frac{TP}{TP + FN} \tag{1.3}$$

Ένα ακόμη χρήσιμο μέτρο είναι και αυτό του f-score, που συνδυάζει τις δύο προαναφερθείσες μετρικές. Ειδικότερα, το μέτρο f-score εκφράζεται ως ο αρμονικός μέσος των precision και recall [5], με τον τύπο να δίνεται από τη Σχέση (1.4):

$$F - score = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} \tag{1.4}$$

1.4 Αυτοματοποιημένη Μηχανική Μάθηση (AutoML)

Η Αυτοματοποιημένη Μηχανική Μάθηση (Automated Machine Learning - AutoML) είναι ένα πεδίο που ασχολείται με την αυτοματοποίηση των διαδικασιών και εργασιών που εφαρμόζονται στη Μηχανική Μάθηση. Κύριος σκοπός της είναι η απλοποίηση αυτών των διαδικασιών προκειμένου να γίνουν περισσότερο κατανοητές και προσιτές, ακόμη και σε άτομα που δεν έχουν την απαραίτητη εξειδίκευση στο εν λόγω πεδίο. Παράλληλα, αποτελεί ένα ιδιαίτερα χρήσιμο εργαλείο και για τους ειδικούς του τομέα, καθώς συμβάλλει στην εξοικονόμηση χρόνου μέσα από τη μείωση του πλήθους των βημάτων που απαιτούνται, για την επιτυχή ολοκλήρωση ορισμένων διαδικασιών [15].

Η AutoML καλύπτει ένα ευρύ φάσμα εργασιών Μηχανικής Μάθησης, από τη λήψη ενός συνόλου δεδομένων έως και τη δημιουργία του προεκπαιδευμένου μοντέλου. Ειδικότερα, τα στάδια που περιλαμβάνονται είναι τα εξής:

Η προεπεξεργασία των δεδομένων
Η AutoML αναλαμβάνει την προετοιμασία των δεδομένων, προκειμένου να λάβουν την κατάλληλη μορφή προτού χρησιμοποιηθούν για την εκπαίδευση του μοντέλου. Οι εργασίες που εκτελούνται περιλαμβάνουν μεταξύ άλλων την κανονικοποίηση των δεδομένων, καθώς και τον χειρισμό ελλειπών τιμών και κατηγορικών δεδομένων [16].

Η επιλογή των κατάλληλων χαρακτηριστικών
Η AutoML εφαρμόζει διαδικασίες μέσα από τις οποίες αναλύονται τα δεδομένα που έχουν δοθεί από τους χρήστες. Με αυτόν τον τρόπο εξάγονται χρήσιμα χαρακτηριστικά που συμβάλλουν στην βελτιστοποίηση του μοντέλου [16].

Η επιλογή του κατάλληλου αλγορίθμου
Η AutoML έχει τη δυνατότητα να εφαρμόζει παράλληλα μια σειρά από αλγορίθμους, ώστε μέσα από την αξιολόγηση της απόδοσής τους, να καταλήξει στην επιλογή εκείνου που κρίνεται ως ο καταλληλότερος για την επίλυση του ζητούμενου προβλήματος [16].

Η βελτιστοποίηση των παραμέτρων
Μέσω της AutoML απλοποιείται και το στάδιο της επιλογής των παραμέτρων και ρυθμίσεων του μοντέλου. Ειδικότερα, ακολουθείται μια αυτοματοποιημένη διαδικασία, κατά την οποία γίνεται δοκιμή διάφορων τιμών προκειμένου να εντοπιστούν εκείνες οι οποίες παράγουν τα βέλτιστα αποτελέσματα [16].

Δημιουργία και αξιολόγηση του μοντέλου
Με την ολοκλήρωση των προηγούμενων σταδίων, είναι πλέον εφικτή η δημιουργία του προεκπαιδευμένου μοντέλου με βάση τις παραμέτρους που έχουν δοθεί. Στη συνέχεια, η AutoML προχωράει στον

έλεγχο και την αξιολόγηση της απόδοσής του. Παράλληλα, παρέχει εκτενείς αναφορές σχετικά με τα χαρακτηριστικά του μοντέλου, για την καλύτερη πληροφόρηση των χρηστών, προτού εκείνοι προχωρήσουν στην εφαρμογή του σε ένα πραγματικό περιβάλλον εργασίας [16].

Όπως γίνεται αντιληπτό και από τα προαναφερθέντα, η AutoML παρουσιάζει πληθώρα θετικών χαρακτηριστικών, τα οποία και την έχουν καταστήσει ένα ιδιαίτερα δημοφιλές εργαλείο. Πιο συγκεκριμένα, επιταχύνει και απλοποιεί τις διαδικασίες που ακολουθούνται στη Μηχανική Μάθηση, καθώς αναλαμβάνει την εκτέλεση ορισμένων εργασιών που είναι ιδιαίτερα χρονοβόρες και απαιτητικές για τους ανθρώπους, επιτρέποντάς τους να επικεντρωθούν σε άλλες σημαντικότερες. Με αυτόν τον τρόπο αυξάνεται η παραγωγικότητά τους, ενώ δημιουργούνται περισσότερο αποδοτικά μοντέλα, σε σχέση με τις απλές μεθόδους ανάπτυξης που δεν περιλαμβάνουν τη χρήση αυτοματοποιημένων διαδικασιών. Επίσης, η χρήση της AutoML καθίσταται ιδιαίτερα επωφελής για επιχειρήσεις και οργανισμούς, καθώς γίνεται περισσότερο προσιτή η υιοθέτηση μεθόδων Μηχανικής Μάθησης και Ανάλυσης Δεδομένων, μιας και οι απαιτήσεις σε χρόνο και πόρους είναι σημαντικά μικρότερες [17]. Ωστόσο, πέρα από την πληθώρα πλεονεκτημάτων, από τη χρήση της AutoML προκύπτουν και ορισμένα αρνητικά στοιχεία. Ένα από τα κυριότερα, είναι το γεγονός ότι πρόκειται για ένα νέο πεδίο που εξελίσσεται ταχύτατα, με αποτέλεσμα η ανάπτυξη λογισμικού που χρησιμοποιεί την AutoML να καθίσταται δυσκολότερη, ενώ αρκετά από τα ήδη υπάρχοντα εργαλεία έχουν περιορισμένες δυνατότητες [15, 17]. Επιπλέον, είναι σημαντικό να τονιστεί ότι η AutoML δεν θα πρέπει να λογίζεται ως ένα μέσο που αντικαθιστά πλήρως τον άνθρωπο, αλλά ως ένα εργαλείο που υποστηρίζει το εξειδικευμένο προσωπικό, για τη διευκόλυνση της εκτέλεσης συγκεκριμένων εργασιών [17].

1.5 Κίνητρο και Συνεισφορά

Μέχρι και αυτό το σημείο έχει γίνει εκτενής αναφορά σε ορισμένες εισαγωγικές έννοιες και αντικείμενα, όπως η κατηγοριοποίηση δεδομένων. Το κύριο θέμα το οποίο πραγματεύεται η παρούσα πτυχιακή εργασία και για το οποίο θα γίνει εκτενέστερη αναφορά στη συνέχεια, είναι τα Δέντρα Απόφασης. Πρόκειται για έναν αλγόριθμο κατηγοριοποίησης, ο οποίος θεωρείται ιδιαίτερα δημοφιλής, ο τρόπος λειτουργίας του είναι εύκολος στην κατανόηση λόγω της δομής του, ενώ έχει καταφέρει να διεισδύσει και σε μία πληθώρα εφαρμογών.

Παρόλα αυτά, κατόπιν σχετικής έρευνας διαπιστώθηκε ότι η χρήση του εν λόγω αλγορίθμου, αρκετά συχνά καθίσταται δύσκολη. Το κύριο εμπόδιο είναι ότι οι χρήστες θα πρέπει να διαθέτουν εξειδικευμένες γνώσεις και εμπειρία στα αντικείμενα της Μηχανικής Μάθησης και της Εξόρυξης Δεδομένων, αλλά και εξοικείωση με τον προγραμματισμό. Για αυτόν το λόγο, καταφεύγουν στην αναζήτηση σχετικών ψηφιακών εργαλείων. Ωστόσο, παρατηρείται η έλλειψη εξειδικευμένου λογισμικού, μέσω του οποίου οι ενδιαφερόμενοι χρήστες θα μπορούν να αποκτούν πρόσβαση στον αλγόριθμο και να εκτελούν απρόσκοπτα τα επιθυμητά πειράματα ή εργασίες που σχετίζονται με αυτόν. Είναι δεδομένη η ύπαρξη ορισμένων λύσεων λογισμικού Μηχανικής Μάθησης και Εξόρυξης Δεδομένων που περιλαμβάνουν, μεταξύ άλλων αλγορίθμων και τα Δέντρα Απόφασης. Είναι σημαντικό όμως να τονιστεί, ότι αυτά τα εργαλεία παρουσιάζουν μια σειρά από μειονεκτήματα, θέτοντας έτσι σοβαρούς περιορισμούς στην αξιοποίηση του αλγορίθμου από τους χρήστες. Πρώτο και κύριο, είναι το γεγονός ότι αρκετές εφαρμογές απαιτούν την καταβολή σημαντικού ποσού για την αγορά συνδρομής, καθώς σε διαφορετική περίπτωση παρέχονται ελάχιστες ή και καμία από τις δυνατότητές τους δωρεάν. Ένα ακόμη ζήτημα, είναι ότι ορισμένα εργαλεία

παρέχονται με τη μορφή εφαρμογής για υπολογιστές. Ως εκ τούτου, προκειμένου το εν λόγω είδος να μπορεί να χρησιμοποιηθεί πλήρως, απαιτείται η εκ των προτέρων εγκατάσταση πακέτων λογισμικού ή ακόμη και υποστηρικτικών βιβλιοθηκών, κάνοντας έτσι περίπλοκη τη χρήση του και καταναλώνοντας πόρους από τον προσωπικό υπολογιστή του χρήστη.

Τα παραπάνω ζητήματα στοχεύει να αντιμετωπίσει η παρούσα πτυχιακή εργασία, μέσα από την ανάπτυξη μίας διαδικτυακής εφαρμογής, που φέρει την ονομασία «AutoDTrees». Η εν λόγω εφαρμογή αποτελεί ένα ελεύθερο και ανοικτού κώδικα λογισμικό, που επιτρέπει σε οποιονδήποτε ενδιαφερόμενο να χρησιμοποιήσει τον αλγόριθμο των Δέντρων Αποφάσεων και τα χαρακτηριστικά που αυτός προσφέρει. Πιο συγκεκριμένα, το AutoDTrees δίνει τη δυνατότητα στους χρήστες να επιλέξουν τα σύνολα δεδομένων που επιθυμούν, ώστε με βάση αυτά να προχωρήσουν στη δημιουργία προεκπαιδευμένων μοντέλων. Ενδιάμεσα, παρεμβάλλεται το στάδιο του καθορισμού των παραμέτρων του κατηγοριοποιητή, όπου οι χρήστες μπορούν είτε να εισάγουν τις τιμές που επιθυμούν, είτε να χρησιμοποιήσουν τις προκαθορισμένες, επιτρέποντας έτσι και σε όσους δεν έχουν την σχετική εμπειρία, να μπορούν να δημιουργούν μοντέλα. Στο επόμενο στάδιο, αξιοποιείται η τεχνική k-fold cross-validation για την αξιολόγηση της αποτελεσματικότητας του μοντέλου, παρέχοντας στους χρήστες εκτενή αναφορά με την παρουσίαση μετρικών απόδοσης, τόσο για την κάθε τιμή της κλάσης, όσο και για το σύνολο του μοντέλου. Με βάση τα αποτελέσματα, ο χρήστης έχει ακολούθως την επιλογή να αποθηκεύσει το προεκπαιδευμένο μοντέλο, ώστε να το χρησιμοποιήσει μελλοντικά για την πρόβλεψη μη κατηγοριοποιημένων στιγμιοτύπων, όπως και για την οπτικοποίηση του δέντρου απόφασης.

Το σύνολο των λειτουργιών της εφαρμογής παρέχεται μέσω ενός ελεύθερου διαδικτυακού API, επιτρέποντας με αυτόν τον τρόπο την αξιοποίησή της από την κοινότητα των προγραμματιστών, είτε επεκτείνοντάς την, είτε αναπτύσσοντας νέες εφαρμογές της αρεσκείας τους, χρησιμοποιώντας μέρος των δυνατοτήτων της παρούσας υλοποίησης. Παράλληλα, έχει αναπτυχθεί σύγχρονη γραφική διεπαφή, ώστε κάθε χρήστης να μπορεί με έναν απλό και γρήγορο τρόπο να προσπελάσει τις λειτουργίες της εφαρμογής, μέσα από ένα φιλικό προς αυτόν περιβάλλον, χωρίς να απαιτείται η χρήση προγραμματιστικών εργαλείων. Το AutoDTrees φιλοξενείται από server του Τμήματος Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων, ωστόσο η πλήρης υλοποίηση είναι διαθέσιμη σε κάθε ενδιαφερόμενο μέσω της πλατφόρμας του GitHub, δίνοντας με αυτόν τον τρόπο τη δυνατότητα να μπορεί να αντιγραφεί και να φιλοξενηθεί από οποιονδήποτε εξωτερικό server, για τις ανάγκες των εκάστοτε χρηστών ή και οργανισμών. Συμπερασματικά, πρόκειται για μία εφαρμογή που συμβάλλει ποικιλοτρόπως στην κάλυψη του κενού που παρατηρείται σε αυτήν την κατηγορία λογισμικού και μπορεί να αποτελέσει ένα χρήσιμο εργαλείο για μία ευρεία ομάδα χρηστών. Αυτοί μπορούν να είναι ερευνητές, φοιτητές, προγραμματιστές, στελέχη που εξειδικεύονται στην Επιστήμη των Δεδομένων, αλλά και σε κάθε άλλο άτομο που ενδιαφέρεται για τους τομείς της Μηχανικής Μάθησης και της Εξόρυξης Δεδομένων.

Η εφαρμογή AutoDTrees είναι διαθέσιμη και πλήρως προσβάσιμη από τον παρακάτω σύνδεσμο: `https://kclusterhub.iee.ihu.gr/autodtrees`.

Επιπλέον, ο κώδικας είναι προσβάσιμος μέσω του GitHub στον ακόλουθο σύνδεσμο: `https://github.com/manthoszog/AutoDTrees`.

1.6 Οργάνωση της εργασίας

Η πτυχιακή εργασία στο σύνολό της οργανώνεται σε επτά κεφάλαια. Ολοκληρώνοντας το πρώτο κεφάλαιο και κάνοντας μία ανασκόπηση των όσων αναφέρθηκαν μέχρι τώρα, έγινε εκτενής παρουσίαση εισαγωγικών εννοιών, όπως η κατηγοριοποίηση δεδομένων και οι τύποι της, τα είδη κατηγοριοποιητών και οι τεχνικές που χρησιμοποιούνται για την αξιολόγηση της απόδοσης ενός μοντέλου. Παράλληλα, έγινε μία εισαγωγή στην έννοια της Αυτοματοποιημένης Μηχανικής Μάθησης (AutoML), προτού αναλυθούν τόσο το κίνητρο για την εκπόνηση της συγκεκριμένης πτυχιακής εργασίας, όσο και η συνεισφορά της. Ακολουθεί μία συνοπτική παρουσίαση των επόμενων κεφαλαίων.

Στο δεύτερο κεφάλαιο, θα γίνει εκτενής αναφορά στον αλγόριθμο των Δέντρων Απόφασης, καταγράφοντας τον τρόπο λειτουργίας του, τη συνεισφορά του σε πραγματικές εφαρμογές, τα πλεονεκτήματα και μειονεκτήματά του, καθώς και τις διάφορες υλοποιήσεις του.

Στο τρίτο κεφάλαιο, θα παρουσιαστούν οι γλώσσες προγραμματισμού, όπως και οι τεχνολογίες που χρησιμοποιήθηκαν κατά την ανάπτυξη της εφαρμογής, τόσο σε επίπεδο back-end, όσο και σε αυτό του front-end.

Στο τέταρτο κεφάλαιο, θα γίνει εκτενής αναφορά στη διαδικασία που ακολουθήθηκε για τη σχεδίαση και υλοποίηση της εφαρμογής. Ξεκινώντας, θα παρουσιαστούν οι λειτουργικές απαιτήσεις και η αρχιτεκτονική του AutoDTrees, καταλήγοντας στην ανάλυση των επιμέρους λειτουργιών που αναπτύχθηκαν στο front-end και στο back-end. Παράλληλα, θα παρατίθενται και τα σχετικά στιγμιότυπα από τον κώδικα.

Στο πέμπτο κεφάλαιο, θα γίνει μία παρουσίαση των λειτουργιών της εφαρμογής μέσω της γραφικής της διεπαφής, καθοδηγώντας ταυτόχρονα τους χρήστες για το πώς θα αξιοποιήσουν πλήρως το σύνολο των δυνατοτήτων που αυτή προσφέρει.

Στο έκτο κεφάλαιο, παρατίθενται στοιχεία από την αξιολόγηση της εφαρμογής. Ειδικότερα, παρουσιάζονται τα αποτελέσματα από τις μετρήσεις εμπειρίας χρήστη, για τις ανάγκες των οποίων χρησιμοποιήθηκε σχετικό ερωτηματολόγιο.

Η πτυχιακή εργασία ολοκληρώνεται με το έβδομο κεφάλαιο. Σε αυτό γίνεται αφενός μία καταγραφή των συμπερασμάτων που εξάγονται από την εκπόνηση της εργασίας και αφετέρου παρουσιάζονται ιδέες για μελλοντική επέκταση της εφαρμογής.

Κεφάλαιο 2ο: Δέντρα Απόφασης

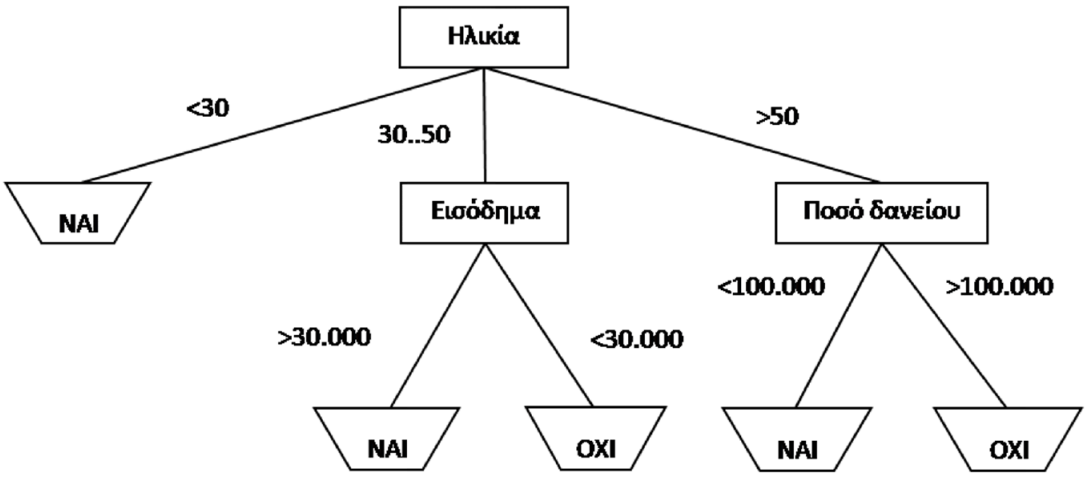
2.1 Εισαγωγή

Τα Δέντρα Απόφασης (Decision Trees) είναι ένας αλγόριθμος που χρησιμοποιείται για την κατηγοριοποίηση δεδομένων και θεωρείται ως ένας εκ των δημοφιλέστερων αυτού του είδους. Αυτό το γεγονός οφείλεται στη δομή του που παρουσιάζεται με τη μορφή δενδροδιαγράμματος, κάτι που τον καθιστά εύκολο στην κατανόηση, ακόμη και από άτομα που δεν έχουν ιδιαίτερη εμπειρία στους τομείς της Μηχανικής Μάθησης και της Εξόρυξης Δεδομένων. Παράλληλα, πρόκειται για έναν αλγόριθμο που χαρακτηρίζεται για τον απλό τρόπο λειτουργίας, όπως και τις υψηλές επιδόσεις στην δημιουργία μοντέλων, κάτι που αποτυπώνεται και από την ευρεία εφαρμογή του σε διάφορους τομείς, όπως η ιατρική, η μοριακή βιολογία, η αστρονομία και η χρηματοοικονομική ανάλυση [3, 5].

Όσον αφορά τον τρόπο με τον οποίο αναπαρίσταται, ένα Δέντρο Απόφασης περιλαμβάνει «κόμβους», «κλαδιά» και «φύλλα». Οι κόμβοι εκφράζουν τους ελέγχους που γίνονται στα δεδομένα, τα κλαδιά δηλώνουν τα αποτελέσματα των ελέγχων, ενώ τα φύλλα περιλαμβάνουν τις αποφάσεις που προκύπτουν από την διαδικασία της κατηγοριοποίησης. Κάθε δέντρο ξεκινάει με τον κόμβο - ρίζα που βρίσκεται στην κορυφή του και μέσω των κλαδιών, συνδέεται με τους κόμβους του επόμενου χαμηλότερου επιπέδου. Στη συνέχεια, κάθε κόμβος συνδέεται με τους κόμβους του αμέσως επόμενου επιπέδου, μέχρις ότου τελικά αυτοί να καταλήξουν στο τελευταίο επίπεδο, όπου βρίσκονται τα φύλλα. Προκειμένου να επιτύχουν αυτή τη διαδικασία, τα Δέντρα Απόφασης προχωρούν σε διαδοχικές διαιρέσεις του δοθέντος συνόλου δεδομένων, από τις οποίες προκύπτουν τόσα υποσύνολα, όσα και τα αποτελέσματα των ελέγχων που γίνονται στους κόμβους. Επιπλέον, θα πρέπει να σημειωθεί ότι τα Δέντρα έχουν συνήθως δυαδική μορφή, δηλαδή κάθε κόμβος - έλεγχος οδηγεί σε δύο κόμβους - αποτελέσματα. Υπάρχουν όμως και ορισμένες υλοποιήσεις που υποστηρίζουν και μη δυαδικές μορφές, όπου ένας έλεγχος μπορεί να οδηγήσει σε παραπάνω από δύο αποτελέσματα [3, 5].

Με την ολοκλήρωση του σταδίου της εκπαίδευσης και τη δημιουργία προεκπαιδευμένου μοντέλου, εισάγονται νέα στιγμιότυπα προς κατηγοριοποίηση. Κάθε ένα από αυτά, ξεκινάει μία πορεία από τον κόμβο - ρίζα και με βάση τα αποτελέσματα των ελέγχων που πραγματοποιούνται, ακολουθεί μία διαδρομή μεταξύ διάφορων κόμβων, μέχρι να καταλήξει σε ένα φύλλο που ορίζει την απόφαση κατηγοριοποίησης [3, 5]. Για την καλύτερη κατανόηση των εννοιών για τις οποίες έγινε λόγος μέχρι τώρα, αξίζει να αναφερθεί και ένα παράδειγμα όπως αυτό του Σχήματος 2.1 [3] που ακολουθεί. Στο συγκεκριμένο παράδειγμα, παρουσιάζεται ένα Δέντρο Απόφασης που αντιπροσωπεύει τη διαδικασία έγκρισης ενός τραπεζικού δανείου [3]. Αρχικά, εισάγεται ένα σύνολο δεδομένων προς κατηγοριοποίηση, το οποίο περιέχει τα στοιχεία ατόμων που έχουν υποβάλλει αίτηση. Ξεκινώντας από τον κόμβο - ρίζα, κάθε εγγραφή του συνόλου ελέγχεται ως προς την ηλικία του ατόμου. Έτσι προκύπτουν τρεις περιπτώσεις, που αναπαριστώνται με τα αντίστοιχα κλαδιά. Το πρώτο κλαδί περιέχει τα άτομα που είναι κάτω των 30 ετών, το δεύτερο όσους είναι από 30 έως 50 έτη, ενώ το τρίτο αφορά άτομα άνω των 50 ετών. Εφόσον το άτομο ανήκει στην πρώτη περίπτωση, τότε το κλαδί οδηγεί στο φύλλο με την τιμή «ΝΑΙ», υποδεικνύοντας έτσι ότι η αίτηση του συγκεκριμένου υποψηφίου πρόκειται να εγκριθεί. Σε αντίθεση με το πρώτο, τα άλλα δύο κλαδιά οδηγούν εκ νέου σε κόμβο, όπου ο υποψήφιος θα ελεγχθεί ως προς το εισόδημά του ή το ποσό δανείου, αντίστοιχα. Εφόσον το εισόδημα είναι μεγαλύτερο των 30.000, ο κόμβος οδηγεί στο φύλλο που υποδεικνύει την έγκριση, διαφορετικά το δάνειο θα απορριφθεί. Ομοίως και στην τελευταία περίπτωση, όπου

αν το ποσό δανείου είναι μικρότερο των 100.000, το δάνειο εγκρίνεται, ενώ αν είναι μεγαλύτερο αυτού του ποσού, τότε θα απορριφθεί.



Σχήμα 2.1: Παράδειγμα Δέντρου Απόφασης: Η έγκριση δανείων

2.2 Πλεονεκτήματα και Μειονεκτήματα

Με βάση τα όσα αναφέρθηκαν μέχρι στιγμής, γίνεται αντιληπτό ότι τα Δέντρα Απόφασης παρουσιάζουν μια πληθώρα από θετικά χαρακτηριστικά, τα οποία και έχουν καταστήσει το συγκεκριμένο είδος αλγορίθμου ιδιαίτερα δημοφιλές. Τα κυριότερα από αυτά είναι τα εξής:

Ευκολία στην κατανόηση

Τα Δέντρα Απόφασης, χάρη στη δομή τους, μπορούν να οπτικοποιηθούν μέσα από μία απλή διαδικασία. Έτσι διευκολύνουν τους χρήστες στο να κατανοήσουν τον τρόπο, με τον οποίο λειτουργεί και αναπαριστά την γνώση ένα μοντέλο. Ένα παράδειγμα είναι τα χαρακτηριστικά γνωρίσματα του μοντέλου. Χάρη στην ύπαρξη ιεραρχικής δομής, ένας χρήστης μπορεί να καταλάβει ποια από αυτά είναι τα πιο σημαντικά, για τη λήψη μίας απόφασης κατηγοριοποίησης [18].

Είναι μη παραμετρικά

Τα Δέντρα Απόφασης αποτελούν μία μη παραμετρική μέθοδο μάθησης. Συνεπώς, η δημιουργία του μοντέλου δεν εξαρτάται από τον καθορισμό διάφορων σύνθετων παραμέτρων [3], αλλά αντίθετα το μοντέλο βασίζεται στα δεδομένα των παρατηρήσεων [19].

Υποστήριξη πολλών τύπων δεδομένων

Τα Δέντρα Απόφασης έχουν την ικανότητα χειρισμού διαφόρων τύπων δεδομένων, κάτι που τα καθιστά ιδιαίτερα ευέλικτα σε σύγκριση με άλλους αλγορίθμους κατηγοριοποίησης. Πιο συγκεκριμένα, η εν λόγω μέθοδος υποστηρίζει τόσο ονομαστικά, όσο και αριθμητικά δεδομένα, ενώ παράλληλα είναι εφικτή και η χρήση δεδομένων με ελλιπείς τιμές [3, 18]. Επιπλέον, αξίζει να σημειωθεί ότι ορισμένες υλοποιήσεις των Δέντρων Απόφασης, εκτός των διακριτών, υποστηρίζουν και συνεχείς τιμές. Αυτό το γεγονός δίνει τη δυνατότητα, πέρα από την κατηγοριοποίηση, να χρησιμοποιούνται και για εργασίες παλινδρόμησης (regression) [18].

Χαμηλό υπολογιστικό κόστος

Ένα ακόμη σημαντικό πλεονέκτημα των Δέντρων Απόφασης, είναι ότι πρόκειται για μία μέθοδο με χαμηλό υπολογιστικό κόστος. Με αυτόν τον τρόπο, η διαδικασία κατασκευής μοντέλων χαρακτηρίζεται για την υψηλή της ταχύτητα, ακόμη και όταν χρησιμοποιούνται σύνολα δεδομένων μεγάλου όγκου. Επιπρόσθετα, ιδιαίτερα ταχύτατη είναι και η ολοκλήρωση της διαδικασίας κατηγοριοποίησης νέων στιγμιοτύπων [10].

Ανθεκτικά στην ύπαρξη περιττών χαρακτηριστικών

Τα Δέντρα Απόφασης, χαρακτηρίζονται επίσης για την ανθεκτικότητά τους στην ύπαρξη περιττών χαρακτηριστικών, εντός του συνόλου δεδομένων εκπαίδευσης. Θα πρέπει να σημειωθεί ότι ο όρος «περιττό», αναφέρεται όταν ένα γνώρισμα παρουσιάζει έντονη συσχέτιση με κάποιο άλλο. Σε αυτή λοιπόν την περίπτωση και προκειμένου να προχωρήσει η διαδικασία διαχωρισμού του συνόλου εκπαίδευσης, ο αλγόριθμος θα επιλέξει μόνο το ένα από αυτά τα χαρακτηριστικά, παραλείποντας το άλλο [10].

Παρά την ύπαρξη αρκετών θετικών στοιχείων, όπως αυτά αναφέρθηκαν προηγουμένως, οι διαδικασίες που ακολουθούν τα Δέντρα Απόφασης παρουσιάζουν και μία σειρά από μειονεκτήματα. Τα σημαντικότερα από αυτά είναι τα παρακάτω:

Ευαισθησία σε μεταβολές των δεδομένων

Τα Δέντρα Απόφασης παρουσιάζουν ιδιαίτερη ευαισθησία όταν το σύνολο δεδομένων εκπαίδευσης μεταβάλλεται. Μάλιστα, ακόμα και οι μικρής έκτασης μεταβολές μπορεί να επιφέρουν ως συνέπεια τη δημιουργία εντελώς διαφορετικών μοντέλων κατηγοριοποίησης [3].

Δημιουργία υπερπροσαρμοσμένων μοντέλων

Σε ορισμένες περιπτώσεις, τα μοντέλα Δέντρων Απόφασης έχουν την τάση να υπερπροσαρμόζονται στα δεδομένα. Αυτό το γεγονός, παρουσιάζεται συνήθως σε Δέντρα που έχουν σύνθετη μορφή και αναπτύσσονται καταλαμβάνοντας πολλά επίπεδα [18]. Εξαιτίας αυτού του μειονεκτήματος, τα μοντέλα δεν εξάγουν γενικούς κανόνες και συσχετίσεις μεταξύ των δεδομένων, αλλά αντίθετα ενσωματώνουν ειδικούς κανόνες και συμπεράσματα που προκύπτουν από τα δεδομένα εκπαίδευσης [3].

Ευαισθησία σε μη ισορροπημένα σύνολα

Τα Δέντρα Απόφασης παρουσιάζουν ευαισθησία, όταν πρόκειται να χειριστούν μη ισορροπημένα σύνολα δεδομένων. Πιο συγκεκριμένα, όταν σε ένα σύνολο δεδομένων εκπαίδευσης υπάρχουν κλάσεις που επικρατούν έναντι άλλων, τότε ο κατηγοριοποιητής τείνει να δημιουργεί μοντέλα που μεροληπτούν υπέρ αυτών των κλάσεων [20].

Μη πρακτικά για χρήση σε μεγάλα σύνολα

Τα Δέντρα Απόφασης, όταν πρόκειται για μεγάλα σύνολα δεδομένων εκπαίδευσης, χαρακτηρίζονται από έλλειψη πρακτικότητας. Αυτό εξηγείται από το γεγονός, ότι όσο μεγαλύτερο είναι ένα σύνολο δεδομένων, τόσο περισσότερα και τα χαρακτηριστικά γνωρίσματα που περιέχονται σε αυτό. Συνεπώς, αυξάνονται παράλληλα και οι διακλαδώσεις του Δέντρου, με αποτέλεσμα να καθίσταται από δύσκολη έως και αδύνατη η κατανόηση και η ερμηνεία του [2].

2.3 Κριτήρια επιλογής χαρακτηριστικών

Όπως έχει ήδη αναφερθεί, προκειμένου να δημιουργηθεί ένα μοντέλο Δέντρου Απόφασης, ακολουθείται μία διαδικασία διαδοχικών διαιρέσεων του συνόλου δεδομένων εκπαίδευσης. Ένα ζήτημα που προκύπτει είναι η επιλογή του κριτηρίου, σύμφωνα με το οποίο θα διαιρεθεί το σύνολο. Ειδικότερα, κάθε κόμβος ενός Δέντρου θα πρέπει εντός της συνθήκης ελέγχου, να περιλαμβάνει το χαρακτηριστικό που διαχωρίζει με τον βέλτιστο τρόπο τα δεδομένα [3]. Προκειμένου κάτι τέτοιο να καταστεί δυνατό, αξιοποιούνται ορισμένα κριτήρια, με τα δημοφιλέστερα από αυτά να είναι το Κέρδος Πληροφορίας (Information Gain), ο Λόγος Κέρδους (Gain Ratio), καθώς και ο Δείκτης Gini (Gini Index) [5].

2.3.1 Κέρδος Πληροφορίας (Information Gain)

Το Κέρδος Πληροφορίας εκφράζει τη διαφορά της εντροπίας, πριν και μετά το διαχωρισμό ενός συνόλου, δεδομένου ενός χαρακτηριστικού [18]. Το Κέρδος Πληροφορίας υπολογίζεται αρχικά για κάθε χαρακτηριστικό του συνόλου. Στη συνέχεια, γίνεται σύγκριση των αποτελεσμάτων και τελικά επιλέγεται το χαρακτηριστικό που έχει τη μεγαλύτερη τιμή, καθώς σύμφωνα με το εν λόγω κριτήριο, θα είναι εκείνο που θα οδηγήσει στη βέλτιστη διαίρεση του συνόλου [3]. Όσον αφορά την έννοια της εντροπίας, πρόκειται για ένα μέτρο που εκτιμά την ανομοιογένεια των τιμών σε ένα δείγμα του συνόλου δεδομένων, με το εύρος τιμών που λαμβάνει να είναι εντός του αριθμητικού συνόλου $(0, 1)$. Συνεπώς, όσο μικρότερη είναι η τιμή της εντροπίας, τόσο μεγαλύτερη είναι η ομοιογένεια [21].

Ο τύπος της εντροπίας δίνεται από τη Σχέση (2.1):

$$E(S) = -\sum_{i=1}^{c} p_i \cdot \log_2(p_i) \tag{2.1}$$

όπου S είναι το σύνολο δεδομένων που εξετάζεται, c είναι το πλήθος των κλάσεων του συνόλου και p_i είναι το ποσοστό των στιγμιοτύπων που ανήκουν στην κλάση i [3].

Για τον υπολογισμό του Κέρδους Πληροφορίας θα πρέπει, πέρα από την εντροπία του συνόλου, να είναι γνωστή και η εντροπία που προκύπτει από τον διαχωρισμό του συνόλου σε υποσύνολα, με βάση τις τιμές του επιλεγμένου χαρακτηριστικού [3]. Ο υπολογισμός της είναι εφικτός μέσω της Σχέσης (2.2):

$$E(S, A) = \sum_{j=1}^{u} \frac{s_j}{s} \cdot E(S_j) \tag{2.2}$$

όπου A είναι το επιλεγμένο χαρακτηριστικό, u είναι το πλήθος των τιμών του χαρακτηριστικού, S_j είναι το υποσύνολο που θα δημιουργηθεί, s_j είναι το πλήθος των στιγμιοτύπων του S_j και s είναι το πλήθος των στιγμιοτύπων του αρχικού συνόλου S [3].

Μετά τον ορισμό των παραπάνω, το Κέρδος Πληροφορίας μπορεί να υπολογιστεί χρησιμοποιώντας τη Σχέση (2.3):

$$IG(S, A) = E(S) - E(S, A) \tag{2.3}$$

2.3.2 Λόγος Κέρδους (Gain Ratio)

Ένα ακόμη κριτήριο που χρησιμοποιείται για τη βέλτιστη επιλογή χαρακτηριστικών, είναι και ο Λόγος Κέρδους. Πρόκειται για ένα μέτρο που αποτελεί επέκταση του Κέρδους Πληροφορίας και προκύπτει από την κανονικοποίηση του τελευταίου ως προς την εντροπία. Με αυτόν τον τρόπο, γίνεται εφικτό να ξεπεραστούν ορισμένα ζητήματα που προκύπτουν από τη χρήση του Κέρδους Πληροφορίας. Ένα από τα κυριότερα, είναι το γεγονός ότι τείνει να επιλέγει χαρακτηριστικά που έχουν μεγάλο πλήθος τιμών, τα οποία ωστόσο δεν περιέχουν χρήσιμη πληροφορία. Συμπερασματικά, ο Λόγος Κέρδους αντιμετωπίζει αποτελεσματικά τα προαναφερθέντα ζητήματα, μειώνοντας τελικά την πολυπλοκότητα του παραγόμενου Δέντρου Απόφασης και βελτιώνοντας παράλληλα την ακρίβεια των αποτελεσμάτων [3, 5].

Ο υπολογισμός του Λόγου Κέρδους είναι εφικτός μέσω της Σχέσης (2.4):

$$GainRatio(S, A) = \frac{IG(S, A)}{E(S, A)} \tag{2.4}$$

2.3.3 Δείκτης Gini (Gini Index)

Ο Δείκτης Gini αποτελεί με τη σειρά του ένα ακόμη δημοφιλές κριτήριο, για τον βέλτιστο διαχωρισμό ενός συνόλου δεδομένων. Χαρακτηρίζεται για τον απλό και εύκολα κατανοητό τρόπο λειτουργίας, ενώ θεωρείται ιδιαίτερα αποδοτικό σε περιπτώσεις, όπου το πλήθος των κλάσεων που διερευνώνται είναι σχετικά μικρό [21]. Όπως ακριβώς συμβαίνει και στην περίπτωση της εντροπίας, ο Δείκτης Gini εκτιμά την ανομοιογένεια των τιμών σε ένα δείγμα του συνόλου δεδομένων, με το εύρος τιμών που λαμβάνει να είναι εντός του αριθμητικού συνόλου $(0, 1)$ [22]. Όπως γίνεται αντιληπτό, όσο μικρότερη είναι η τιμή του Δείκτη, τόσο μικρότερη η αβεβαιότητα του μοντέλου που θα δημιουργηθεί και άρα μεγαλύτερη ομοιογένεια στα δεδομένα [21].

Ο τύπος για τον υπολογισμό του Δείκτη Gini δίνεται από τη Σχέση (2.5):

$$Gini(S) = 1 - \sum_{j=1}^{k} p_j^2 \tag{2.5}$$

όπου S είναι το σύνολο δεδομένων που εξετάζεται, k είναι το πλήθος των κλάσεων του συνόλου και p_j είναι η πιθανότητα εμφάνισης της κλάσης j στο σύνολο δεδομένων S [22].

Στην περίπτωση όπου το σύνολο δεδομένων διασπάται σε δύο υποσύνολα, ο υπολογισμός του Δείκτη Gini είναι εφικτός μέσω της Σχέσης (2.6):

$$Gini(S) = \frac{n_1}{n} \cdot Gini(S_1) + \frac{n_2}{n} \cdot Gini(S_2) \tag{2.6}$$

όπου S_1 και S_2 είναι τα δύο υποσύνολα, n_1 και n_2 είναι το πλήθος των στιγμιοτύπων που ανήκουν αντίστοιχα στα προαναφερθέντα υποσύνολα και n είναι το συνολικό πλήθος των στιγμιοτύπων [22].

2.4 Υλοποιήσεις του αλγορίθμου

Προκειμένου να καταστεί εφικτή η κατηγοριοποίηση δεδομένων χρησιμοποιώντας τα Δέντρα Απόφασης, κρίνεται απαραίτητη η υιοθέτηση κάποιας εκ των υλοποιήσεων που έχουν δημιουργηθεί για αυτόν το σκοπό. Ορισμένοι από τους δημοφιλέστερους αλγορίθμους κατασκευής Δέντρων Απόφασης, για τους οποίους θα γίνει λόγος στη συνέχεια, είναι ο ID3, ο C4.5, καθώς και ο αλγόριθμος CART.

2.4.1 Ο αλγόριθμος ID3

Η λειτουργία του αλγορίθμου ID3 βασίζεται στην επαναληπτική διαίρεση ενός συνόλου δεδομένων, αξιοποιώντας το κριτήριο του Κέρδους Πληροφορίας (Information Gain). Συνεπώς, σε κάθε επανάληψη επιλέγει το γνώρισμα με τη μεγαλύτερη τιμή, καθώς σύμφωνα με το εν λόγω κριτήριο, είναι αυτό που διαιρεί το σύνολο με τον βέλτιστο τρόπο [23]. Ακολουθώντας την παραπάνω λογική, τα Δέντρα αρχικά αναπτύσσονται μέχρι να φτάσουν το μέγιστο επίπεδό τους και στη συνέχεια, προκειμένου να περιοριστεί το μέγεθος, εφαρμόζονται τεχνικές κλαδέματος (pruning). Το τελευταίο είναι ένα ιδιαίτερα σημαντικό βήμα, καθώς συμβάλλει στη δημιουργία αποδοτικότερων μοντέλων κατηγοριοποίησης [20].

Μερικά ακόμη χαρακτηριστικά του αλγορίθμου ID3, εκτός των προαναφερθέντων, είναι ότι τείνει να δημιουργεί μικρά σε έκταση Δέντρα Απόφασης [23], ενώ έχει επιπλέον τον περιορισμό, ότι μπορεί να χειρίζεται μόνο χαρακτηριστικά που περιλαμβάνουν ονομαστικές τιμές και όχι αριθμητικές. Παράλληλα, θα πρέπει να σημειωθεί ότι τα Δέντρα που παράγει ο ID3 έχουν μη δυαδική μορφή, δηλαδή κάθε κόμβος - έλεγχος μπορεί να οδηγήσει σε παραπάνω από δύο κόμβους - αποτελέσματα [20].

Αλγόριθμος 1 Ο αλγόριθμος ID3

1: **ΥΠΟΛΟΓΙΣΕ** το Κέρδος Πληροφορίας κάθε χαρακτηριστικού.
2: **ΘΕΣΕ** ως ρίζα του Δέντρου το χαρακτηριστικό με το μεγαλύτερο Κέρδος Πληροφορίας.
3: **ΕΠΑΝΑΛΑΒΕ**
4: **ΔΗΜΙΟΥΡΓΗΣΕ** τόσα κλαδιά όσες είναι οι τιμές του χαρακτηριστικού.
5: **ΧΩΡΙΣΕ** το σύνολο δεδομένων σε τόσα υποσύνολα όσες είναι οι τιμές του χαρακτηριστικού.
6: **ΕΠΙΛΕΞΕ** ένα υποσύνολο που δεν χρησιμοποιήθηκε ήδη.
7: **ΑΝ** όλες οι εγγραφές του υποσυνόλου ανήκουν στην ίδια κλάση
ΤΟΤΕ
ΠΗΓΑΙΝΕ στο Βήμα 8.
ΑΛΛΙΩΣ
ΠΗΓΑΙΝΕ στο Βήμα 9.
8: **ΘΕΣΕ** ως φύλλο την τιμή της κλάσης και **ΠΗΓΑΙΝΕ** στο Βήμα 6.
9: **ΥΠΟΛΟΓΙΣΕ** το Κέρδος Πληροφορίας των υπόλοιπων χαρακτηριστικών του υποσυνόλου.
10: **ΘΕΣΕ** ως κόμβο το χαρακτηριστικό με το μεγαλύτερο Κέρδος Πληροφορίας.
11: **ΜΕΧΡΙΣ ΟΤΟΥ** να μην δημιουργούνται νέα φύλλα.

Ο Αλγόριθμος 1 [22] παρουσιάζει συγκεντρωτικά τα βήματα που υιοθετεί ο ID3, για την κατασκευή ενός Δέντρου Απόφασης. Αναλυτικότερα, η διαδικασία που ακολουθείται είναι η εξής: Αρχικά, δημιουργείται ο κόμβος - ρίζα που αντιστοιχεί στο πλήρες σύνολο δεδομένων εκπαίδευσης, προτού ξεκινήσει η επαναληπτική διάσπασή του. Στη συνέχεια, για κάθε χαρακτηριστικό γνώρισμα του συνόλου υπολογίζεται το Κέρδος Πληροφορίας και επιλέγεται εκείνο που διαθέτει τη μεγαλύτερη τιμή, για να τοποθετηθεί στον κόμβο - ρίζα. Παράλληλα, προχωράει η διαίρεση του συνόλου σε υποσύνολα με βάση το πλήθος των τιμών του χαρακτηριστικού, δημιουργώντας και τα αντίστοιχα κλαδιά. Ακολούθως, για κάθε ένα εκ

των υποσυνόλων ελέγχονται οι εγγραφές και εφόσον όλες αντιστοιχούν στην ίδια κλάση, δημιουργείται το αντίστοιχο φύλλο που περιλαμβάνει την τιμή της κλάσης. Σε διαφορετική περίπτωση, αντί για φύλλο κατασκευάζεται ένας νέος κόμβος. Το γνώρισμα που θα τοποθετηθεί εντός αυτού, προκύπτει από τον εκ νέου υπολογισμό του Κέρδους Πληροφορίας για τα υπόλοιπα γνωρίσματα του υποσυνόλου. Ομοίως, τα βήματα επαναλαμβάνονται εκ νέου διαχωρίζοντας περαιτέρω το σύνολο δεδομένων, με τη διαδικασία να ολοκληρώνεται όταν δεν θα είναι δυνατό να δημιουργηθούν νέα φύλλα [22].

2.4.2 Ο αλγόριθμος C4.5

Ο αλγόριθμος C4.5 αποτελεί μία επέκταση του ID3, με την κύρια διαφορά τους να είναι το κριτήριο με βάση το οποίο διαχωρίζεται το σύνολο δεδομένων. Ειδικότερα, ο C4.5 αξιοποιεί τον Λόγο Κέρδους (Gain Ratio) έναντι του Κέρδους Πληροφορίας, με αποτέλεσμα τα μοντέλα που παράγονται να είναι λιγότερο πολύπλοκα και να έχουν μεγαλύτερη ακρίβεια, σε σχέση με εκείνα του προκατόχου του [3]. Με βάση τα παραπάνω, ο C4.5 προχωράει στη δημιουργία του Δέντρου μέσα από την επαναληπτική διαίρεση του συνόλου δεδομένων εκπαίδευσης, επιλέγοντας κάθε φορά το γνώρισμα με το μεγαλύτερο Λόγο Κέρδους. Με την ολοκλήρωση της κατασκευής του Δέντρου, ο αλγόριθμος εφαρμόζει τεχνικές κλαδέματος, οδηγώντας με αυτόν τον τρόπο στην αύξηση της απόδοσης των μοντέλων [24] και στη μείωση του πλήθους των κανόνων που τα αποτελούν, αφαιρώντας όσους κρίνονται περιττοί [25].

Όσον αφορά τα πλεονεκτήματα του C4.5, τα κυριότερα από αυτά είναι ότι έχει την ικανότητα χειρισμού χαρακτηριστικών που περιέχουν αριθμητικά δεδομένα, πέρα από ονομαστικά, ενώ παρουσιάζει ευχέρεια στην επεξεργασία χαρακτηριστικών με ελλιπείς τιμές [24]. Μάλιστα, στην τελευταία περίπτωση οι εν λόγω τιμές, κατά τη διάρκεια των υπολογισμών του Λόγου Κέρδους, εξαιρούνται [25]. Επιπλέον, η ευελιξία των Δέντρων Απόφασης που παράγει ο αλγόριθμος αυξάνεται περαιτέρω, χάρη στο γεγονός ότι υποστηρίζουν τόσο διακριτές, όσο και συνεχείς τιμές στα γνωρίσματα [24].

Από την άλλη πλευρά, ο τρόπος λειτουργίας του C4.5 παρουσιάζει και ορισμένα μειονεκτήματα. Ένα από τα κυριότερα, είναι το γεγονός ότι έχει ιδιαίτερη ευαισθησία στην ύπαρξη θορύβου στα δεδομένα, κάτι που μπορεί στη συνέχεια να οδηγήσει στη δημιουργία υπερπροσαρμοσμένων μοντέλων. Επίσης, αξίζει να σημειωθεί ότι σε ορισμένες περιπτώσεις έχει παρατηρηθεί, ότι ο αλγόριθμος παρουσιάζει την τάση να δημιουργεί κλαδιά, τα οποία περιέχουν μηδενικές τιμές. Κάτι τέτοιο, έχει αρνητικό αντίκτυπο στην πολυπλοκότητα και τον όγκο του παραγόμενου Δέντρου Απόφασης, μιας και πρόκειται για περιττά στοιχεία που δεν συντελούν με κανένα τρόπο στην εξαγωγή χρήσιμων συμπερασμάτων [24].

2.4.3 Ο αλγόριθμος CART

Μία ακόμη δημοφιλή υλοποίηση των Δέντρων Απόφασης, αποτελεί και ο αλγόριθμος CART. Όπως συμβαίνει και με τους προαναφερθέντες αλγορίθμους, ο CART προχωράει στην επαναληπτική διαίρεση του συνόλου δεδομένων εκπαίδευσης, για την κατασκευή του Δέντρου. Η κύρια διαφορά εντοπίζεται στο γεγονός, ότι για την επιλογή των χαρακτηριστικών που διαχωρίζουν το σύνολο, χρησιμοποιείται το κριτήριο του Δείκτη Gini (Gini Index). Συνεπώς, σε κάθε επανάληψη επιλέγεται το χαρακτηριστικό γνώρισμα που έχει την μικρότερη τιμή, καθώς σύμφωνα με το συγκεκριμένο κριτήριο θα είναι εκείνο που διαιρεί το σύνολο με τον βέλτιστο τρόπο [25]. Θα πρέπει επίσης να σημειωθεί, ότι σε αντίθεση με τους ID3 και C4.5, το Δέντρο που προκύπτει έχει δυαδική μορφή, δηλαδή κάθε κόμβος - έλεγχος

οδηγεί σε ακριβώς δύο κόμβους - αποτελέσματα [24]. Ακόμη, με την ολοκλήρωση του σταδίου της κατασκευής του μοντέλου, εφαρμόζονται και σε αυτή την περίπτωση προηγμένες τεχνικές κλαδέματος για τη βελτιστοποίηση του Δέντρου [25].

Ο αλγόριθμος CART παρουσιάζει μια σειρά από σημαντικά πλεονεκτήματα. Πρώτο και κύριο, είναι το γεγονός ότι παρέχει στους χρήστες τη δυνατότητα να αξιοποιούν τα Δέντρα Απόφασης, όχι μόνο για την κατηγοριοποίηση δεδομένων, αλλά και για εργασίες παλινδρόμησης. Όπως γίνεται αντιληπτό, με αυτόν τον τρόπο ο αλγόριθμος μπορεί, εκτός από διακριτές τιμές κλάσεων, να πραγματοποιεί προβλέψεις και για συνεχή διαστήματα τιμών [24]. Επιπλέον, ο CART χαρακτηρίζεται για την ακρίβεια στις προβλέψεις και την υψηλή ταχύτητα κατά την κατασκευή των μοντέλων Δέντρων Απόφασης, ενώ παρουσιάζει ιδιαίτερη αποτελεσματικότητα στο χειρισμό γνωρισμάτων που περιλαμβάνουν ελλιπείς τιμές [3]. Μάλιστα στην τελευταία περίπτωση, όταν ανιχνευτούν τέτοιου είδους γνωρίσματα, προκειμένου να εξαχθούν τα τελικά συμπεράσματα ο αλγόριθμος υλοποιεί πλήθος εναλλακτικών ελέγχων για την προσέγγιση του αποτελέσματος [25]. Εκτός από τα πλεονεκτήματα που αναλύθηκαν παραπάνω, κατά τη χρήση του αλγορίθμου CART προκύπτουν και ορισμένα μειονεκτήματα. Το κυριότερο από αυτά είναι το γεγονός, ότι σε μερικές περιπτώσεις τα παραγόμενα Δέντρα Απόφασης χαρακτηρίζονται από αστάθεια. Ειδικότερα, ακόμα και το ενδεχόμενο μικρών αλλαγών στις παραμέτρους του μοντέλου ή στο σύνολο δεδομένων εκπαίδευσης, μπορεί να οδηγήσει στην τροποποίηση των γνωρισμάτων που χρησιμεύουν στη διαίρεση του συνόλου, αλλάζοντας έτσι σημαντικά τη δομή και την πολυπλοκότητα του Δέντρου [24].

2.5 Επίλογος

Ολοκληρώνοντας το Κεφάλαιο 2, στα πλαίσιά του έγινε λόγος στα Δέντρα Απόφασης και τη μεθοδολογία που ακολουθείται τόσο για την κατασκευή των μοντέλων, όσο και για την κατηγοριοποίηση νέων δεδομένων. Επίσης, καταγράφηκαν οι απαραίτητες ορολογίες που τα συνοδεύουν, ενώ παρουσιάστηκαν τα πλεονεκτήματα και μειονεκτήματα, καθώς και οι διάφοροι αλγόριθμοι που χρησιμοποιούνται για την υλοποίησή τους.

Κεφάλαιο 3ο: Γλώσσες και Τεχνολογίες

3.1 Εισαγωγή

Σε αυτό το κεφάλαιο, θα γίνει αναφορά στις τεχνολογίες και τις γλώσσες προγραμματισμού που χρησιμοποιήθηκαν κατά την ανάπτυξη της εφαρμογής, τόσο σε επίπεδο back-end, όσο και σε αυτό του front-end. Ο όρος «back-end» αφορά το τμήμα εκείνο που εκτελείται από την πλευρά του server, δεν είναι ορατό στον τελικό χρήστη και περιλαμβάνει λειτουργίες που αποτελούν το υπόβαθρο της εφαρμογής. Από την άλλη, ο όρος front-end σχετίζεται με το τμήμα εκείνο που εκτελείται από την πλευρά του περιηγητή και περιλαμβάνει τη γραφική διεπαφή, με την οποία αλληλεπιδρά ο χρήστης.

3.2 Back-end

3.2.1 REST API

Ο όρος «Διεπαφή Προγραμματισμού Εφαρμογών» (Application Programming Interface) ή απλούστερα API, αναφέρεται σε ένα σύνολο από κανόνες που καθορίζουν τον τρόπο με τον οποίο διάφορες εφαρμογές ή τμήματά τους, επικοινωνούν μεταξύ τους και ανταλλάσσουν πληροφορίες. Αποτελούν ένα χρήσιμο εργαλείο, τόσο για τους προγραμματιστές, όσο και για εταιρείες ή οργανισμούς, καθώς διευκολύνουν τη μεταφορά και τον διαμοιρασμό δεδομένων μεταξύ διαφόρων συστημάτων, ενώ επιτρέπουν την παροχή πρόσβασης σε ορισμένες λειτουργικότητες των εφαρμογών. Το τελευταίο, είναι ένα από τα κυριότερα πλεονεκτήματα ενός API, μιας και συμβάλλει στην ευκολότερη ανάπτυξη νέων, καθώς και στην κατανόηση, βελτίωση και επέκταση των ήδη υπαρχόντων εφαρμογών. Ένα ακόμη πλεονέκτημα, είναι ότι τα API προσθέτουν επιπλέον επίπεδα ασφαλείας κατά τη ροή πληροφοριών μεταξύ των εφαρμογών, διαχωρίζοντας το σύστημα που αποστέλλει αιτήματα, από εκείνο που τα διεκπεραιώνει και αποστέλλει την απόκριση [26].

Ένα API καλείται και REST API, εφόσον ακολουθεί ορισμένους κανόνες τους οποίους επιτάσσει η αρχιτεκτονική REST (Representational State Transfer). Οι κυριότεροι από αυτούς είναι οι εξής:

Ομοιόμορφη διασύνδεση
Σε ένα REST API, τα αιτήματα που αφορούν την ίδια οντότητα θα πρέπει να έχουν παρόμοια μορφή και να είναι προσβάσιμα μέσω ενός μοναδικού αναγνωριστικού URL [27].

Διαχωρισμός μεταξύ πελάτη και εξυπηρέτη
Η αρχιτεκτονική REST API, προβλέπει την ύπαρξη διαφορετικών και ανεξάρτητων εφαρμογών για την πλευρά του πελάτη (client) και αυτήν του εξυπηρέτη (server). Η όποια αλληλεπίδρασή τους συνίσταται στην ανταλλαγή πληροφοριών μέσω αιτημάτων, με χρήση του πρωτοκόλλου HTTP [27].

Ακαταστατικότητα
Προκειμένου ένα API να ακολουθεί την αρχιτεκτονική REST, θα πρέπει να είναι ακαταστατικό. Κάτι τέτοιο σημαίνει ότι δεν επιτρέπεται στον εξυπηρέτη να διατηρεί δεδομένα των αιτημάτων. Συνεπώς, κατά την αποστολή τους θα πρέπει εκ των προτέρων να ενσωματώνουν το σύνολο των απαραίτητων πληροφοριών που τα συνοδεύουν [27].

Όπως προαναφέρθηκε, ένα REST API προκειμένου να επιτύχει την ανταλλαγή δεδομένων αξιοποιεί το πρωτόκολλο HTTP. Ειδικότερα γίνεται χρήση μεθόδων, όπως η GET για την ανάκτηση δεδομένων και η POST για την καταχώριση νέων. Επιπλέον μεθόδους, αποτελούν η PUT που χρησιμεύει στην ενημέρωση ενός πόρου, καθώς και η DELETE για την διαγραφή του. Παράλληλα, για τη διευκόλυνση της διαδικασίας επικοινωνίας μεταξύ πελάτη και εξυπηρέτη, συνηθίζεται η μορφοποίηση των δεδομένων χρησιμοποιώντας ορισμένα πρότυπα. Ένα από τα δημοφιλέστερα είναι το πρότυπο JSON (JavaScript Object Notation) που χαρακτηρίζεται για την υψηλή συμβατότητά του με τις γλώσσες προγραμματισμού, όπως επίσης για το γεγονός ότι είναι εύκολο στην κατανόηση [27].

3.2.2 PHP

Η PHP είναι μία γλώσσα προγραμματισμού γενικού σκοπού που χρησιμοποιείται κατά κύριο λόγο για την ανάπτυξη δυναμικών διαδικτυακών εφαρμογών. Ξεκινώντας την πορεία της το 1994, η ονομασία της προήλθε από τα αρχικά των λέξεων «Personal Home Page», ωστόσο στην συνέχεια αυτή τροποποιήθηκε σε «Hypertext Preprocessor» προκειμένου να ανταποκρίνεται καλύτερα στα χαρακτηριστικά και τον ρόλο της. Ειδικότερα, πρόκειται για μία ανοικτού κώδικα scripting γλώσσα που εκτελείται από την πλευρά του server, γνωρίζει ιδιαίτερη δημοφιλία και μπορεί να αξιοποιηθεί για την ανάπτυξη ενός ευρέως φάσματος εφαρμογών και λειτουργιών [28].

Μερικά από τα κυριότερα χαρακτηριστικά που προσφέρει η συγκεκριμένη γλώσσα στους προγραμματιστές είναι τα εξής:

Συμβατότητα

Η PHP παρουσιάζει υψηλή συμβατότητα, καλύπτοντας μεγάλο πλήθος δημοφιλών λειτουργικών συστημάτων. Αυτό το γεγονός συνέβαλε στην εξάλειψη των διάφορων περιορισμών, διευκολύνοντας τη συνεργασία μεταξύ πολλών τύπων συστημάτων, αλλά και των χρηστών τους [28]. Παράλληλα, πέραν των προαναφερθέντων, διακρίνεται για την ικανότητα να συνεργάζεται με την πλειοψηφία των σύγχρονων τύπων server, ενώ η ευελιξία της αυξάνεται επιπλέον χάρη και στο γεγονός ότι υποστηρίζει ένα ευρύ φάσμα συστημάτων διαχείρισης βάσεων δεδομένων [29].

Ευκολία στην εκμάθηση και κατανόηση

Η PHP αποτελεί μία γλώσσα προγραμματισμού, η οποία σε σύγκριση με άλλες γλώσσες, θεωρείται αρκετά εύκολη τόσο στην εκμάθησή της, όσο και στην κατανόηση του τρόπου με τον οποίο συντάσσεται και λειτουργεί. Παράλληλα, χάρη στην ύπαρξη μεγάλης κοινότητας προγραμματιστών που την χρησιμοποιεί, είναι εφικτή η πρόσβαση σε πληθώρα υποστηρικτικού υλικού, διευκολύνοντας περαιτέρω την εκμάθησή της ακόμη και από άτομα χωρίς ιδιαίτερη εμπειρία [28].

Είναι ανοικτού κώδικα

Η PHP είναι μία ανοικτού κώδικα γλώσσα, κάτι που επιτρέπει την περαιτέρω αξιοποίησή της από την κοινότητα των προγραμματιστών, συμβάλλοντας στη συνεχή βελτίωση και επέκτασή της. Παράλληλα, διατίθεται δωρεάν απαλλάσσοντας τους χρήστες από την υποχρέωση απόκτησης συνδρομής και καταβολής κάποιου επιπλέον ποσού [28].

Όσον αφορά τον τρόπο με τον οποίο συντάσσεται η PHP, θα πρέπει αρχικά να αναφερθεί ότι ο κώδικας περικλείεται εντός των χαρακτηριστικών ετικετών `'<?php'` και `'?>'` που σημαίνουν την έναρξη και το

τέλος, αντίστοιχα. Κάτι τέτοιο, επιτρέπει παράλληλα την ενσωμάτωσή του σε αρχείο με κώδικα HTML. Ωστόσο η συνηθέστερη επιλογή, είναι ο κώδικας να περιλαμβάνεται εντός ενός ξεχωριστού αρχείου, με την επέκταση `'.php'`. Μερικά ακόμη σημαντικά στοιχεία της σύνταξης, είναι ότι οι τύποι των μεταβλητών δεν δίνονται από τους προγραμματιστές, αλλά καθορίζονται αυτόματα κατά την εκτέλεση, ενώ το όνομα κάθε μεταβλητής αρχίζει πάντοτε με το σύμβολο `'$'`. Επιπλέον, θα πρέπει να σημειωθεί ότι η PHP επιτρέπει τη σύνταξη κώδικα με τη χρήση είτε της διαδικαστικής, είτε της αντικειμενοστρεφούς προσέγγισης [29].

Στο Σχήμα 3.1 φαίνεται ένα παράδειγμα σύνταξης κώδικα PHP.

```php
<?php
    require_once "../dbconnect.php";
    $method = $_SERVER['REQUEST_METHOD'];
    $input = json_decode(file_get_contents('php://input'),true);
    if($method != "POST") {
        header("HTTP/1.1 405 Method Not Allowed");
        print json_encode(['errormesg'=>"Method not allowed."]);
        exit;
    }
    if(!isset($input['email'])){
        header("HTTP/1.1 400 Bad Request");
        print json_encode(['errormesg'=>"Email is not set."]);
        exit;
    }
?>
```

Σχήμα 3.1: Παράδειγμα σύνταξης κώδικα PHP

3.2.3 Composer

Το Composer αποτελεί ένα ιδιαίτερα δημοφιλές εργαλείο που χρησιμοποιείται υποστηρικτικά από την γλώσσα προγραμματισμού PHP, για τη διαχείριση εξαρτήσεων (dependency management). Η ιδέα για τη δημιουργία του, καθώς και ο τρόπος λειτουργίας του βασίστηκε σε άλλα προϋπάρχοντα εργαλεία, όπως τα npm και bundler των node.js και ruby, αντίστοιχα [30].

Τα βήματα τα οποία ακολουθούνται από το Composer, είναι τα εξής:

Αρχικά, δίνεται η δυνατότητα στους προγραμματιστές να δηλώσουν, μέσα από απλά βήματα και εντολές, τις βιβλιοθήκες που προτίθενται να χρησιμοποιήσουν, κατά την ανάπτυξη μίας εφαρμογής.

Στη συνέχεια, εκείνο θα προχωρήσει, μέσω μίας αυτοματοποιημένης διαδικασίας, στην αναζήτηση και τον εντοπισμό του συνόλου των πακέτων και αρχείων που αυτές περιλαμβάνουν, καθώς και τις εκδόσεις που απαιτούνται.

Τελικά, το Composer πραγματοποιεί τη λήψη και εγκατάστασή τους τοπικά, στον φάκελο όπου βρίσκεται και ο κώδικας της εφαρμογής. Επιπρόσθετα, δίνεται στους χρήστες η επιλογή να ενημερώσουν ή και να διαγράψουν τις εξαρτήσεις που επιθυμούν, με τη χρήση απλών εντολών [30].

Στο Σχήμα 3.2, φαίνεται ένα παράδειγμα της μορφής με την οποία γίνεται η δήλωση των εξαρτήσε-

ων. Αρχικά, δημιουργείται το αρχείο `'composer.json'`, ενώ στη συνέχεια πρέπει να κληθεί η εντολή `'composer install'`.

```
{
    "require": {
        "phpmailer/phpmailer": "^6.8.0",
        "php": "^7.4",
        "vendor/package": "^version"
    }
}
```

Σχήμα 3.2: Παράδειγμα δήλωσης εξαρτήσεων στο Composer

3.2.4 MySQL

Όσον αφορά τη MySQL, πρόκειται για ένα ιδιαίτερα δημοφιλές Σχεσιακό Σύστημα Διαχείρισης Βάσεων Δεδομένων (RDBMS) που χαρακτηρίζεται για την ευκολία στη χρήση του, καθώς και την υψηλή αξιοπιστία και ασφάλεια που προσφέρει. Χάρη σε αυτά τα στοιχεία, η MySQL μπορεί να χρησιμοποιηθεί κατά την ανάπτυξη διαφόρων τύπων λογισμικού, γνωρίζοντας ευρεία αποδοχή και ενσωμάτωση από τις εφαρμογές ιστού. Η λειτουργία του συγκεκριμένου συστήματος βασίζεται στην SQL, τη συνηθέστερα χρησιμοποιούμενη γλώσσα χειρισμού Σχεσιακών Βάσεων Δεδομένων [31].

Μερικά από τα κυριότερα χαρακτηριστικά που προσφέρει το σύστημα της MySQL είναι τα εξής:

Είναι ανοικτού κώδικα

Πρόκειται για ένα ελεύθερο και ανοικτού κώδικα σύστημα. Με αυτόν τον τρόπο, επιτρέπεται η άμεση χρήση του χωρίς την υποχρέωση απόκτησης συνδρομής, ενώ μπορεί να αξιοποιηθεί και από την πλευρά των προγραμματιστών, δίνοντας πρόσβαση στον πηγαίο κώδικα του λογισμικού και παρέχοντας τη δυνατότητα παραμετροποίησης με βάση τις εκάστοτε ανάγκες [31].

Υψηλή ταχύτητα

Η υψηλή ταχύτητα αποτελεί βασικό γνώρισμα της MySQL. Το σύστημά της έχει σχεδιαστεί και αναπτυχθεί με τέτοιο τρόπο, ώστε να επιτρέπει την εύκολη και γρήγορη διαχείριση μεγάλου όγκου δεδομένων, χωρίς να επιβαρύνει ιδιαίτερα τα υπολογιστικά συστήματα μέσω των οποίων εκτελείται. Το συγκεκριμένο γεγονός υπήρξε καθοριστικό, μιας και αποτέλεσε τη βάση για να καταστεί η MySQL ως ένα από τα υψηλότερα χρησιμοποιούμενα συστήματα RDBMS, βρίσκοντας εφαρμογή ακόμη και σε επαγγελματικές εφαρμογές μεγάλης κλίμακας [31].

Συμβατότητα

Η MySQL χαρακτηρίζεται επίσης για την υψηλή της συμβατότητα με ένα ευρύ φάσμα συστημάτων και εφαρμογών. Πιο συγκεκριμένα, έχει τη δυνατότητα υποστήριξης πλήθους διαφορετικών τύπων λειτουργικών συστημάτων και γλωσσών προγραμματισμού, διευκολύνοντας την υιοθέτησή της από μία μεγάλη ομάδα χρηστών. Επιπλέον, προσφέρονται διάφορα εργαλεία διαχείρισης μέσω γραφικής διεπαφής, καθιστώντας περισσότερο άνετο και γρήγορο τον χειρισμό μίας Βάσης Δεδομένων τύπου MySQL, από τους προγραμματιστές [31].

Όπως προαναφέρθηκε η MySQL, όπως και η πλειοψηφία των υπόλοιπων Σχεσιακών Συστημάτων Βάσεων Δεδομένων, αξιοποιεί για τη λειτουργία της, τη γλώσσα ερωτημάτων SQL. Μέσω της SQL, επιτρέπεται η αποθήκευση και ο χειρισμός δεδομένων στις σχεσιακές βάσεις, χρησιμοποιώντας τη δομή πινάκων. Εντός αυτών, κάθε γραμμή αντιπροσωπεύει μία εγγραφή, ενώ οι στήλες περιέχουν τα ονόματα των πεδίων που χαρακτηρίζουν τα δεδομένα. Μέσω κατάλληλων εντολών, όπως είναι οι `'SELECT'`, `'UPDATE'`, `'INSERT'` και `'DELETE'`, δίνεται η δυνατότητα για την αναζήτηση, ενημέρωση, καταχώριση και διαγραφή δεδομένων, αντίστοιχα [32].

Στο Σχήμα 3.3 φαίνεται ένα παράδειγμα σύνταξης κώδικα SQL.

```sql
create table if not exists model_class(
    id int not null,
    model_name varchar(50) not null,
    class_name varchar(50) not null,
    primary key(id,model_name),
    foreign key(id) references users(id) on delete cascade on update no action
);
select count(*) from users;
```

Σχήμα 3.3: Παράδειγμα σύνταξης κώδικα SQL

3.2.5 Python

Η Python είναι μία δημοφιλής γλώσσα προγραμματισμού γενικού σκοπού που αξιοποιείται στην ανάπτυξη διαφόρων τύπων λογισμικού, όπως είναι οι διαδικτυακές εφαρμογές και οι εφαρμογές Μηχανικής Μάθησης και Εξόρυξης Δεδομένων. Αποτελεί μία scripting γλώσσα, η οποία είναι ισχυρή και υψηλού επιπέδου, αλλά ταυτόχρονα είναι εύκολη στην εκμάθηση και κατανόηση, συμβάλλοντας με αυτόν τον τρόπο στην ευρεία διάδοσή της στην κοινότητα των προγραμματιστών [33, 34].

Τα κυριότερα χαρακτηριστικά που προσφέρει η Python στους προγραμματιστές είναι τα εξής:

Απλή σύνταξη

Ένα θεμελιώδες γνώρισμα της Python, είναι το γεγονός ότι έχει ιδιαίτερα απλή σύνταξη. Ειδικότερα, οι εντολές που αξιοποιεί έχουν εύκολη διατύπωση και αποτελούνται από συχνά χρησιμοποιούμενες λέξεις της αγγλικής γλώσσας. Με αυτόν τον τρόπο, διευκολύνεται η διαδικασία ανάπτυξης μίας εφαρμογής και επιταχύνεται η ολοκλήρωσή της. Παράλληλα, αυτό το γεγονός συνετέλεσε, ώστε η συγκεκριμένη γλώσσα να καταστεί αρκετά φιλική και να προσεγγίσει ακόμα και χρήστες με μικρή ή και καθόλου προηγούμενη εμπειρία, στον προγραμματισμό [34].

Είναι ανοικτού κώδικα

Η Python αποτελεί μία ανοικτού κώδικα γλώσσα προγραμματισμού. Συνεπώς, η αξιοποίησή της καθίσταται εύκολη και γρήγορη ακόμη και για εμπορικούς σκοπούς, ενώ οι χρήστες απαλλάσσονται από την πιθανότητα επιπρόσθετων εξόδων για την αγορά συνδρομής [34].

Είναι εύκολα επεκτάσιμη

Ένα ακόμη σημαντικό χαρακτηριστικό της Python, είναι και το γεγονός ότι αποτελεί μία γλώσσα προγραμματισμού, η οποία είναι εύκολα επεκτάσιμη, κάτι που απορρέει και από το γνώρισμα που αναφέρ-

θηκε προηγουμένως. Πιο συγκεκριμένα, αξίζει να τονιστεί η ύπαρξη μεγάλης κοινότητας προγραμματιστών που συμβάλλουν στη συνεχή βελτίωση και επέκτασή της, μέσα από την ανάπτυξη μεγάλου πλήθους βιβλιοθηκών κώδικα. Παράλληλα, παρέχεται πρόσβαση σε υποστηρικτικό υλικό, για την διευκόλυνση και την αποτελεσματικότερη καθοδήγηση των χρηστών τους [34].

Όσον αφορά τον τρόπο με τον οποίο συντάσσεται, θα πρέπει αρχικά να σημειωθεί ότι η Python, σε αντίθεση με άλλες δημοφιλείς γλώσσες προγραμματισμού, δεν διαθέτει κάποια εξειδικευμένη εντολή για τον καθορισμό των μεταβλητών, αλλά αντίθετα αυτές δημιουργούνται αφότου γίνει η ανάθεση μίας τιμής. Μία ακόμη σημαντική διαφοροποίηση, είναι το γεγονός ότι προκειμένου να επισημανθεί το τέλος μίας εντολής, δεν χρησιμοποιείται ο χαρακτήρας του ερωτηματικού (semicolon - `';'`), αλλά αντίθετα γίνεται αλλαγή γραμμής. Σχετικά με την ύπαρξη εσοχών στον κώδικα, αυτές έχουν σημαντικό ρόλο κατά τη συγγραφή ενός προγράμματος στην Python. Ειδικότερα, οι εσοχές χρησιμοποιούνται σε δομές όπως είναι οι `'if - else'`, `'for'` και `'while'` για την υπόδειξη του κώδικα που εσωκλείουν. Ολοκληρώνοντας, αξίζει να αναφερθεί ότι η λειτουργία της Python βασίζεται στη χρήση διερμηνευτή, γεγονός που επιτρέπει την άμεση εκτέλεση ενός προγράμματος μόλις ολοκληρωθεί η ανάπτυξή του [33].

Στο Σχήμα 3.4 φαίνεται ένα παράδειγμα σύνταξης κώδικα Python.

```python
import pandas as pd
import json

if className != None:
    classlabel = dataset[className]
    acc = metrics.accuracy_score(classlabel, pred_values)
    acc = round(acc,2)
data = []
data.append(columns)
for i in range(len(rows)):
    data.append(rows[i])
print(json.dumps({"dataset": data, "accuracy": acc}))
```

Σχήμα 3.4: Παράδειγμα σύνταξης κώδικα Python

3.2.6 Scikit-learn

Η Scikit-learn είναι μία δημοφιλής βιβλιοθήκη που χρησιμοποιείται μέσω της Python, για την εκτέλεση εργασιών Μηχανικής Μάθησης. Αξιοποιώντας τα χαρακτηριστικά της συγκεκριμένης γλώσσας, η βιβλιοθήκη παρέχει μέσω κατάλληλα σχεδιασμένης διεπαφής (API) ένα μεγάλο εύρος μεθόδων και λειτουργιών. Ειδικότερα, περιλαμβάνει μεθόδους κατηγοριοποίησης, συσταδοποίησης και παλινδρόμησης, ενώ υποστηρίζει λειτουργίες όπως είναι η εκπαίδευση μοντέλων, η αξιολόγηση της απόδοσής τους ή η προεπεξεργασία δεδομένων [35].

Τα παραπάνω χαρακτηριστικά, συμβάλλουν στην ευρύτερη προώθηση του πεδίου της Μηχανικής Μάθησης, καθώς και στη διευκόλυνση της υιοθέτησης από μία μεγάλη ομάδα χρηστών, των μεθόδων που αυτή χρησιμοποιεί. Σε αυτό συντελεί και το γεγονός, ότι πρόκειται για μία βιβλιοθήκη ανοικτού κώδικα. Συνεπώς, παρέχεται πρόσβαση στον πηγαίο κώδικα του λογισμικού, κάτι που δίνει τη δυνατότητα στους προγραμματιστές να συμβάλλουν στη βελτίωση και επέκτασή του. Παράλληλα, επιτρέπεται η ελεύθερη

χρήση τόσο για εκπαιδευτικούς, όσο και για εμπορικούς σκοπούς [35].

Στο Σχήμα 3.5 που ακολουθεί, φαίνεται ένα παράδειγμα εκπαίδευσης κατηγοριοποιητή Δέντρων Απόφασης με χρήση της Python και του Scikit-learn. Αξίζει να σημειωθεί ότι η εν λόγω βιβλιοθήκη προσφέρει για τα Δέντρα Απόφασης μία υλοποίηση που βασίζεται στον αλγόριθμο CART, ωστόσο δεν υποστηρίζει τη χρήση ονομαστικών δεδομένων, γεγονός που θα αναλυθεί εκτενέστερα στο επόμενο Κεφάλαιο.

```
import pandas as pd
from sklearn.tree import DecisionTreeClassifier

train_filename = 'C:/Users/user/Documents/Datasets/iris-train.csv'
data_train = pd.read_csv(train_filename)
attr_train = data_train[["sepal.length","petal.length"]]
class_train = data_train["variety"]
model = DecisionTreeClassifier()
model.fit(attr_train, class_train)
```

Σχήμα 3.5: Παράδειγμα χρήσης της βιβλιοθήκης Scikit-learn

3.3 Front-end

3.3.1 HTML

Η HTML είναι μία γλώσσα σήμανσης υπερκειμένου (Hypertext Markup Language), η οποία αποτελεί θεμελιώδες στοιχείο του παγκόσμιου ιστού και χρησιμοποιείται για τον καθορισμό της δομής και του περιεχομένου των ιστοσελίδων. Για να επιτύχει κάτι τέτοιο, η HTML αξιοποιεί ένα σύνολο από στοιχεία (HTML elements) που οργανώνουν και ομαδοποιούν το περιεχόμενο, ενώ επίσης είναι ικανά να εφαρμόζουν μορφοποιήσεις και να καθορίζουν τον τρόπο, με τον οποίο συμπεριφέρονται τα διάφορα τμήματα της ιστοσελίδας [36].

Όσον αφορά τον τρόπο με τον οποίο συντάσσεται, η HTML χρησιμοποιεί τις χαρακτηριστικές τριγωνικές αγκύλες `'<'` και `'>'`, εντός των οποίων περικλείονται οι ονομασίες που καθορίζουν το είδος των στοιχείων. Το προαναφερθέν, συνολικά αποκαλείται και ως ετικέτα (HTML tag). Παραδείγματα ετικετών, αποτελούν τα `'<ul>'` για τον καθορισμό μίας μη αριθμημένης λίστας, `'<img>'` για τις εικόνες και `'<p>'` για τον καθορισμό μίας παραγράφου. Πέρα από τις ονομασίες, εντός των ετικετών τοποθετούνται και χαρακτηριστικά (attributes), για τον καθορισμό περαιτέρω ιδιοτήτων. Επιπρόσθετα, ανάλογα με το είδος του στοιχείου, οι ετικέτες χρησιμοποιούνται με τέτοιο τρόπο, ώστε να υποδεικνύουν την έναρξη και την ολοκλήρωση του, αντίστοιχα [36]. Στο Σχήμα 3.6 φαίνεται ένα εκτενέστερο παράδειγμα του τρόπου, με τον οποίο συντάσσεται ο κώδικας της HTML.

```
<div class="col-12 col-lg-6">
  <h3>Classify Data using Pretrained Models</h3>
  <p>Upload unclassified datasets and predict their labels.</p>
</div>
<div class="col-12 col-lg-6">
  <img class="img-fluid" src="./src/img/predicted_data.jpg" alt="Predicted Data">
</div>
```

Σχήμα 3.6: Παράδειγμα σύνταξης κώδικα HTML

3.3.2 CSS

Μία ακόμη βασική τεχνολογία του παγκόσμιου ιστού, αποτελεί και η CSS (Cascading Style Sheets). Πρόκειται για μία γλώσσα που χρησιμοποιείται για τον καθορισμό του τρόπου, με τον οποίο εμφανίζεται το περιεχόμενο ενός εγγράφου HTML. Για αυτόν το λόγο, χρησιμοποιεί ένα σύνολο από κανόνες τους οποίους εφαρμόζει, επιλέγοντας κάθε φορά το κατάλληλο στοιχείο της HTML [37].

Όσον αφορά τον τρόπο σύνταξης των κανόνων CSS, το αρχικό βήμα σχετίζεται με τον καθορισμό του επιλογέα (selector). Τα είδη των επιλογέων ποικίλλουν, καθώς πέρα από την ονομασία του στοιχείου HTML, είναι εφικτή και η χρήση ορισμένων, περισσότερο εξειδικευμένων. Ειδκότερα, ο χρήστης μπορεί να επιλέξει ανάμεσα σε επιλογείς με βάση το μοναδικό αναγνωριστικό (ID selectors), επιλογείς κλάσεων (class selectors) ή ψευδοκλάσεων (pseudo-class selectors), καθώς και επιλογείς με βάση τα χαρακτηριστικά (attribute selectors). Στη συνέχεια, η σύνταξη των κανόνων CSS είναι εφικτή, μέσα από κατάλληλες εντολές. Αυτές περιέχουν την ιδιότητα που επιθυμεί ο χρήστης να τροποποιήσει, συνοδευόμενη από την αντίστοιχη τιμή. Θα πρέπει επίσης να σημειωθεί, ότι τα ζεύγη ιδιοτήτων και τιμών χωρίζονται με την άνω και κάτω τελεία (`':'`), οι κανόνες χωρίζονται μεταξύ τους με τον χαρακτήρα του semicolon (`';'`), ενώ το σύνολο των κανόνων που ανήκουν σε έναν επιλογέα, ομαδοποιείται με τη χρήση αγκυλών (`'{ ... }'`) [37].

Ένα εκτενέστερο παράδειγμα του τρόπου με τον οποίο συντάσσεται ο κώδικας της CSS, φαίνεται στο Σχήμα 3.7.

```css
.field-text{
    color: #898989;
    font-size: 17px;
    text-align: center;
}
.page-subtitle h5 a:hover{
    text-decoration: underline;
}
#results_tableDiv{
    overflow-x: auto;
    max-height: 500px;
}
```

Σχήμα 3.7: Παράδειγμα σύνταξης κώδικα CSS

3.3.3 Bootstrap

Μία ακόμη τεχνολογία που αξιοποιήθηκε κατά την ανάπτυξη της εφαρμογής, είναι και το Bootstrap. Το Bootstrap, αποτελεί ένα πλαίσιο (framework) που χρησιμοποιείται κατά την ανάπτυξη του Front-end τμήματος μίας διαδικτυακής εφαρμογής και προσφέρει μεγάλο πλήθος εργαλείων για τον σχεδιασμό γραφικών διεπαφών. Πρόκειται για ένα ελεύθερο και ανοικτού κώδικα framework, η δομή του οποίου βασίζεται στις γλώσσες HTML, CSS και JavaScript [38].

Το Bootstrap στοχεύει στο να διευκολύνει τους προγραμματιστές κατά τη διαδικασία δημιουργίας της γραφικής διεπαφής, μέσα από την διάθεση μίας σειράς από έτοιμα συστατικά και λειτουργίες. Μερικά παραδείγματα είναι το προκαθορισμένο σύστημα πλέγματος, τα διαφόρων ειδών κουμπιά και οι γραμμές

πλοήγησης. Κοινό χαρακτηριστικό όλων, αποτελεί το γεγονός ότι έχουν σχεδιαστεί με τέτοιο τρόπο, ώστε να ανταποκρίνονται στους κανόνες και τις αρχές του responsive web design. Κάτι τέτοιο σημαίνει, ότι κάθε στοιχείο που παρέχεται από το Bootstrap, έχει την ικανότητα να προσαρμόζει τον τρόπο με τον οποίο παρουσιάζεται και διατάσσεται ανάλογα με τις διαστάσεις της οθόνης του χρήστη [38].

3.3.4 JavaScript

Η JavaScript αποτελεί μία από τις ευρύτερα χρησιμοποιούμενες γλώσσες προγραμματισμού που αρχικά δημιουργήθηκε, με σκοπό την αποκλειστική χρήση της στην ανάπτυξη του Front-end τμήματος, των διαδικτυακών εφαρμογών. Ωστόσο, χάρη στην δημιουργία και υιοθέτηση μεγάλου πλήθους βιβλιοθηκών και frameworks, κατέστη εφικτή η επέκταση του εύρους των δυνατοτήτων της. Συνεπώς, η JavaScript έχει πλέον την ικανότητα αποτελεσματικής κάλυψης τομέων, όπως είναι ο προγραμματισμός του Back-end και η ανάπτυξη εφαρμογών για κινητές συσκευές, με ορισμένα χαρακτηριστικά παραδείγματα να αποτελούν το περιβάλλον Node.js και το framework του React Native, αντίστοιχα [39].

Στα πλαίσια της παρούσας πτυχιακής εργασίας, η συγκεκριμένη γλώσσα χρησιμοποιήθηκε για την ανάπτυξη του Front-end της εφαρμογής. Στον συγκεκριμένο τομέα, η JavaScript αξιοποιείται από κοινού με τις HTML και CSS, για να μετατρέψει τις απλές ιστοσελίδες σε διαδραστικές εφαρμογές. Κάτι τέτοιο σημαίνει, ότι η δομή και η εμφάνιση μίας σελίδας μπορεί να μεταβάλλεται ζωντανά, ανάλογα με τις ενέργειες ενός χρήστη, χωρίς να είναι απαραίτητη η επαναφόρτωση της σελίδας [39].

3.3.5 jQuery

Όσον αφορά τη jQuery, πρόκειται για μία βιβλιοθήκη (library) της JavaScript, η οποία αναπτύχθηκε με σκοπό την απλοποίηση των διαδικασιών και λειτουργιών που ακολουθεί η συγκεκριμένη γλώσσα προγραμματισμού. Η jQuery αποτελεί μία ιδιαίτερα δημοφιλή και ανοικτού κώδικα βιβλιοθήκη που είναι εύκολη στην κατανόηση και εκμάθηση, ακόμη και από άτομα με μικρή εμπειρία. Σε αυτό συμβάλλει και το γεγονός της ύπαρξης μίας ευρείας κοινότητας προγραμματιστών που την χρησιμοποιεί, παρέχοντας παράλληλα μεγάλο πλήθος υποστηρικτικού υλικού [40].

Το κύριο χαρακτηριστικό της jQuery, είναι ότι επιτρέπει την εκτέλεση ορισμένων βασικών λειτουργιών της JavaScript, όπως είναι ο χειρισμός του DOM (Document Object Model), η διαχείριση των συμβάντων (event handling), καθώς και η εκτέλεση αιτημάτων HTTP, με τη χρήση απλοποιημένης σύνταξης. Επιπρόσθετα, παρέχει ενσωματωμένες μεθόδους για τον αποτελεσματικότερο καθορισμό του τρόπου εμφάνισης, καθώς και της μορφοποίησης τμημάτων της ιστοσελίδας. Για να επιτύχει κάτι τέτοιο, χειρίζεται τα στοιχεία της HTML αξιοποιώντας επιλογείς, όμοιους με αυτούς της CSS [40].

Στο Σχήμα 3.8 που ακολουθεί, φαίνεται ένα παράδειγμα του τρόπου με τον οποίο συντάσσεται ο κώδικας της JavaScript, με χρήση της βιβλιοθήκης jQuery.

```
if(sessionStorage.getItem("token") !== null){
    $("#loginb").hide();
    $("#regbtn").hide();
}
$("#logoutb").click(function(){
    sessionStorage.clear();
    window.location.href = './';
});
var input_key = $("input");
input_key.on("keypress", function(event){
    if(event.key === "Enter"){
        event.preventDefault();
        $("#loginb").click();
    }
});
```

Σχήμα 3.8: Παράδειγμα σύνταξης κώδικα JavaScript - jQuery

3.4 Επίλογος

Ολοκληρώνοντας το Κεφάλαιο 3, στα πλαίσιά του παρουσιάστηκαν οι γλώσσες προγραμματισμού, καθώς και οι τεχνολογίες που αξιοποιήθηκαν κατά την ανάπτυξη της εφαρμογής. Έγινε εκτενής αναφορά σε αυτές, τόσο σε επίπεδο back-end, όσο και σε αυτό του front-end, σημειώνοντας στοιχεία όπως τα χαρακτηριστικά του τρόπου λειτουργίας τους, τα πλεονεκτήματα και μειονεκτήματά τους, καθώς και παραδείγματα σύνταξης και χρήσης του κώδικά τους.

Κεφάλαιο 4ο: Σχεδίαση και Υλοποίηση του AutoDTrees

4.1 Λειτουργικές απαιτήσεις

Οι λειτουργικές απαιτήσεις, χρησιμοποιούνται για τον λεπτομερή καθορισμό των δυνατοτήτων ενός συστήματος λογισμικού, καθώς και του τρόπου με τον οποίο αυτό ανταποκρίνεται σε ορισμένες απαιτήσεις, που τίθενται από τους χρήστες.

Όσον αφορά την εφαρμογή AutoDTrees, οι λειτουργικές της απαιτήσεις είναι οι ακόλουθες:

Δημιουργία λογαριασμού χρήστη
Κάθε χρήστης θα μπορεί να εγγραφεί στην εφαρμογή, χρησιμοποιώντας μία σελίδα που θα δημιουργηθεί για αυτόν τον σκοπό. Εντός της σελίδας θα περιλαμβάνεται μία φόρμα, όπου ο χρήστης θα πρέπει να συμπληρώσει τα εξής προσωπικά του στοιχεία: Όνομα, Επώνυμο, Διεύθυνση Email, καθώς και τον επιθυμητό κωδικό πρόσβασης. Ακολούθως, προκειμένου να ολοκληρωθεί η εγγραφή, ο χρήστης θα μπορεί να επιβεβαιώσει τα στοιχεία του, μέσω σχετικού ηλεκτρονικού μηνύματος που θα του αποσταλλεί.

Είσοδος χρήστη στην εφαρμογή
Κάθε χρήστης θα μπορεί να πραγματοποιήσει είσοδο στην εφαρμογή, χρησιμοποιώντας τη διεύθυνση Email και τον κωδικό πρόσβασης που δήλωσε κατά την εγγραφή του. Η συγκεκριμένη λειτουργική απαίτηση είναι σημαντική, καθώς θα δίνει τη δυνατότητα στους χρήστες να μπορούν να αποθηκεύουν διάφορα στοιχεία και αρχεία που σχετίζονται με την εφαρμογή.

Επαναφορά κωδικού πρόσβασης
Κάθε χρήστης θα μπορεί να επαναφέρει τον λογαριασμό του μετά από απώλεια του κωδικού πρόσβασης. Για να επιτευχθεί κάτι τέτοιο, θα ακολουθεί διαδικασία επιβεβαίωσης μέσω της διεύθυνσης Email που δήλωσε κατά την εγγραφή του.

Επεξεργασία στοιχείων χρήστη
Θα δίνεται η δυνατότητα σε κάθε χρήστη, να μπορεί να τροποποιήσει τα στοιχεία που εισήγαγε κατά την εγγραφή του στην εφαρμογή. Στην περίπτωση αλλαγής της διεύθυνσης Email, θα ζητείται εκ νέου επιβεβαίωση μέσα από την αποστολή σχετικού ηλεκτρονικού μηνύματος, ενώ η αλλαγή του κωδικού πρόσβασης θα επιτρέπεται κατόπιν επιβεβαίωσης του τρέχοντος κωδικού.

Διαγραφή λογαριασμού χρήστη
Θα δίνεται η δυνατότητα σε κάθε χρήστη, να μπορεί να διαγράψει τον λογαριασμό του. Για τον σκοπό αυτό, θα ζητείται επιβεβαίωση μέσα από την εκ νέου εισαγωγή του κωδικού πρόσβασης.

Σελίδες πληροφόρησης
Θα δημιουργηθούν δύο σελίδες πληροφοριακού περιεχομένου που θα είναι προσβάσιμες ακόμη και από μη εγγεγραμμένους χρήστες. Ειδικότερα, η «Αρχική Σελίδα» θα καλωσορίζει τον χρήστη και θα τον εισάγει στις βασικές λειτουργίες που προσφέρει η εφαρμογή, ενώ η σελίδα «Σχετικά» θα περιλαμβάνει γενικές πληροφορίες σχετικά με την εκπόνηση της εργασίας.

Σελίδες εκτέλεσης του αλγορίθμου
Οι εργασίες που αφορούν τον αλγόριθμο των Δέντρων Απόφασης, θα ομαδοποιηθούν σε δύο σελίδες

που θα δημιουργηθούν για αυτόν το σκοπό και θα είναι προσβάσιμες μόνο σε εγγεγραμμένους χρήστες. Η πρώτη θα αφορά τις εργασίες δημιουργίας νέων μοντέλων και η δεύτερη σελίδα τις εργασίες με προεκπαιδευμένα μοντέλα.

Μεταφόρτωση αρχείου

Κάθε χρήστης θα μπορεί να προχωρήσει στην μεταφόρτωση αρχείων που περιέχουν σύνολα δεδομένων. Αυτά θα μπορούν, ανάλογα με την περίπτωση να χρησιμοποιηθούν είτε για την εκπαίδευση ενός μοντέλου, είτε για την κατηγοριοποίηση των στιγμιοτύπων τους. Σε ορισμένες περιπτώσεις, θα δίνεται επιπλέον η επιλογή να είναι δημόσια, δηλαδή διαθέσιμα σε κάθε χρήστη, κατόπιν παροχής σχετικής άδειας από τον διαχειριστή. Επιπλέον, θα επιτρέπεται η μεταφόρτωση μόνο αρχείων τύπου `.csv`.

Προεπισκόπηση αρχείου

Για κάθε σύνολο δεδομένων που επιλέγει ο χρήστης, θα έχει τη δυνατότητα προεπισκόπησης τμήματος του περιεχομένου του, με τη μορφή πίνακα. Κάτι τέτοιο, θα διευκολύνει τον χρήστη στην κατανόηση της δομής των δεδομένων που προτίθεται να χρησιμοποιήσει, προτού εκτελέσει τον αλγόριθμο.

Διαγραφή αρχείου

Κάθε χρήστης θα έχει την επιλογή να διαγράψει το επιθυμητό σύνολο δεδομένων. Η διαγραφή δημόσιων αρχείων θα επιτρέπεται, μόνο εφόσον ο χρήστης διαθέτει τη σχετική άδεια από τον διαχειριστή.

Δημιουργία μοντέλου κατηγοριοποίησης

Κάθε χρήστης θα μπορεί να προχωρήσει στη δημιουργία μοντέλων Δέντρων Απόφασης, χρησιμοποιώντας το επιλεγμένο σύνολο δεδομένων εκπαίδευσης. Επιπλέον, θα έχει τη δυνατότητα επιλογής, μέσω σχετικής λίστας, τόσο των πεδίων του συνόλου που θα χρησιμοποιηθούν ως χαρακτηριστικά (features), όσο και του πεδίου που θα χρησιμοποιηθεί ως γνώρισμα κλάσης. Ο χρήστης θα μπορεί κατόπιν, να ορίσει τις παραμέτρους του μέγιστου βάθους (`max_depth`) και του ελάχιστου αριθμού δειγμάτων ανά φύλλο (`min_samples_leaf`) του κατηγοριοποιητή ή να χρησιμοποιήσει τις προκαθορισμένες τιμές, προς διευκόλυνσή του.

Αξιολόγηση της απόδοσης του μοντέλου

Για κάθε μοντέλο που δημιουργείται, θα εκτελείται η μέθοδος k-fold cross-validation για την αξιολόγηση της απόδοσής του. Για αυτόν το σκοπό ο χρήστης θα εισάγει την παράμετρο `k`. Στη συνέχεια, θα έχει τη δυνατότητα προβολής των μετρικών απόδοσης, τόσο για κάθε τιμή της κλάσης (label), όσο και για το σύνολο του μοντέλου.

Αποθήκευση μοντέλου

Κάθε χρήστης θα μπορεί εφόσον το επιθυμεί, μετά το πέρας της αξιολόγησης ενός μοντέλου, να το αποθηκεύσει στον προσωπικό του λογαριασμό, για μελλοντική αξιοποίησή του. Για αυτόν το σκοπό, θα ζητείται από τον χρήστη η εισαγωγή ενός ονόματος για το μοντέλο.

Προεπισκόπηση και οπτικοποίηση μοντέλου

Για κάθε μοντέλο που επιλέγει ο χρήστης, θα έχει τη δυνατότητα προεπισκόπησης του περιεχομένου του, δηλαδή τα features και το class από τα οποία αποτελείται. Επιπλέον, θα μπορεί να προχωρήσει στην οπτικοποίηση του Δέντρου Απόφασης, κάτι που θα τον διευκολύνει στην κατανόηση της δομής και του τρόπου εξαγωγής συμπερασμάτων του μοντέλου.

Διαγραφή μοντέλου
Κάθε χρήστης, θα έχει την επιλογή να διαγράψει από τον λογαριασμό του, ένα αποθηκευμένο μοντέλο που επιθυμεί.

Κατηγοριοποίηση νέων στιγμιοτύπων
Κάθε χρήστης, αξιοποιώντας ένα προεκπαιδευμένο μοντέλο, θα μπορεί να προχωρήσει στην κατηγοριοποίηση νέων στιγμιοτύπων, τα οποία θα παρέχονται μέσα από το σχετικό σύνολο δεδομένων που θα επιλέξει. Στη συνέχεια, θα προβάλλεται ένας πίνακας που θα περιλαμβάνει τα δεδομένα, μαζί με τα αποτελέσματα της κατηγοριοποίησης. Λόγω του όγκου του συνόλου δεδομένων, θα προβάλλεται ένα δείγμα των εγγραφών του και ακολούθως θα δίνεται η δυνατότητα στον χρήστη να εξάγει το πλήρες dataset σε μορφή `.csv`. Επιπρόσθετα, θα δίνεται η επιλογή προβολής μετρικών, για την εκτίμηση της ποιότητας των αποτελεσμάτων. Αυτή η δυνατότητα, θα είναι διαθέσιμη μόνο σε περιπτώσεις, όπου το επιλεγμένο dataset περιέχει ήδη το πεδίο της κλάσης.

Δημόσιο Web API και σελίδα τεκμηρίωσής του
Κάθε ενδιαφερόμενος προγραμματιστής, θα πρέπει να μπορεί να αποκτήσει πρόσβαση στις προαναφερθείσες λειτουργίες της εφαρμογής ελεύθερα, μέσω ενός δημόσιου Web API. Προς διευκόλυνσή τους, θα δημιουργηθεί και μία σελίδα τεκμηρίωσης του API, μέσα από την οποία θα παρέχονται οδηγίες για τη χρήση του, καθώς και παραδείγματα σύνταξης για κάθε ένα endpoint που αντιστοιχεί σε μία λειτουργία.

4.2 Αρχιτεκτονική του AutoDTrees

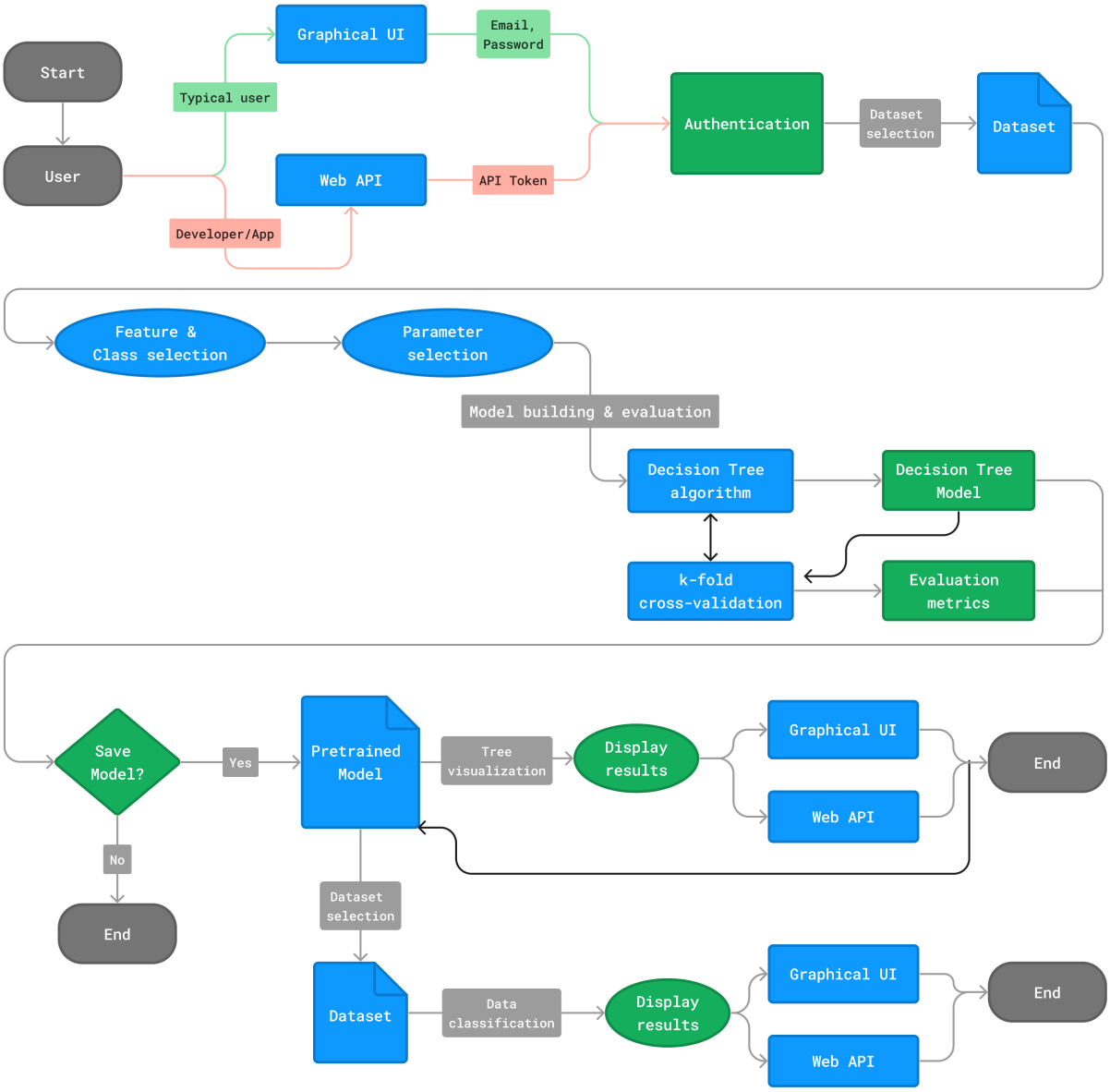
Όπως έχει ήδη σημειωθεί, ο στόχος της διαδικτυακής εφαρμογής AutoDTrees είναι η παροχή ενός ολοκληρωμένου περιβάλλοντος που θα επιτρέπει την κατηγοριοποίηση δεδομένων, με χρήση του αλγορίθμου των Δέντρων Αποφάσης. Ειδικότερα, το AutoDTrees θα δίνει τη δυνατότητα στους χρήστες κατόπιν αυθεντικοποίησης, να επιλέξουν αρχικά τα επιθυμητά σύνολα δεδομένων εκπαίδευσης. Ακολουθεί το στάδιο του καθορισμού, τόσο των πεδίων του συνόλου που θα χρησιμοποιηθούν ως χαρακτηριστικά, όσο και του πεδίου που θα χρησιμοποιηθεί ως γνώρισμα κλάσης. Ο χρήστης θα μπορεί επιπλέον, να ορίσει τις παραμέτρους του κατηγοριοποιητή `'max_depth'` και `'min_samples_leaf'`, καθώς και την παράμετρο `'k'` της μεθόδου k-fold cross-validation. Στη συνέχεια, θα εκτελείται ο αλγόριθμος των Δέντρων Απόφασης για τη δημιουργία του μοντέλου και θα αξιολογείται η αποτελεσματικότητά του, μέσω της τεχνικής k-fold cross-validation, παρέχοντας τα αποτελέσματα των μετρικών.

Η εφαρμογή θα επιτρέπει επίσης την αποθήκευση των μοντέλων, ώστε οι χρήστες να μπορούν να τα αξιοποιήσουν μελλοντικά, είτε για την οπτικοποίηση του Δέντρου Απόφασης, είτε για την πρόβλεψη μη κατηγοριοποιημένων στιγμιοτύπων. Στην πρώτη περίπτωση, θα δίνεται η δυνατότητα προβολής του διαγράμματος, καθώς και εξαγωγής του σε αρχείο `.png`. Στη δεύτερη περίπτωση, θα πρέπει αρχικά ο χρήστης να επιλέξει το επιθυμητό μοντέλο, όπως και το σύνολο με τα μη κατηγοριοποιημένα στιγμιότυπα. Ακολούθως, θα εκτελείται η κατηγοριοποίηση και θα εμφανίζονται τα αποτελέσματα, επιτρέποντας την εξαγωγή τους σε αρχείο `.csv`. Επιπρόσθετα, θα δίνεται η επιλογή προβολής μετρικών για την εκτίμηση της ποιότητας των αποτελεσμάτων, με τη δυνατότητα αυτή να είναι διαθέσιμη, εφόσον το επιλεγμένο dataset περιέχει ήδη το πεδίο της κλάσης.

Το σύνολο των προαναφερθέντων λειτουργιών θα είναι προσπελάσιμο, τόσο μέσω γραφικής διεπαφής,

φιλικής προς τον χρήστη, όσο και μέσω ενός ελεύθερου διαδικτυακού API. Η διαφοροποίηση μεταξύ αυτών των δύο, έγκειται στον τρόπο ταυτοποίησης των χρηστών και φυσικά στον τρόπο εμφάνισης των αποτελεσμάτων. Ειδικότερα, για την ταυτοποίηση μέσω της γραφικής διεπαφής οι χρήστες θα εισάγουν τη διεύθυνση Email και τον κωδικό πρόσβασης που όρισαν κατά την εγγραφή τους, ενώ για το Web API θα εισάγουν το προσωπικό τους API Token, που δημιουργήθηκε αυτόματα κατά την εγγραφή τους. Όσον αφορά τον τρόπο εμφάνισης, σε περίπτωση χρήσης του Web API, τα αποτελέσματα παρουσιάζονται σε μορφή `JSON`.

Το Σχήμα 4.1 προβάλλει συγκεντρωτικά τα παραπάνω, με τη μορφή διαγράμματος ροής.



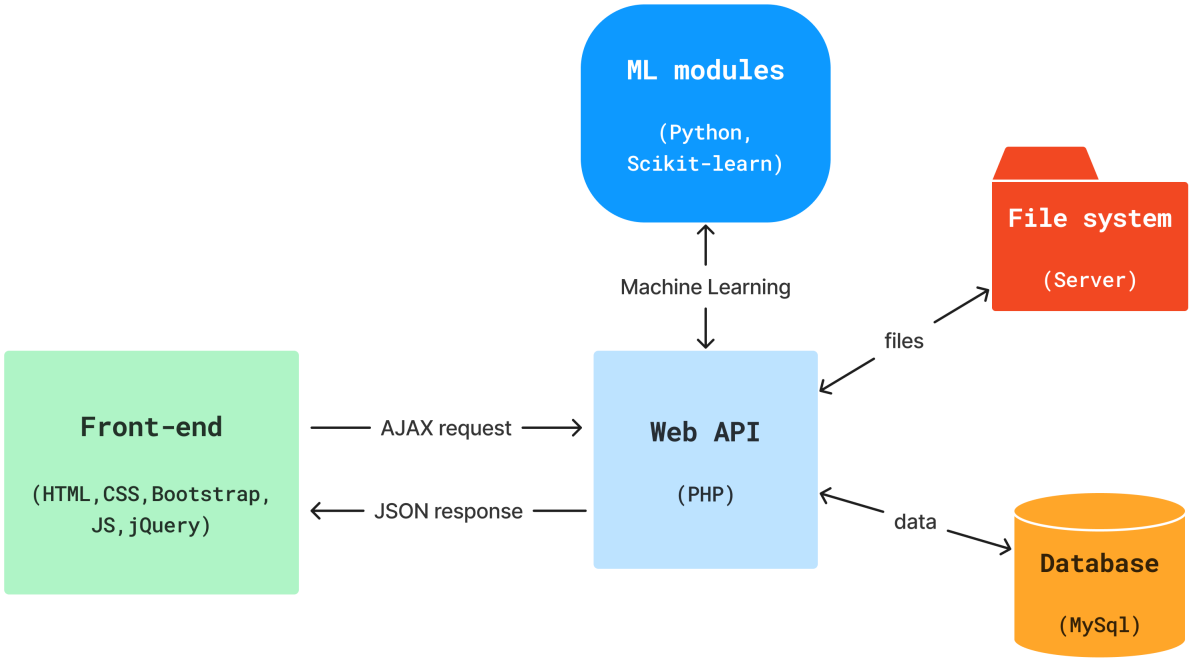
Σχήμα 4.1: Το διάγραμμα ροής του AutoDTrees

Η αρχιτεκτονική του AutoDTrees περιλαμβάνει μία δομή, αποτελούμενη από πέντε κύρια τμήματα. Την κινητήριο δύναμη της εφαρμογής αποτελεί το Web API, που έχει αναπτυχθεί με τη γλώσσα PHP. Μέσω αυτού, είναι εφικτή η επικοινωνία με το Back-end και δρομολογείται η εκτέλεση της πλειοψηφίας των λειτουργιών. Όποτε είναι απαραίτητο, ενεργεί σε συνεργασία με τα υπόλοιπα τμήματα της εφαρμογής.

Συνεπώς, σε ό,τι αφορά την Βάση Δεδομένων MySQL, αυτή αξιοποιείται κυρίως σε λειτουργίες όπως είναι η εγγραφή, η αυθεντικοποίηση και η διαχείριση των χρηστών. Ωστόσο, σε ορισμένες συνθήκες χρησιμοποιείται υποστηρικτικά, για την κάλυψη ενεργειών που απαιτούνται κατά την αποθήκευση ενός προεκπαιδευμένου μοντέλου. Ένα ακόμη τμήμα με το οποίο επικοινωνεί το Web API, είναι και το σύστημα αρχείων (file system) του server, για την αποθήκευση των αρχείων του χρήστη, όπως είναι τα σύνολα δεδομένων και τα μοντέλα.

Επιπρόσθετα, για την εκτέλεση του αλγορίθμου των Δέντρων Απόφασης, καθώς και των πάσης φύσεως εργασιών Μηχανικής Μάθησης, το Web API προχωράει στην κλήση των σχετικών modules που έχουν αναπτυχθεί σε γλώσσα Python και τη βιβλιοθήκη Scikit-learn. Τελικά, το Web API επικοινωνεί με τη γραφική διεπαφή, ώστε κάθε χρήστης να μπορεί να προσπελάσει τις λειτουργίες της εφαρμογής, μέσα από ένα φιλικότερο προς αυτόν περιβάλλον. Ο σχεδιασμός και η ανάπτυξη της γραφικής διεπαφής έγινε με τις HTML, CSS, Bootstrap και jQuery, όπως σημειώθηκε και στο προηγούμενο κεφάλαιο.

Συγκεντρωτικά, η αρχιτεκτονική του AutoDTrees παρουσιάζεται μέσω του Σχήματος 4.2 που ακολουθεί.



Σχήμα 4.2: Η αρχιτεκτονική του AutoDTrees

4.3 Χρήστες, δημόσια και ιδιωτικά σύνολα δεδομένων

Στην εφαρμογή AutoDTrees εφαρμόζεται η λογική της ύπαρξης δύο τύπων συνόλων δεδομένων, ανάλογα με το κοινό που έχει πρόσβασή σε αυτά. Έτσι, ο πρώτος τύπος αφορά τα ιδιωτικά σύνολα, τα οποία κάθε εγγεγραμμένος χρήστης μπορεί να μεταφορτώσει στον φάκελο του προσωπικού του λογαριασμού και να τα χρησιμοποιήσει ελεύθερα, χωρίς ωστόσο να μπορεί να τα μοιραστεί με άλλους χρήστες της εφαρμογής.

Αντίθετα, ο δεύτερος τύπος αφορά την ύπαρξη δημόσιων συνόλων δεδομένων. Κάτι τέτοιο σημαίνει, ότι ένας χρήστης μπορεί να μεταφορτώσει αρχεία συνόλων στον σχετικό φάκελο, τα οποία ακολούθως θα είναι διαθέσιμα, τόσο για προβολή, όσο και για χρήση, σε όλους τους εγγεγραμμένους χρήστες της εφαρ-

μογής. Η εν λόγω δυνατότητα στοχεύει γενικότερα στη διευκόλυνση των χρηστών στο να μπορούν να προχωρούν στη δημιουργία αποτελεσματικών μοντέλων Δέντρων Απόφασης, αλλά και σε περιπτώσεις όπως είναι ο διαμοιρασμός συνόλων δεδομένων μεταξύ καθηγητών και φοιτητών στα πλαίσια εργαστηριακών ασκήσεων. Θα πρέπει σε αυτό το σημείο να τονιστεί, ότι η επιλογή των δημόσιων συνόλων δεδομένων προσφέρεται μόνο κατά τη λειτουργία της εκπαίδευσης νέων μοντέλων. Στην περίπτωση της κατηγοριοποίησης νέων στιγμιοτύπων, η επιλογή ενός σχετικού συνόλου δεδομένων είναι στην ευχέρεια του χρήστη και είναι ορατό μόνο σε προσωπικό επίπεδο.

Για λόγους ασφαλείας το δικαίωμα μεταφόρτωσης και διαγραφής δημόσιων συνόλων δεδομένων, περιορίζεται μόνο σε χρήστες που τους έχει δοθεί σχετική άδεια από τον διαχειριστή της εφαρμογής. Συνεπώς, ανάλογα με το εύρος των δικαιωμάτων τους, οι χρήστες κατατάσσονται στα ακόλουθα τέσσερα είδη:

- **Διαχειριστής**
  Ο συγκεκριμένος χρήστης, είναι ο διαχειριστής του συστήματος, έχοντας πλήρη δικαιώματα στον χειρισμό των δημόσιων συνόλων δεδομένων. Επιπλέον, μπορεί να καθορίζει μέσω της Βάσης Δεδομένων, ποιοι άλλοι χρήστες θα επιτρέπεται να έχουν δημόσια δικαιώματα.
- **Εγγεγραμμένοι με δημόσια δικαιώματα**
  Το συγκεκριμένο είδος χρηστών διαθέτει δικαιώματα χειρισμού δημόσιων συνόλων δεδομένων, τα οποία παραχωρήθηκαν από τον διαχειριστή, μετά την εγγραφή τους στην εφαρμογή.
- **Εγγεγραμμένοι χωρίς δημόσια δικαιώματα**
  Το συγκεκριμένο είδος χρηστών, έχει εγγραφεί και μπορεί να αξιοποιήσει πλήρως την εφαρμογή. Ωστόσο, δεν επιτρέπεται σε αυτούς να διαγράφουν ή να ανεβάζουν νέα δημόσια σύνολα δεδομένων.
- **Μη εγγεγραμμένοι**
  Το συγκεκριμένο είδος χρηστών δεν έχει εγγραφεί και εφόσον δεν το πράξει στη συνέχεια, δεν θα μπορεί να την αξιοποιήσει.

4.4 Υλοποίηση του Back-end

4.4.1 Βάση Δεδομένων

Όσον αφορά τη Βάση Δεδομένων τύπου MySQL της εφαρμογής, αυτή αξιοποιείται κυρίως σε λειτουργίες όπως είναι η εγγραφή, η αυθεντικοποίηση και η διαχείριση των χρηστών, ενώ όπως θα αναλυθεί στη συνέχεια, σε ορισμένες περιπτώσεις έχει υποστηρικτικό ρόλο, για την κάλυψη ενεργειών που απαιτούνται κατά την αποθήκευση ενός προεκπαιδευμένου μοντέλου.

Τα δεδομένα που χρησιμοποιήθηκαν κατανέμονται σε τρεις πίνακες, τους `'users'`, `'verify_account'` και `'model_class'`.

Ο πίνακας `'users'`, όπως φαίνεται και από το Σχήμα 4.3, περιλαμβάνει κυρίως, στοιχεία που σχετίζονται με την ταυτοποίηση των χρηστών και αποτελείται από οκτώ πεδία. Ειδικότερα:

- Το πεδίο `'id'`, λειτουργεί ως μοναδικό αναγνωριστικό του κάθε χρήστη. Δημιουργείται και εισάγεται με αυτόματο τρόπο, μέσω της εντολής `'auto_increment'` και αποτελεί το κύριο κλειδί

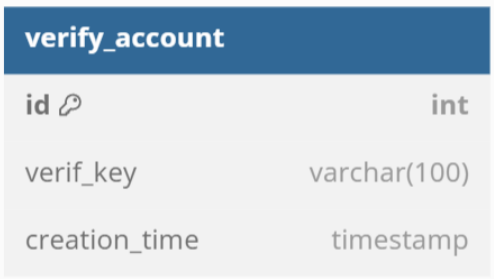
του πίνακα.

- Το πεδίο `'fname'`, περιλαμβάνει το όνομα του χρήστη.
- Το πεδίο `'lname'`, περιλαμβάνει το επώνυμο του χρήστη.
- Το πεδίο `'email'`, περιλαμβάνει τη διεύθυνση Email του χρήστη.
- Το πεδίο `'pass'`, περιλαμβάνει τον προσωπικό κωδικό πρόσβασης που έχει ορίσει ο χρήστης. Προτού καταχωρηθεί στη Βάση Δεδομένων, ο κωδικός πρόσβασης για λόγους ασφαλείας, μετατρέπεται σε μορφή hash χρησιμοποιώντας τον αλγόριθμο `CRYPT_BLOWFISH`.
- Το πεδίο `'token'`, περιλαμβάνει έναν κωδικό που επιτρέπει στους χρήστες να αποκτούν πρόσβαση στο Web API και να εκτελούν αιτήματα προς αυτό. Δημιουργείται και εισάγεται με αυτόματο τρόπο, ακολουθώντας την εξής διαδικασία: Αρχικά, γίνεται συνένωση του ονόματος του χρήστη με το αποτέλεσμα της συνάρτησης `'NOW()'` της MySQL, που επιστρέφει την τρέχουσα ημερομηνία και ώρα. Στη συνέχεια, χρησιμοποιώντας τον αλγόριθμο `MD5`, το αποτέλεσμα της συνένωσης μετατρέπεται σε μορφή hash, από όπου τελικά προκύπτει ένας δεκαεξαδικός αριθμός, μήκους 32 χαρακτήρων.
- Το πεδίο `'email_verif'`, υποδεικνύει αν ο χρήστης έχει προχωρήσει σε επιβεβαίωση της διεύθυνσης Email του. Από προεπιλογή, το πεδίο έχει την τιμή `'0'`, ενώ μόλις ολοκληρωθεί η επιβεβαίωση λαμβάνει την τιμή `'1'`.
- Το πεδίο `'allowPublic'`, υποδεικνύει αν ο χρήστης έχει το δικαίωμα χειρισμού δημόσιων συνόλων δεδομένων. Από προεπιλογή, το πεδίο έχει την τιμή `'0'` και εφόσον ο διαχειριστής παραχωρήσει τη σχετική άδεια, θα λάβει την τιμή `'1'`.

Σχήμα 4.3: Ο πίνακας `'users'` της Βάσης Δεδομένων της εφαρμογής

Ο πίνακας `'verify_account'`, εξυπηρετεί τη διαδικασία επιβεβαίωσης για ορισμένες ενέργειες του χρήστη. Η επιβεβαίωση είναι απαραίτητη σε περιπτώσεις, όπως είναι η εγγραφή στην εφαρμογή, η τροποποίηση της διεύθυνσης Email, καθώς και το αίτημα για ανάκτηση του λογαριασμού μετά από απώλεια του κωδικού. Όπως φαίνεται και από το Σχήμα 4.4, ο πίνακας αποτελείται από τρία πεδία. Ειδικότερα:

- Το πεδίο `'id'` αποτελεί ξένο κλειδί του ομώνυμου πεδίου του πίνακα `'users'`. Παράλληλα είναι το κύριο κλειδί του πίνακα.
- Το πεδίο `'verif_key'`, περιλαμβάνει έναν κωδικό που είναι απαραίτητος για την επιτυχή επιβεβαίωση μίας ενέργειας στο λογαριασμό. Το περιεχόμενο του κωδικού, αποτελείται από μία τυχαία συμβολοσειρά που δημιουργείται χρησιμοποιώντας τον αλγόριθμο `MD5` και τη μέθοδο `'random_bytes()'` της PHP.
- Το πεδίο `'creation_time'`, περιλαμβάνει την ημερομηνία και ώρα δημιουργίας του κωδικού επιβεβαίωσης και είναι χρήσιμο, προκειμένου να ελέγχεται το αν έχει λήξει.



Σχήμα 4.4: Ο πίνακας `'verify_account'` της Βάσης Δεδομένων της εφαρμογής

Ο πίνακας `'model_class'` χρησιμοποιείται υποστηρικτικά, για την κάλυψη ενεργειών που απαιτούνται κατά την αποθήκευση ενός προεκπαιδευμένου μοντέλου. Πιο συγκεκριμένα, ο πίνακας διατηρεί για κάθε χρήστη και για κάθε μοντέλο που αυτός δημιουργεί, την πληροφορία που αφορά το όνομα του πεδίου που χρησιμοποιήθηκε ως κλάση. Κάτι τέτοιο, είναι απαραίτητο όταν επιλέγεται από τον χρήστη ένα προεκπαιδευμένο μοντέλο, προκειμένου να μπορεί να προβληθεί το περιεχόμενό του, δηλαδή τα features και το class, από τα οποία αποτελείται. Όσον αφορά τα features, δίνεται η δυνατότητα ανάκτησής τους για κάθε μοντέλο, με χρήση σχετικής μεθόδου μέσω της βιβλιοθήκης Scikit-learn στην Python. Ωστόσο, για το πεδίο της κλάσης δεν υπάρχει κάποια αντίστοιχη επιλογή, για αυτό και αναπτύχθηκε μία λύση μέσω της Βάσης Δεδομένων.

Το προαναφερθέν θα γίνει καλύτερα κατανοητό, αφότου αναλυθεί η υλοποίηση της εν λόγω λειτουργικότητας, τόσο από την πλευρά του Web API, όσο και από αυτή του αντίστοιχου Python module.

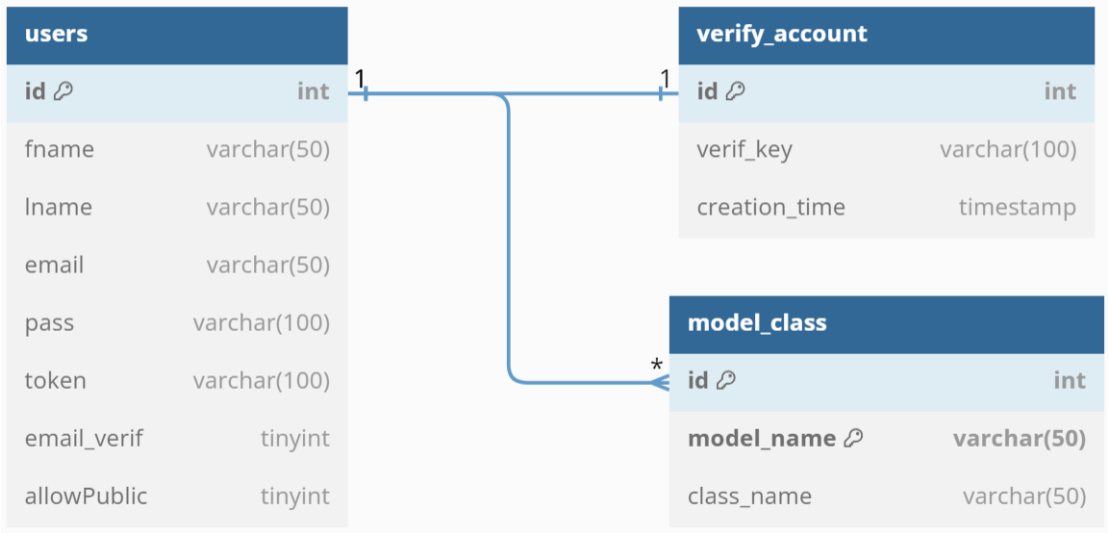
Όπως φαίνεται και από το Σχήμα 4.5, ο πίνακας αποτελείται από τρία πεδία. Ειδικότερα:

- Το πεδίο `'id'`, αποτελεί ξένο κλειδί του ομώνυμου πεδίου του πίνακα `'users'`.
- Το πεδίο `'model_name'`, περιλαμβάνει το όνομα του μοντέλου Δέντρου Απόφασης, που δημιουργείται. Το όνομα εισάγεται από τον χρήστη.
- Το πεδίο `'class_name'`, περιλαμβάνει το όνομα του πεδίου που χρησιμοποιήθηκε ως κλάση και αποτελεί χρήσιμη πληροφορία για τη συνέχεια.

Όσον αφορά το κύριο κλειδί του παραπάνω πίνακα, αυτό είναι σύνθετο και προκύπτει από τον συνδυασμό των πεδίων `'id'` και `'model_name'`, που είναι μοναδικός για κάθε χρήστη.

Σχήμα 4.5: Ο πίνακας `'model_class'` της Βάσης Δεδομένων της εφαρμογής

Στο Σχήμα 4.6, φαίνεται το διάγραμμα οντοτήτων-συσχετίσεων (ER diagram) της Βάσης Δεδομένων της εφαρμογής.



Σχήμα 4.6: Το διάγραμμα ER της Βάσης Δεδομένων της εφαρμογής

4.4.2 Web API

Το Web API αποτελεί την κινητήριο δύναμη της εφαρμογής, καθώς μέσω αυτού δρομολογείται η εκτέλεση της πλειοψηφίας των λειτουργιών και καθίσταται εφικτή η κάθε είδους επικοινωνία με το Back-end. Προκειμένου ένας χρήστης ή μία εφαρμογή, να αποκτήσει πρόσβαση στο Web API του AutoDTrees, θα πρέπει να προχωρήσει στην κλήση του endpoint που αντιστοιχεί σε μία λειτουργία. Το endpoint καλεί με τη σειρά του, το αντίστοιχο αρχείο κώδικα PHP και με την ολοκλήρωση της εκτέλεσής του, επιστρέφει τα αποτελέσματα σε μορφή `JSON`. Επιπλέον, σε κάθε endpoint θα πρέπει να παρέχονται και οι αντίστοιχες παράμετροι, ανάλογα με την περίπτωση. Ο τρόπος σύνταξης των παραμέτρων ποικίλλει ανάλογα με τη μέθοδο HTTP που χρησιμοποιείται. Έτσι, σε μεθόδους όπως είναι οι POST και DELETE οι παράμετροι εισάγονται στο σώμα (body) του αιτήματος, ενώ αντίθετα στη μέθοδο GET ενσωματώνονται στο URL του αιτήματος, μετά τον χαρακτήρα `'?'`. Μερικά παραδείγματα αντίστοιχων περιπτώσεων, φαίνονται και στον Πίνακα 4.1. Σε αυτόν συμπεριλαμβάνεται το σύνολο των endpoints που διαθέτει το Web API του AutoDTrees. Θα πρέπει επιπλέον να σημειωθεί, ότι κάθε ένα από τα αρχεία PHP που φαίνονται σε αυτόν, είναι αποθηκευμένα κάτω από το ακόλουθο directory: `https://kclusterhub.iee.ihu.gr/autodtrees/server/php/api/`. Συνεπώς, αν επί παραδείγ-

ματι κάποιος χρήστης επιθυμεί να καλέσει το endpoint για την εμφάνιση της λίστας με τα αποθηκευμένα μοντέλα, θα πρέπει η σύνταξή του να είναι η ακόλουθη: `https://kclusterhub.iee.ihu.gr/autodtrees/server/php/api/get_models.php?token=userToken`. Αν αντίθετα επιθυμεί την κλήση του endpoint για την διαγραφή μοντέλου, το URL θα είναι το `https://kclusterhub.iee.ihu.gr/autodtrees/server/php/api/delete_model.php` και θα προστεθεί στο body του αιτήματος το εξής: `{"token": "myToken", "file": "model.pkl"}`.

Πίνακας 4.1: Τα endpoints του Web API της εφαρμογής

Α/Α	**Μέθοδος HTTP**	**Endpoint**	**Περιγραφή**
1	POST	register.php	Εγγραφή χρήστη
2	POST	login.php	Είσοδος χρήστη
3	POST	edit-account.php	Ενημέρωση πληροφοριών χρήστη
4	DELETE	delete-account.php	Διαγραφή λογαριασμού χρήστη
5	GET	get_datasets.php? token={token}	Επιστρέφει τα σύνολα δεδομένων εκπαίδευσης
6	POST	upload_dataset.php	Ανέβασμα συνόλου δεδομένων εκπαίδευσης
7	GET	get_dataset_content.php? token={token}& folder={public|private}& file={file}	Επιστρέφει το περιεχόμενο ενός συνόλου δεδομένων εκπαίδευσης
8	DELETE	delete_dataset.php	Διαγράφει ένα σύνολο δεδομένων εκπαίδευσης
9	GET	get_unclassified_datasets.php? token={token}	Επιστρέφει τα μη κατηγοριοποιημένα datasets
10	POST	upload_unclassified_dataset.php	Ανέβασμα μη κατηγοριοποιημένου dataset
11	GET	get_unclassified_dataset_content.php? token={token}&file={file}	Επιστρέφει το περιεχόμενο ενός μη κατηγοριοποιημένου dataset
12	DELETE	delete_unclassified_dataset.php	Διαγράφει ένα μη κατηγοριοποιημένο dataset
13	GET	get_models.php? token={token}	Επιστρέφει τα προεκπαιδευμένα μοντέλα
14	GET	get_model_content.php? token={token}&file={file}	Επιστρέφει το περιεχόμενο ενός προεκπαιδευμένου μοντέλου
15	DELETE	delete_model.php	Διαγράφει ένα προεκπαιδευμένο μοντέλο
16	GET	visualize_tree.php? token={token}&file={file}	Οπτικοποιεί το Δέντρο Απόφασης για το επιλεγμένο μοντέλο
17	POST	cross_validation.php	Δημιουργεί και αξιολογεί ένα μοντέλο. Επιστρέφει τις μετρικές απόδοσης
18	POST	save_model.php	Αποθήκευση προεκπαιδευμένου μοντέλου
19	POST	classifyData.php	Κατηγοριοποιεί ένα dataset με χρήση του επιλεγμένου μοντέλου και επιστρέφει τα αποτελέσματα

Για την καλύτερη κατανόηση του τρόπου με τον οποίο λειτουργεί το Web API της εφαρμογής, παρατίθενται μερικά παραδείγματα από την υλοποίηση του κώδικα σε γλώσσα PHP.

Αρχικά, αφότου κληθεί ένα αρχείο κώδικα, το πρωταρχικό μέλημά του είναι ο έλεγχος της ορθότητας των πληροφοριών που εισήγαγε ο χρήστης. Όπως φαίνεται και στο απόσπασμα κώδικα του Σχήματος

4.7, αρχικά ελέγχεται αν έχει δοθεί η σωστή μέθοδος HTTP, η οποία στη συγκεκριμένη περίπτωση είναι η POST. Ακολουθούν, οι παράμετροι του αιτήματος, όπως είναι για παράδειγμα τα προσωπικά στοιχεία του χρήστη (όνομα, επώνυμο, διεύθυνση Email). Αν εισαχθεί λανθασμένα κάποιο άλλο είδος μεθόδου HTTP, είτε ο χρήστης δεν συμπληρώσει όλες τις απαιτούμενες παραμέτρους, τότε επιστρέφεται το κατάλληλο header και σχετικό μήνυμα λάθους, ενώ τερματίζεται η εκτέλεση του κώδικα.

```php
$method = $_SERVER['REQUEST_METHOD'];
$input = json_decode(file_get_contents('php://input'),true);

if($method != "POST") {
    header("HTTP/1.1 405 Method Not Allowed");
    print json_encode(['errormesg'=>"Method not allowed."]);
    exit;
}

if(!isset($input['fname'])){
    header("HTTP/1.1 400 Bad Request");
    print json_encode(['errormesg'=>"First name is not set."]);
    exit;
}

if(!isset($input['lname'])){
    header("HTTP/1.1 400 Bad Request");
    print json_encode(['errormesg'=>"Last name is not set."]);
    exit;
}

if(!isset($input['email'])){
    header("HTTP/1.1 400 Bad Request");
    print json_encode(['errormesg'=>"Email is not set."]);
    exit;
}
```

Σχήμα 4.7: Κώδικας ελέγχου ορθότητας παραμέτρων

Σε ορισμένες συνθήκες, για την επιτυχή ολοκλήρωση ενός ελέγχου, απαιτείται η επικοινωνία του Web API με τη Βάση Δεδομένων της εφαρμογής. Στο Σχήμα 4.8 φαίνεται ένα παράδειγμα του τρόπου, με τον οποίο συντάσσεται κώδικας SQL μέσα από την PHP και τη βιβλιοθήκη MySQLi, για την εκτέλεση αιτημάτων προς τη Βάση Δεδομένων. Ο παρακάτω έλεγχος διερευνά το ενδεχόμενο, ένα Email που εισάγεται κατά την εγγραφή νέου χρήστη, να έχει ήδη χρησιμοποιηθεί από άλλο άτομο.

```php
$query = 'select count(*) as c from users where email=?';
$st = $mysqli->prepare($query);
$st->bind_param('s',$email);
$st->execute();
$res = $st->get_result();
$count = $res->fetch_assoc()['c'];

if($count > 0){
    header("HTTP/1.1 400 Bad Request");
    print json_encode(['errormesg'=>"Email already exists."]);
    exit;
}
```

Σχήμα 4.8: Κώδικας για την εκτέλεση αιτημάτων προς τη Βάση Δεδομένων

Όπως είχε σημειωθεί και νωρίτερα, ο κωδικός πρόσβασης που εισάγει ένας χρήστης κατά την εγγραφή του, πριν να καταχωρηθεί στη Βάση Δεδομένων, μετατρέπεται για λόγους ασφαλείας σε μορφή hash,

χρησιμοποιώντας τον αλγόριθμο `CRYPT_BLOWFISH`. Επίσης, η δημιουργία του token γίνεται μέσω συνένωσης του ονόματος του χρήστη με την ημερομηνία και ώρα εγγραφής, εφαρμόζοντας τον αλγόριθμο `MD5` για τη μετατροπή σε μορφή hash. Ο κώδικας για τις παραπάνω διαδικασίες, συμπεριλαμβάνεται στο Σχήμα 4.9.

```php
$pass_hash = password_hash($pass, PASSWORD_BCRYPT);

$query = 'insert into users(fname, lname, email, pass, token)
values(?,?,?,?,md5(CONCAT( ?, NOW())))';
$st = $mysqli->prepare($query);
$st->bind_param('sssss',$fname,$lname,$email,$pass_hash,$fname);
$st->execute();
```

Σχήμα 4.9: Κώδικας για την καταχώριση στοιχείων χρήστη στη Βάση Δεδομένων

Για την εκτέλεση της πλειοψηφίας των λειτουργιών του Web API, απαιτείται η ταυτοποίηση του χρήστη μέσω του token που του έχει δοθεί για αυτόν το σκοπό. Συνεπώς, σε περιπτώσεις όπως είναι η μεταφόρτωση ενός συνόλου δεδομένων εκπαίδευσης, εκτελείται ο κώδικας του Σχήματος 4.10, όπου ελέγχεται το αν εισήχθη η παράμετρος του token και στη συνέχεια αν αυτό είναι υπαρκτό. Ο τελευταίος έλεγχος, αξιοποιεί μία function με την ονομασία `'token_exists'`, της οποίας ο κώδικας φαίνεται στο Σχήμα 4.11. Αυτή η function αξιοποιεί τη Βάση Δεδομένων και επιστρέφει την τιμή `'true'` αν το token υπάρχει και `'false'` στην αντίθετη περίπτωση.

```php
if(!isset($_POST['token'])){
    header("HTTP/1.1 400 Bad Request");
    print json_encode(['errormesg'=>"Token is not set."]);
    exit;
}

if(!token_exists($_POST['token'])){
    header("HTTP/1.1 400 Bad Request");
    print json_encode(['errormesg'=>"Token doesn't exist."]);
    exit;
}
```

Σχήμα 4.10: Κώδικας για την ταυτοποίηση χρήστη μέσω token

```php
function token_exists($token){
    global $mysqli;
    $query = 'select count(*) as c from users where token=?';
    $st = $mysqli->prepare($query);
    $st->bind_param('s',$token);
    $st->execute();
    $res = $st->get_result();
    $count = $res->fetch_assoc()['c'];

    if($count > 0){
        return true;
    }
    else{
        return false;
    }
}
```

Σχήμα 4.11: Κώδικας της μεθόδου `'token_exists'`

Συνεχίζοντας την αναφορά στη λειτουργία της μεταφόρτωσης συνόλων δεδομένων εκπαίδευσης, όταν επιλεγεί ένα αρχείο από τον χρήστη, θα πρέπει πρώτα να εκτελεστούν μια σειρά από σχετικούς ελέγχους. Όπως φαίνεται και στο Σχήμα 4.12, αρχικά διερευνάται η πιθανότητα να είναι κενή η αντίστοιχη παράμετρος και στη συνέχεια ελέγχονται τόσο ο τύπος του αρχείου, όσο και το μέγεθός του σε bytes. Θα πρέπει να σημειωθεί ότι μόνο τα αρχεία `.csv` είναι αποδεκτά, ενώ το μέγιστο επιτρεπόμενο μέγεθος είναι 10 MB (10.485.760 bytes).

```php
if(!isset($_FILES['file'])){
    header("HTTP/1.1 400 Bad Request");
    print json_encode(['errormesg'=>"Please select a file to upload."]);
    exit;
}

$fileType = pathinfo($_FILES['file']['name'],PATHINFO_EXTENSION);

if($fileType != "csv"){
    header("HTTP/1.1 415 Unsupported Media Type");
    print json_encode(['errormesg'=>"Only .csv files are supported."]);
    exit;
}

if($_FILES["file"]["size"] > 10485760){
    header("HTTP/1.1 400 Bad Request");
    print json_encode(['errormesg'=>"Max file size is 10 MB."]);
    exit;
}
```

Σχήμα 4.12: Κώδικας για τον έλεγχο ενός αρχείου dataset

```php
if($folder == "public"){
    $query = 'select allowPublic from users where token=?';
    $st = $mysqli->prepare($query);
    $st->bind_param('s',$_POST['token']);
    $st->execute();
    $res = $st->get_result();
    $allowPublic = $res->fetch_assoc()['allowPublic'];
    //Check if user's allowed to upload public data.

    if($allowPublic == 0){
        header("HTTP/1.1 403 Forbidden");
        print json_encode(['errormesg'=>"You aren't allowed to upload public data."]);
        exit;
    }

    $file_path = "../../py/public/datasets/" . basename($_FILES['file']['name']);

    if(file_exists($file_path)){
        header("HTTP/1.1 400 Bad Request");
        print json_encode(['errormesg'=>"File already exists."]);
        exit;
    }

    $upload = move_uploaded_file($_FILES["file"]["tmp_name"], $file_path);

}
```

Σχήμα 4.13: Κώδικας για τη μεταφόρτωση δημόσιου συνόλου δεδομένων εκπαίδευσης

Ακολούθως, προχωρά η διαδικασία μεταφόρτωσης του αρχείου. Ανάλογα με το αν πρόκειται για δημόσιο ή ιδιωτικό dataset, επιλέγεται και ο αντίστοιχος φάκελος για την αποθήκευσή του και διερευνάται το ενδεχόμενο να υπάρχει ήδη κάποιο αρχείο με το ίδιο όνομα. Στην περίπτωση του δημόσιου dataset, αξιοποιείται η Βάση Δεδομένων για να ελεγχθεί πρώτα, αν ο χρήστης έχει το δικαίωμα να χειρίζεται αυτού του είδους τα αρχεία, κάτι που φαίνεται και στο Σχήμα 4.13 που βρίσκεται παραπάνω. Με όμοιο τρόπο εξελίσσεται και η διαδικασία μεταφόρτωσης ενός μη κατηγοριοποιημένου dataset, με τη διαφορά ότι σε αυτήν την περίπτωση δεν υπάρχει η διάκριση σε δημόσια και ιδιωτικά σύνολα δεδομένων.

Όσον αφορά τη διαδικασία ανάκτησης του περιεχομένου ενός συνόλου δεδομένων, αυτή παρουσιάζεται στο Σχήμα 4.14. Μέσω του συγκεκριμένου κώδικα, διαβάζεται κάθε στοιχείο του αρχείου και αποθηκεύεται στην μεταβλητή `'$csv_array'` που είναι ένας δισδιάστατος πίνακας, με τόσες σειρές, όσες και αυτές του αρχείου και τόσες στήλες, όσα είναι τα πεδία του dataset.

```php
$row = 0;
if(($open_file = fopen($file_path, "r")) !== FALSE){
    while(($row_data = fgetcsv($open_file, 2048, ",")) !== FALSE){
        $countFields = count($row_data);
        for($i = 0; $i < $countFields; $i++){
            $csv_array[$row][$i] = $row_data[$i];
        }
        $row++;
    }
    fclose($open_file);
```

Σχήμα 4.14: Κώδικας για την ανάκτηση του περιεχομένου ενός συνόλου δεδομένων

Όταν πρόκειται για σύνολα δεδομένων εκπαίδευσης, είναι σημαντικό να γνωρίζει ο χρήστης ποια πεδία του συνόλου είναι εφικτό να χρησιμοποιηθούν ως features και ποια μπορούν να επιλεγούν ως το γνώρισμα κλάσης. Όπως έχει σημειωθεί στο Κεφάλαιο 2, η πλειοψηφία των υλοποιήσεων των Δέντρων Απόφασης υποστηρίζουν τόσο αριθμητικά δεδομένα, όσο και ονομαστικά (categorical) για τα πεδία των features. Η βιβλιοθήκη Scikit-learn, μέσω της οποίας χρησιμοποιούνται τα Δέντρα Απόφασης στην παρούσα εφαρμογή, εφαρμόζει μία παραλλαγή του αλγορίθμου CART, ωστόσο σε αντίθεση με αυτόν δεν υποστηρίζει την ύπαρξη categorical δεδομένων στα features. Συνεπώς, λόγω αυτού του γεγονότος εκτελείται σχετικός έλεγχος για την εξαίρεση τέτοιου είδους πεδίων από τις επιλογές που διατίθενται προς τον χρήστη. Ο κώδικας αυτής της υλοποίησης, φαίνεται στο Σχήμα 4.15. Ειδικότερα, ο πίνακας `'$csv_array'` σαρώνεται κατά στήλες, διατηρώντας μόνο τα πεδία με αριθμητικές τιμές, τα οποία και αποθηκεύει στον πίνακα `'$num_fields'`.

```php
for($j = 0; $j < $countFields; $j++){
    $columns = array();
    for($i2 = 1; $i2 < count($csv_array); $i2++){
        array_push($columns,$csv_array[$i2][$j]);
    }
    if((count(array_filter($columns,"is_numeric"))) == (count($csv_array) - 1)){
        if($csv_array[0][$j] != ""){
            array_push($num_fields,$csv_array[0][$j]);
        }
    }
}
```

Σχήμα 4.15: Κώδικας για την εξαίρεση των πεδίων με ονομαστικές τιμές

Προτού γίνει η ανάλυση του τρόπου εκτέλεσης του αλγορίθμου των Δέντρων Απόφασης, καθώς και των πάσης φύσεως λειτουργιών Μηχανικής Μάθησης της εφαρμογής, είναι σημαντικό να σημειωθεί και ο τρόπος με τον οποίο καλούνται τα modules που αντιστοιχούν σε αυτές. Όπως φαίνεται και στο σχετικό απόσπασμα του Σχήματος 4.16, η κλήση ενός αρχείου με κώδικα Python είναι εφικτή χρησιμοποιώντας τη μέθοδο `'shell_exec()'` της PHP. Ουσιαστικά, αυτή η μέθοδος επιτρέπει την κλήση του αρχείου Python, μέσω εκτέλεσης ενός shell script στον server. Με την ολοκλήρωσή του, πραγματοποιείται επιπλέον, έλεγχος για την πιθανότητα να προέκυψε κάποιο σφάλμα κατά την εκτέλεση του κώδικα της Python.

```php
$results = shell_exec("python3 ../../py/get_model_content.py $file_path");

if(!$results || $results == null){
    header("HTTP/1.1 400 Bad Request");
    print json_encode(['errormesg'=>"An error has occured while trying to run the Python module."]);
    exit;
}
```

Σχήμα 4.16: Κώδικας για την κλήση ενός αρχείου Python μέσα από την PHP

4.4.3 Δημιουργία, αξιολόγηση και αποθήκευση μοντέλου

Αρχικά, όταν ένας χρήστης πρόκειται να δημιουργήσει ένα νέο μοντέλο, θα πρέπει πρώτα να επιλέξει το σύνολο δεδομένων εκπαίδευσης, να εισάγει τα πεδία που θα χρησιμοποιηθούν ως features και το πεδίο του class και τελικά να επιλέξει τις παραμέτρους του κατηγοριοποιητή (`'max_depth'` και `'min_samples_leaf'`) και της μεθόδου k-fold cross-validation (`'k'`). Στη συνέχεια, εκτελείται ο κώδικας PHP για τον έλεγχο αυτών των στοιχείων, προτού τελικά προωθηθούν στο Python module που ασχολείται με την εκτίμηση της απόδοσης του μοντέλου. Ένα απόσπασμα του τελευταίου φαίνεται στο Σχήμα 4.17.

Ειδικότερα, αρχικοποιείται η μέθοδος `'KFold'` και ο κατηγοριοποιητής `'DecisionTreeClassifier'`, με βάση τις προτιμήσεις του χρήστη και εκτελείται η τεχνική k-fold cross-validation. Όπως σημειώθηκε και στο Κεφάλαιο 1, κατά τη διάρκειά της, το σύνολο δεδομένων διασπάται με τυχαίο τρόπο σε k διαφορετικά υποσύνολα που περιέχουν διαφορετικά στιγμιότυπα και πραγματοποιείται επαναληπτικά εκπαίδευση και έλεγχος του μοντέλου για k φορές.

```python
kf = KFold(n_splits = k, random_state = None, shuffle = True)
model = DecisionTreeClassifier(max_depth = max_depth, min_samples_leaf = min_samples_leaf)

for train_index , test_index in kf.split(attr):
    X_train , X_test = attr.iloc[train_index,:],attr.iloc[test_index,:]
    y_train , y_test = classlabel[train_index] , classlabel[test_index]

    model.fit(X_train,y_train)
    pred_values = model.predict(X_test)
```

Σχήμα 4.17: Κώδικας για την εκτέλεση της τεχνικής k-fold cross-validation

Κατόπιν, όπως φαίνεται και στο απόσπασμα του Σχήματος 4.18, για κάθε μία επανάληψη υπολογίζονται οι μετρικές απόδοσης accuracy, precision, recall και f-score και οι τιμές τους διατηρούνται προσωρινά μέσω βοηθητικών πινάκων. Μετά το τέλος της μεθόδου, εξάγεται ο μέσος όρος τους με στρογγυλοποί-

ηση των αποτελεσμάτων σε δύο δεκαδικά ψηφία. Αυτά τα αποτελέσματα, αποτελούν και τις τελικές μετρήσεις που αφορούν την απόδοση του μοντέλου και παρουσιάζονται στον χρήστη, προκειμένου να τον διευκολύνουν να αποφασίσει αν θα το αποθηκεύσει, για μελλοντική χρήση. Οι μετρικές υπολογίζονται τόσο για το κάθε ένα label, όσο και για το σύνολο του μοντέλου. Εξαίρεση αποτελεί το accuracy που λόγω της φύσης του, είναι ένα μέτρο που υπολογίζεται μόνο για το σύνολο του μοντέλου.

```
    accuracy = metrics.accuracy_score(y_test, pred_values)
    acc_class.append(accuracy)

    precision, recall, fscore, supp = metrics.precision_recall_fscore_support(
        y_test, pred_values, average = None, labels = labels)
    arr_pre.append(precision)
    arr_rec.append(recall)
    arr_fsc.append(fscore)


avg_acc = sum(acc_class)/k
avg_acc = round(avg_acc,2)
```

Σχήμα 4.18: Κώδικας για τον υπολογισμό των μετρικών απόδοσης

Εφόσον ο χρήστης επιλέξει να αποθηκεύσει το μοντέλο, τότε του ζητείται αρχικά να δώσει ένα όνομα για αυτό. Στη συνέχεια, καλείται το αντίστοιχο endpoint του Web API που, μαζί με το όνομα, λαμβάνει εκ νέου τις παραμέτρους που αφορούν το dataset και τον κατηγοριοποιητή και τις προωθεί στο Python module που εκτελεί τη λειτουργία της αποθήκευσης. Όπως φαίνεται και στο απόσπασμα του Σχήματος 4.19, η μέθοδος του κατηγοριοποιητή (`'DecisionTreeClassifier()'`) αρχικοποιείται με βάση τις παραμέτρους που δόθηκαν και ακολούθως το μοντέλο εκπαιδεύεται, χρησιμοποιώντας τα features (μεταβλητή `'attr'`) και το class (μεταβλητή `'classlabel'`) που επέλεξε ο χρήστης. Τελικά, η αποθήκευση του μοντέλου είναι εφικτή αξιοποιώντας τις δυνατότητες που προσφέρει η βιβλιοθήκη `'joblib'` της Python. Μέσω αυτής, το μοντέλο Δέντρου Απόφασης μετατρέπεται σε αρχείο με την κατάληξη `.pkl`. Όπως φαίνεται και στον παρακάτω κώδικα, το μοντέλο (μεταβλητή `'model'`) αποθηκεύεται στο path του προσωπικού φακέλου, με το όνομα που έδωσε ο χρήστης.

```
model = DecisionTreeClassifier(max_depth = max_depth, min_samples_leaf = min_samples_leaf)
model.fit(attr,classlabel)
joblib.dump(model,model_path)
print(json.dumps({"message": "Model successfully saved."}))
```

Σχήμα 4.19: Κώδικας για την αποθήκευση μοντέλου

Θα πρέπει επιπρόσθετα να σημειωθεί, ότι μετά την ολοκλήρωση του Python module και την επιστροφή των αποτελεσμάτων στο PHP αρχείο που το κάλεσε, εκτελείται στο τελευταίο ένα επιπλέον τμήμα κώδικα. Αυτό αφορά, την αποθήκευση του ονόματος του πεδίου που χρησιμοποιείται ως κλάση του μοντέλου, γεγονός ιδιαίτερα χρήσιμο για τη συνέχεια. Όπως φαίνεται και στο Σχήμα 4.20, η υλοποίησή του γίνεται μέσω της Βάσης Δεδομένων. Η εν λόγω πληροφορία λαμβάνεται από τη μεταβλητή `'$selected'`.

```
$query = 'insert into model_class(id, model_name, class_name) values(?,?,?)';
$st = $mysqli->prepare($query);
$st->bind_param('iss',$id,$model_file,$selected);
$st->execute();
```

Σχήμα 4.20: Κώδικας για την αποθήκευση του πεδίου κλάσης

4.4.4 Χρήση προεκπαιδευμένου μοντέλου

Όταν ένας χρήστης επιθυμεί να αξιοποιήσει κάποιο από τα αποθηκευμένα μοντέλα του, είναι χρήσιμο να μπορεί πρώτα να προβάλλει το περιεχόμενό του, δηλαδή τα features και το class από τα οποία αποτελείται. Κάτι τέτοιο θα τον διευκολύνει στην υπενθύμιση της δομής, αλλά και του είδους των πληροφοριών που αυτό περιέχει. Αφότου ο χρήστης επιλέξει το επιθυμητό μοντέλο, καλείται το σχετικό endpoint του Web API που ελέγχει την παράμετρο του ονόματος. Αυτό κατόπιν προωθείται στο Python module που είναι υπεύθυνο, για την ανάκτηση του περιεχομένου του μοντέλου. Όπως φαίνεται και στο Σχήμα 4.21, αρχικά διαβάζεται το path όπου είναι αποθηκευμένο το αρχείο `'.pkl'` με το μοντέλο και αξιοποιώντας τις δυνατότητες της βιβλιοθήκης `'joblib'` της Python, το αρχείο μετατρέπεται ξανά σε ένα μοντέλο Δέντρου Απόφασης και διατηρείται εντός της μεταβλητής `'joblib_model'`. Στη συνέχεια, χρησιμοποιώντας τη σχετική επιλογή που δίνει η βιβλιοθήκη Scikit-learn (ιδιότητα `'feature_names_in_'`), επιστρέφονται τα features από τα οποία αποτελείται το προεκπαιδευμένο μοντέλο.

```
model_file = sys.argv[1]
joblib_model = joblib.load(model_file)
columns = joblib_model.feature_names_in_.tolist()
print(json.dumps({"columns": columns}))
```

Σχήμα 4.21: Κώδικας για την ανάκτηση του περιεχομένου ενός μοντέλου

Ωστόσο, θα πρέπει να τονιστεί ότι δεν προσφέρεται κάποια αντίστοιχη επιλογή για την ανάκτηση του πεδίου class που χρησιμοποιείται στο μοντέλο. Για αυτόν τον λόγο, όπως είχε σημειωθεί και νωρίτερα, αναπτύχθηκε μία λύση όπου η πληροφορία για το εν λόγω πεδίο αποθηκεύεται εκ των προτέρων μέσω της Βάσης Δεδομένων. Συνεπώς, όπως φαίνεται και στο Σχήμα 4.22, μόλις ολοκληρωθεί η εκτέλεση του παραπάνω Python module και τα αποτελέσματα επιστραφούν στο αρχείο PHP που το κάλεσε, υλοποιείται εντός του τελευταίου ένα επιπλέον τμήμα κώδικα. Μέσω αυτού ανακτάται το πεδίο της κλάσης, δίνοντας ως παράμετρο στο SQL ερώτημα το `'id'` του χρήστη και το όνομα του προεκπαιδευμένου μοντέλου του. Τελικά, αποθηκεύεται στην PHP μεταβλητή `'$class_name'`.

```
$query = 'select class_name from model_class where id=? and model_name=?';
$st = $mysqli->prepare($query);
$st->bind_param('is',$id,$file);
$st->execute();
$res = $st->get_result();
$class_name = $res->fetch_assoc()['class_name'];
```

Σχήμα 4.22: Κώδικας για την ανάκτηση του πεδίου της κλάσης ενός μοντέλου

Μετά από τα παραπάνω βήματα, είναι πλέον εφικτό να επιστραφεί στον χρήστη το περιεχόμενο ενός μοντέλου συγκεντρωτικά, σε μορφή `JSON`. Ο σχετικός κώδικας PHP φαίνεται στο Σχήμα 4.23.

```
$results = json_decode($results,true);
$columns = $results['columns'];
print json_encode(['columns'=>$columns,'class_name'=>$class_name]);
```

Σχήμα 4.23: Κώδικας για τη συγκεντρωτική παρουσίαση του περιεχομένου ενός μοντέλου

Χρησιμοποιώντας ένα προεκπαιδευμένο μοντέλο Δέντρου Απόφασης, ένας χρήστης έχει στη συνέχεια τη δυνατότητα να εκτελεί προβλέψεις μη κατηγοριοποιημένων στιγμιοτύπων. Ειδικότερα, αφού επιλεγεί το επιθυμητό μοντέλο, ακολουθεί το βήμα της επιλογής του κατάλληλου συνόλου δεδομένων, που περιέχει μη κατηγοριοποιημένα στιγμιότυπα. Όπως είναι λογικό, για να καταστεί δυνατή η εκτέλεση της διαδικασίας, η δομή του dataset θα πρέπει να είναι όμοια με αυτή του μοντέλου, δηλαδή τα features του μοντέλου να υπάρχουν και στο dataset. Αυτό το γεγονός, ελέγχεται μέσω του κώδικα PHP με την κλήση του αντίστοιχου endpoint του Web API και στη συνέχεια όλες οι απαραίτητες παράμετροι, προωθούνται στο Python module που είναι υπεύθυνο για την κατηγοριοποίηση δεδομένων. Όπως φαίνεται και στο ακόλουθο απόσπασμα του Σχήματος 4.24, αρχικά το μοντέλο ανακαλείται μέσω της βιβλιοθήκης `'joblib'` της Python και με τη μέθοδο `'predict()'` πραγματοποιούνται οι προβλέψεις. Τα αποτελέσματά τους διατηρούνται στη μεταβλητή `'pred_values'`. Εν συνεχεία, επιστρέφεται στον χρήστη το dataset, με τη διαφορά ότι σε αυτό πλέον έχει προστεθεί μία επιπλέον στήλη, για την εμφάνιση των αποτελεσμάτων. Το νέο σύνολο δεδομένων, εκτός από τη μορφή `JSON`, επιστρέφεται και με τη μορφή αρχείου `.csv`, ώστε εφόσον ο χρήστης το επιθυμεί, να μπορεί να προχωρήσει στη λήψη του. Διευκρινίζεται επίσης, ότι η σύνταξη `'dataset[cols]'` χρησιμοποιείται, ώστε το νέο dataset που θα δημιουργηθεί, να περιλαμβάνει στις στήλες του μόνο τα features που υπάρχουν και στο μοντέλο, καθώς όπως είναι φυσικό, με βάση αυτά προέκυψαν και τα αποτελέσματα των προβλέψεων.

```
model = joblib.load(model_path)
pred_values = model.predict(attr)

dataset["predicted"] = pred_values
cols = checkVal
cols.append("predicted")
dataset[cols].to_csv(save_path, index = False, encoding = 'utf-8')
```

Σχήμα 4.24: Κώδικας για την πρόβλεψη μη κατηγοριοποιημένων στιγμιοτύπων

```
if className != None:
    classlabel = dataset[className]
    labels = classlabel.unique()

    acc = metrics.accuracy_score(classlabel, pred_values)
    acc = round(acc,2)

    pre_per_label, rec_per_label, fsc_per_label, supp = metrics.precision_recall_fscore_support(
        classlabel, pred_values, average = None, labels = labels)

    for i in range(len(pre_per_label)):
        pre_per_label[i] = round(pre_per_label[i],2)

    for i in range(len(rec_per_label)):
        rec_per_label[i] = round(rec_per_label[i],2)

    for i in range(len(fsc_per_label)):
        fsc_per_label[i] = round(fsc_per_label[i],2)
```

Σχήμα 4.25: Κώδικας για τον υπολογισμό μετρικών ποιότητας των αποτελεσμάτων

Σε περίπτωση που το επιλεγμένο dataset, περιέχει ήδη τη στήλη με την κλάση των παρατηρήσεων, τότε καθίσταται εφικτή η πραγματοποίηση συγκρίσεων με τα αποτελέσματα των προβλέψεων, μέσα από την αξιοποίηση μετρικών, για την εκτίμηση της ποιότητας των αποτελεσμάτων. Έτσι, αφότου διαπιστωθεί μέσω κατάλληλων ελέγχων η ύπαρξη του εν λόγω πεδίου, εντός του παραπάνω Python module

υπολογίζονται οι μετρικές απόδοσης accuracy, precision, recall και f-score. Οι τιμές τους κατόπιν στρογγυλοποιούνται σε δύο δεκαδικά ψηφία. Επίσης, θα πρέπει να σημειωθεί ότι και σε αυτή την περίπτωση, οι μετρικές υπολογίζονται τόσο για το κάθε ένα label, όσο και για το σύνολο του μοντέλου. Εξαίρεση αποτελεί το accuracy που λόγω της φύσης του, είναι ένα μέτρο που υπολογίζεται μόνο για το σύνολο του μοντέλου. Ένα στιγμιότυπο από την υλοποίηση των προαναφερθέντων, είναι ορατό μέσω του Σχήματος 4.25 που βρίσκεται παραπάνω.

Μία ακόμη επιλογή που δίνει η εφαρμογή AutoDTrees στον χρήστη, είναι να προχωρήσει στην οπτικοποίηση του Δέντρου Απόφασης για ένα προεκπαιδευμένο μοντέλο της επιλογής του. Κάτι τέτοιο είναι ιδιαίτερα χρήσιμο, καθώς θα τον διευκολύνει να κατανοήσει την δομή και τον τρόπο με τον οποίο εξάγει συμπεράσματα το μοντέλο.

Η διαδικασία υλοποίησης αυτής της λειτουργίας, εκτελείται σε δύο στάδια. Όσον αφορά το πρώτο, αρχικά καλείται το αντίστοιχο endpoint του Web API για τον έλεγχο των επιλογών του χρήστη και στη συνέχεια οι παράμετροι προωθούνται στο Python module που είναι υπεύθυνο για την κατασκευή του γραφήματος του Δέντρου Απόφασης. Όπως φαίνεται και από το Σχήμα 4.26, εντός του κώδικα Python, αρχικά ανακαλείται το επιθυμητό μοντέλο από το αντίστοιχο αρχείο `'.pkl'`, μέσω της βιβλιοθήκης `'joblib'`. Στη συνέχεια, χρησιμοποιείται η μέθοδος `'export_graphviz()'`, η οποία προχωρά στην εξαγωγή του διαγράμματος του Δέντρου Απόφασης, σε αρχείο τύπου `'.dot'` και σε μορφή συμβατή με το λογισμικό οπτικοποίησης γραφημάτων `'graphviz'`. Στη μέθοδο εισάγονται διάφορες παράμετροι, όπως είναι η μεταβλητή του μοντέλου (`'joblib_model'`), τα features και τα labels που αυτό περιλαμβάνει, το path όπου θα αποθηκευτεί το αρχείο `'.dot'`, καθώς και άλλες παραμέτρους που αφορούν την μορφοποίηση της εμφάνισης του διαγράμματος.

```
joblib_model = joblib.load(model_file)
feat = joblib_model.feature_names_in_.tolist()
cla = joblib_model.classes_.tolist()

dot_data = tree.export_graphviz(joblib_model, out_file = tree_path2,
    feature_names = feat, class_names = cla, filled = True, node_ids = True,
    rounded = True, precision = 2, max_depth = 10)
print(json.dumps({"message": "Dot data created successfully."}))
```

Σχήμα 4.26: Κώδικας για την οπτικοποίηση του Δέντρου Απόφασης

```
shell_exec("/var/www/html/webkmeans/kclusterhub/autodtrees/miniconda3/bin/dot
    -Tpng $tree_pathDot -o $tree_path");
```

Σχήμα 4.27: Κώδικας για την μετατροπή του αρχείου `'.dot'` σε εικόνα `'.png'`

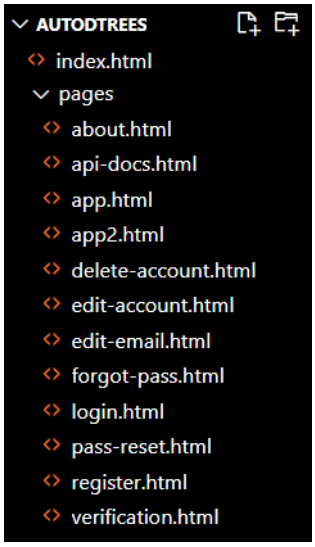
Το δεύτερο στάδιο της υλοποίησης, αφορά την μετατροπή του αρχείου `'.dot'`, σε αρχείο της μορφής `'.png'`, κάτι που διευκολύνει την παρουσίαση του διαγράμματος στον χρήστη, δίνοντάς του επίσης την επιλογή λήψης του αρχείου. Μόλις ολοκληρωθεί η εκτέλεση του προηγούμενου Python module και τα αποτελέσματα επιστραφούν στο αρχείο PHP που το κάλεσε, υλοποιείται εντός του τελευταίου ένα επιπλέον τμήμα κώδικα που σχετίζεται με την εν λόγω μετατροπή. Όπως φαίνεται και στο Σχήμα 4.27, καλείται η PHP μέθοδος `'shell_exec()'` για την εκτέλεση ενός shell script στον server. Μέσω αυτού, αξιοποιείται η εντολή `'dot'` που προσφέρει το λογισμικό `'graphviz'` για την εκτέλεση τέτοιου είδους μετατροπών. Η εντολή αφότου λάβει ως παράμετρο το path όπου βρίσκεται το αρχείο της μορφής `'.dot'`

(PHP μεταβλητή `'$tree_pathDot'`), το μετατρέπει σε αρχείο `'.png'` αποθηκεύοντάς το τελικά στον φάκελο του χρήστη (PHP μεταβλητή `'$tree_path'`).

4.5 Υλοποίηση του Front-end

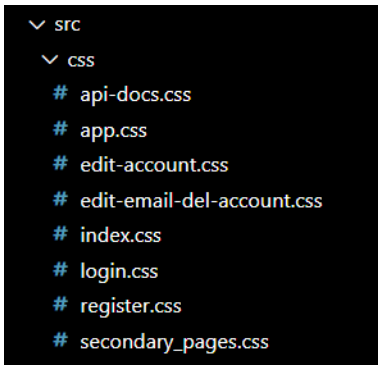
Ένα ακόμα ιδιαίτερα σημαντικό τμήμα του AutoDTrees, αποτελεί και το Front-end. Το Front-end, αξιοποιώντας την επικοινωνία του με το Web API της εφαρμογής, καθιστά εφικτή την χρήση όλων των δυνατοτήτων που αυτή προσφέρει, μέσα από μία φιλική προς τον χρήστη γραφική διεπαφή και χωρίς να απαιτείται η χρήση προγραμματιστικών εργαλείων. Για την ανάπτυξη αυτού του τμήματος της εφαρμογής, χρησιμοποιήθηκαν οι γλώσσες HTML και CSS από κοινού με το framework του Bootstrap, καθώς και η γλώσσα προγραμματισμού JavaScript με την βιβλιοθήκη jQuery.

Ειδικότερα, μέσω της γλώσσας σήμανσης υπερκειμένου HTML, κατέστη δυνατή η δημιουργία κάθε μίας από τις σελίδες που περιέχει η εφαρμογή, καθώς και η οργάνωση του περιεχομένου τους. Όπως φαίνεται και από το Σχήμα 4.28, τα αρχεία κώδικα HTML που αντιπροσωπεύουν τις σελίδες της εφαρμογής είναι 13 στο σύνολό τους. Η πλειοψηφία των αρχείων βρίσκεται εντός του φακέλου «pages», ενώ το `'index.html'` βρίσκεται στο αρχικό directory καθώς αντιπροσωπεύει την αρχική σελίδα της εφαρμογής.



Σχήμα 4.28: Αρχεία κώδικα HTML της εφαρμογής

Μέσω της γλώσσας CSS, κατέστη εφικτός ο καθορισμός του τρόπου με τον οποίο εμφανίζεται το περιεχόμενο των σελίδων, εφαρμόζοντας κατάλληλους κανόνες στα στοιχεία της HTML. Όπως φαίνεται και από το Σχήμα 4.29 που ακολουθεί, τα αρχεία κώδικα CSS περιλαμβάνονται κάτω από το directory `'/src/css'`. Τα αρχεία αυτά είναι οκτώ στο σύνολό τους, μιας και είναι εφικτή η επαναχρησιμοποίηση ενός από αυτά, για περισσότερες από μία σελίδες.



Σχήμα 4.29: Αρχεία κώδικα CSS της εφαρμογής

Επιπρόσθετα, για τον καθορισμό του τρόπου εμφάνισης των σελίδων αξιοποιήθηκε και το framework του Bootstrap. Ειδικότερα, στα στοιχεία HTML ενσωματώθηκαν μερικά από τα εργαλεία που προσφέρει το συγκεκριμένο framework, όπως είναι ορισμένες έτοιμες κλάσεις που συμβάλλουν στον σχεδιασμό συστατικών και λειτουργιών, με ταχύτερο και αποτελεσματικότερο τρόπο. Παράλληλα, μέσω του Bootstrap κατέστη εφικτή η υιοθέτηση των αρχών του responsive web design, συμβάλλοντας τελικά στο να δημιουργηθεί μία γραφική διεπαφή εύκολα αξιοποιήσιμη και προσβάσιμη από κάθε είδος συσκευής που μπορεί να κατέχει ένας χρήστης. Ένα τέτοιο παράδειγμα, φαίνεται και στο απόσπασμα κώδικα HTML του Σχήματος 4.30, όπου η στοίχιση και το μέγεθος των στοιχείων HTML προσαρμόζεται ανάλογα με το μέγεθος της οθόνης, χρησιμοποιώντας τις έτοιμες κλάσεις του Bootstrap.

```
<div class="row">
  <div class="col-12 col-lg-6">
    <h3>Classify Data using Pretrained Models</h3>
    <p>Upload unclassified datasets and predict their labels</p>
  </div>
  <div class="col-12 col-lg-6">
    <img class="img-fluid" src="./src/img/predicted_data.jpg" alt="Predicted Data">
  </div>
</div>
```

Σχήμα 4.30: Καθορισμός εμφάνισης στοιχείων HTML με χρήση του Bootstrap

Μέσω της γλώσσας προγραμματισμού JavaScript που χρησιμοποιήθηκε από κοινού με την βιβλιοθήκη jQuery, κατέστη εφικτός ο αποτελεσματικότερος καθορισμός του τρόπου εμφάνισης, καθώς και της μορφοποίησης τμημάτων της ιστοσελίδας. Ο αντίκτυπος αυτού του γεγονότος, ήταν η ιστοσελίδα να μετατραπεί σε μία διαδραστική εφαρμογή, όπου η δομή και το περιεχόμενό της μπορεί να μεταβάλλεται ζωντανά, ανάλογα με τις ενέργειες του χρήστη και χωρίς να απαιτείται συνεχώς η επαναφόρτωση της κάθε μίας σελίδας. Ένα τέτοιο παράδειγμα αποτελεί η διαδικασία εκτέλεσης αιτημάτων προς το Web API και η εν συνεχεία παρουσίαση των αποτελεσμάτων τους στον χρήστη μέσω της διεπαφής. Η κλήση των endpoints του Web API, πραγματοποιείται υιοθετώντας την τεχνική AJAX (Asynchronous JavaScript and XML). Στην εφαρμογή, κάτι τέτοιο είναι δυνατό χρησιμοποιώντας τη μέθοδο `'$.ajax()'` που προσφέρει η βιβλιοθήκη jQuery. Όπως φαίνεται και στο Σχήμα 4.31, αρχικά καλείται το endpoint που εκτελεί τη διαγραφή του λογαριασμού ενός χρήστη, εισάγοντας το αντίστοιχο URL, τη μέθοδο HTTP που απαιτείται, καθώς και τις τιμές που εισήγαγε ο χρήστης για τις παραμέτρους του αιτήματος, σε μορφή `JSON`. Στη συνέχεια, η συγκεκριμένη μέθοδος της jQuery μέσω των `'success'` και `'error'`, επιτρέπει τον καθορισμό του κώδικα που θα υλοποιηθεί μετά το τέλος της εκτέλεσης του endpoint, ανάλογα με

την επιτυχή ή μη ολοκλήρωσή του αντίστοιχα. Συνεπώς, σε περίπτωση επιτυχούς διαγραφής λογαριασμού, εμφανίζεται το κατάλληλο μήνυμα ειδοποίησης προς τον χρήστη και οι προσωπικές πληροφορίες του διαγράφονται από τη μνήμη του περιηγητή. Θα πρέπει να αναφερθεί, ότι η αποθήκευση αυτών των πληροφοριών εκτελείται όταν συνδέεται ο χρήστης στην εφαρμογή και αποτελεί ένα σημαντικό γεγονός, ώστε να μπορεί να έχει συνεχή πρόσβαση τόσο στις σελίδες που απαιτούν ταυτοποίηση, όσο και στα αντίστοιχα endpoints του Web API, χωρίς να απαιτείται κάθε φορά να εισάγει εκ νέου τα προσωπικά του στοιχεία σύνδεσης. Ο χειρισμός των αποθηκευμένων πληροφοριών του χρήστη, είναι εφικτός μέσω της ιδιότητας `'sessionStorage'` της JavaScript.

```
$.ajax({
    url: '../server/php/api/delete-account.php',
    method: 'DELETE',
    data: JSON.stringify({pass_del: pass_del,
        pass_del_confirm: pass_del_confirm, token: token}),
    dataType: "json",
    contentType: 'application/json',
    success: function(){
        sessionStorage.clear();
        alert_success("Account successfully deleted.");
        $("#loadingbtn").hide();
        $("#homeref").show();
    },
    error: function(xhr,status,error){
        var response = JSON.parse(xhr.responseText);
        var errormes = response.errormesg;
        alert_danger(errormes);
        $("#loadingbtn").hide();
        $("#delbtn").show();
    }
});
```

Σχήμα 4.31: Κώδικας JavaScript - jQuery για την κλήση endpoint του Web API

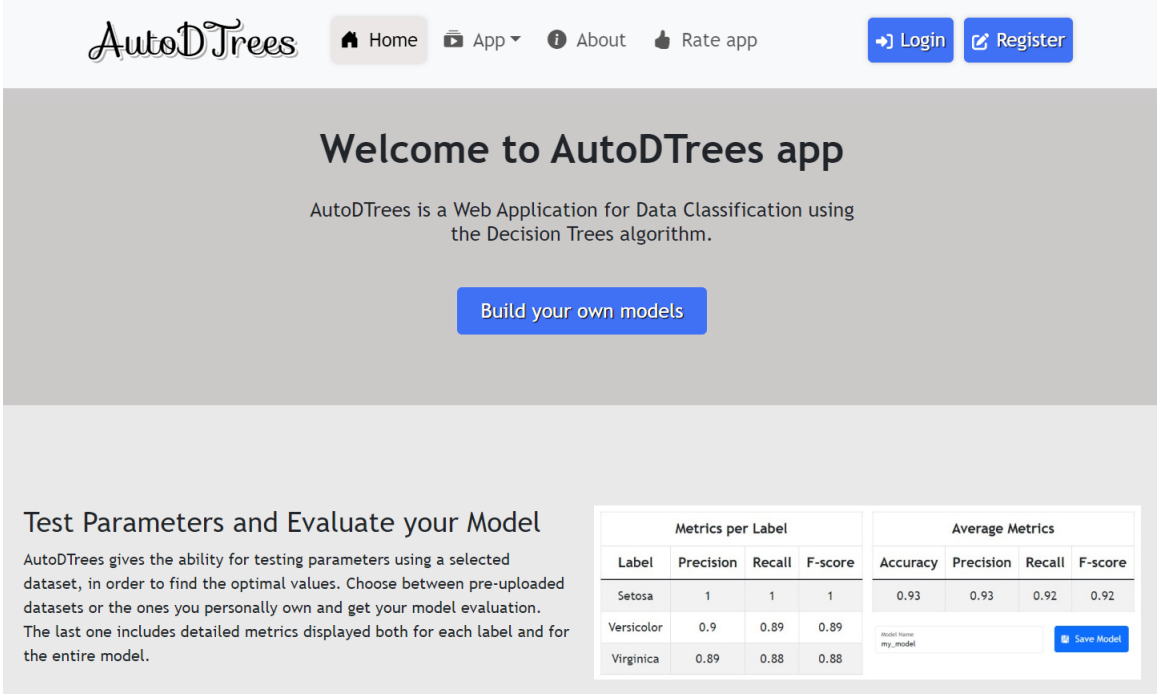
4.6 GitHub repository

Ολοκληρώνοντας την περιγραφή του τρόπου υλοποίησης της εφαρμογής, αλλά και συνολικά το Κεφάλαιο 4, θα πρέπει να σημειωθεί ότι το σύνολο του κώδικα του AutoDTrees είναι διαθέσιμο και πλήρως προσβάσιμο, μέσω της πλατφόρμας του GitHub στον ακόλουθο σύνδεσμο:
`https://github.com/manthoszog/AutoDTrees`.

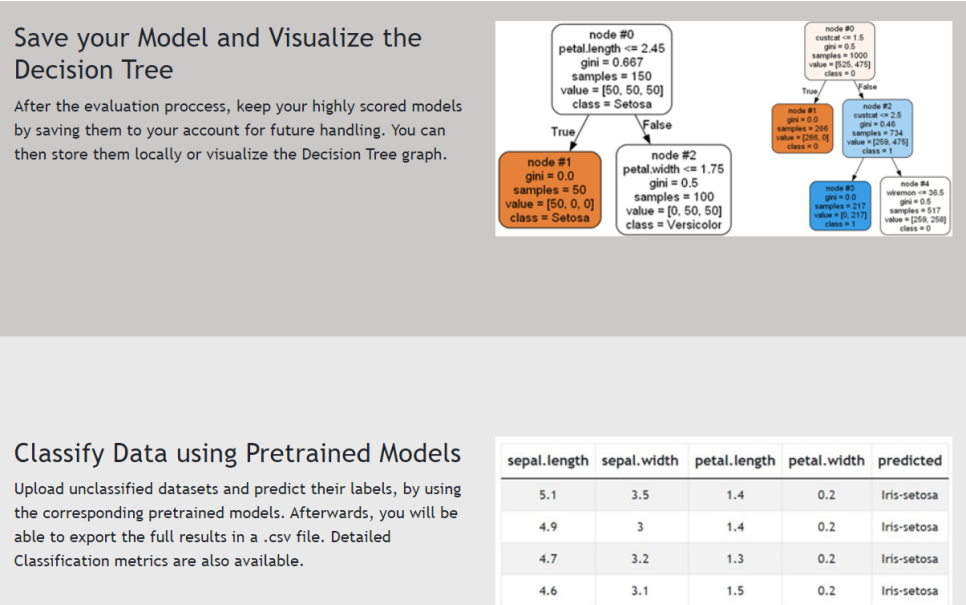
Κεφάλαιο 5ο: Παρουσίαση του AutoDTrees

5.1 Αρχική Σελίδα

Ακολουθώντας τον σύνδεσμο `https://kclusterhub.iee.ihu.gr/autodtrees`, ο χρήστης εισέρχεται στην Αρχική Σελίδα του AutoDTrees. Όπως φαίνεται και στο Σχήμα 5.1, εμφανίζεται ένα μήνυμα που τον καλωσορίζει στην εφαρμογή και τον προτρέπει να προχωρήσει στην αξιοποίηση των λειτουργιών της. Στο πάνω μέρος βρίσκεται το μενού, από όπου ο χρήστης μπορεί να περιηγηθεί στις διάφορες σελίδες, ενώ του δίνεται η επιλογή να προχωρήσει στην εγγραφή ή είσοδό του με τον προσωπικό του λογαριασμό. Στο κάτω μέρος της Αρχικής Σελίδας (συνέχεια στο Σχήμα 5.2), παρουσιάζονται οι κύριες δυνατότητες και λειτουργίες της εφαρμογής, για την καλύτερη καθοδήγηση του χρήστη.



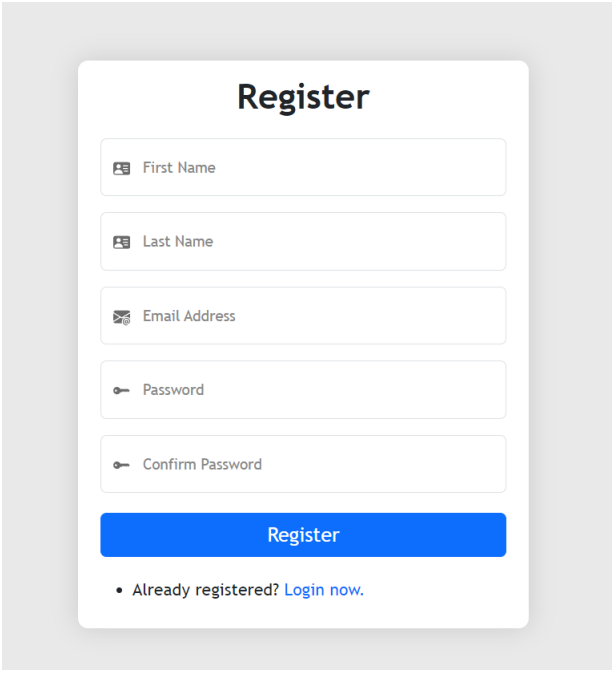
Σχήμα 5.1: Η Αρχική Σελίδα του AutoDTrees



Σχήμα 5.2: Παρουσίαση δυνατοτήτων της εφαρμογής

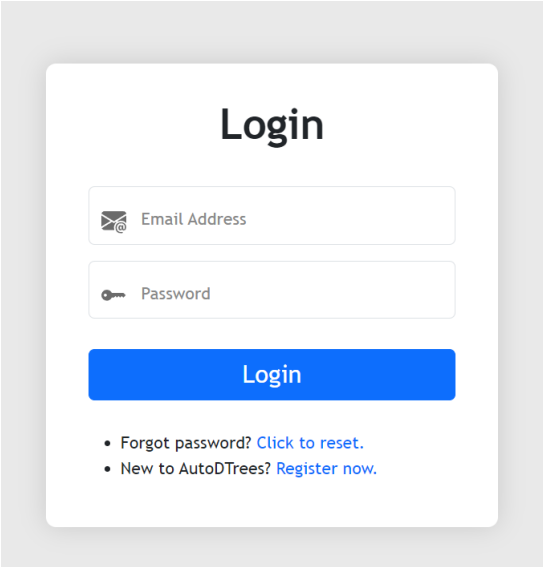
5.2 Δημιουργία λογαριασμού και σύνδεση στην εφαρμογή

Κάνοντας κλικ στο αντίστοιχο κουμπί του μενού, εμφανίζεται η σελίδα για τη δημιουργία λογαριασμού. Όπως φαίνεται και στο Σχήμα 5.3, μέσω μίας σχετικής φόρμας ο χρήστης καλείται να συμπληρώσει τα προσωπικά στοιχεία του, όπως είναι το όνομα, το επώνυμο, η διεύθυνση Email, καθώς και έναν επιθυμητό κωδικό πρόσβασης. Στη συνέχεια, θα πρέπει να επιλέξει το κουμπί «Register» και μετά την αποστολή ηλεκτρονικού μηνύματος επιβεβαίωσης, η διαδικασία εγγραφής ολοκληρώνεται και ο χρήστης μπορεί να προχωρήσει με τη σύνδεσή του στην εφαρμογή.

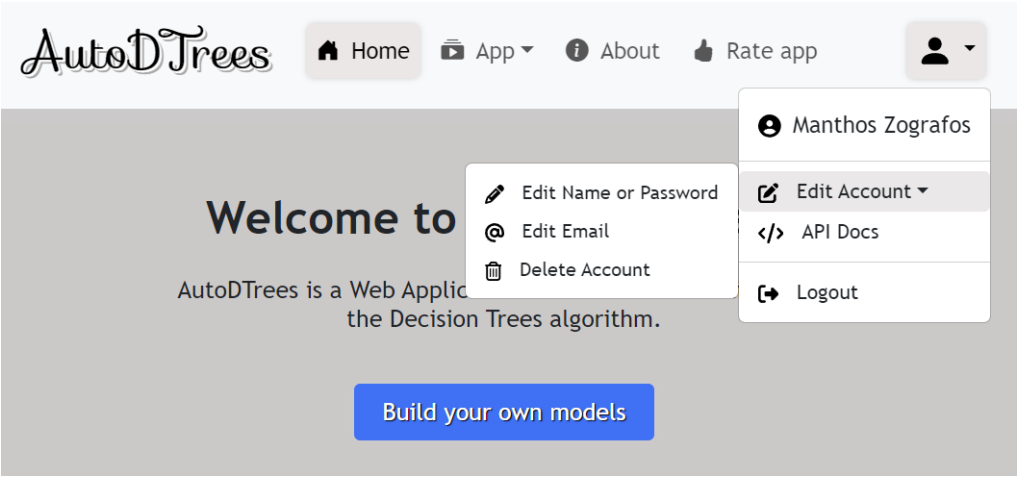


Σχήμα 5.3: Η σελίδα για τη δημιουργία λογαριασμού

Όπως φαίνεται στο Σχήμα 5.4, εφόσον ο χρήστης επιλέξει να συνδεθεί στον λογαριασμό του, εμφανίζεται η σχετική σελίδα, όπου καλείται να εισάγει την διεύθυνση Email και τον προσωπικό του κωδικό πρόσβασης. Αν η ταυτοποίηση ολοκληρωθεί με επιτυχία, μεταφέρεται εκ νέου στην Αρχική Σελίδα, με τη διαφορά ότι πλέον στο πάνω μέρος, αντί για τα κουμπιά «Login» και «Register», εμφανίζεται ένα εικονίδιο που επιτρέπει μέσω μίας λίστας, τη διαχείριση του λογαριασμού (Σχήμα 5.5).



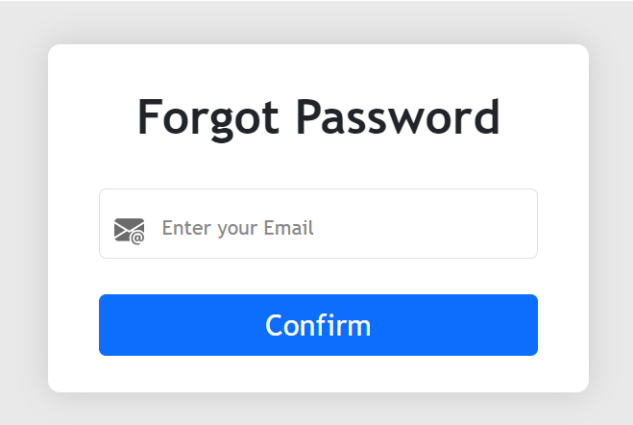
Σχήμα 5.4: Η σελίδα για τη σύνδεση στον λογαριασμό



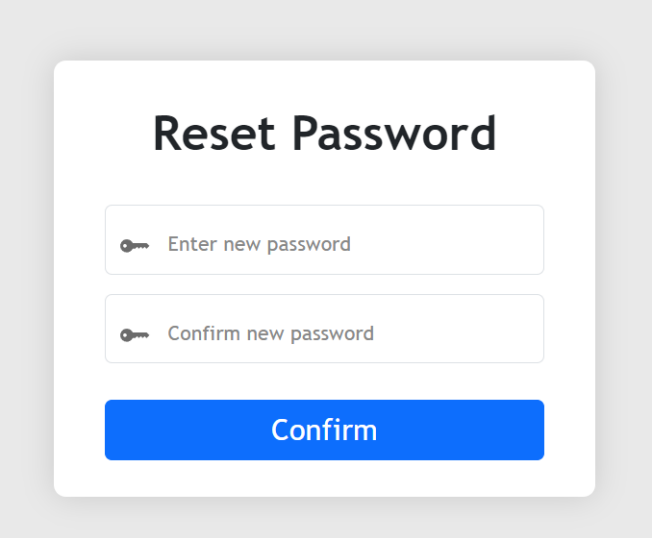
Σχήμα 5.5: Λίστα επιλογών για τη διαχείριση λογαριασμού

5.3 Ανάκτηση και διαχείριση λογαριασμού

Όσον αφορά την ανάκτηση του λογαριασμού μετά από πιθανή απώλεια του κωδικού, αυτή η λειτουργία είναι διαθέσιμη κάνοντας κλικ στη σχετική επιλογή που υπάρχει εντός της σελίδας «Login» και συγκεκριμένα στο κάτω μέρος αυτής (Σχήμα 5.4 που προηγήθηκε). Κατόπιν, όπως φαίνεται και στο Σχήμα 5.6, ζητείται αρχικά από τον χρήστη να εισάγει τη διεύθυνση Email του και μετά την αποστολή ηλεκτρονικού μηνύματος επιβεβαίωσης, μπορεί στη συνέχεια να δηλώσει ένα νέο κωδικό πρόσβασης (Σχήμα 5.7), για την ολοκλήρωση της διαδικασίας ανάκτησης.

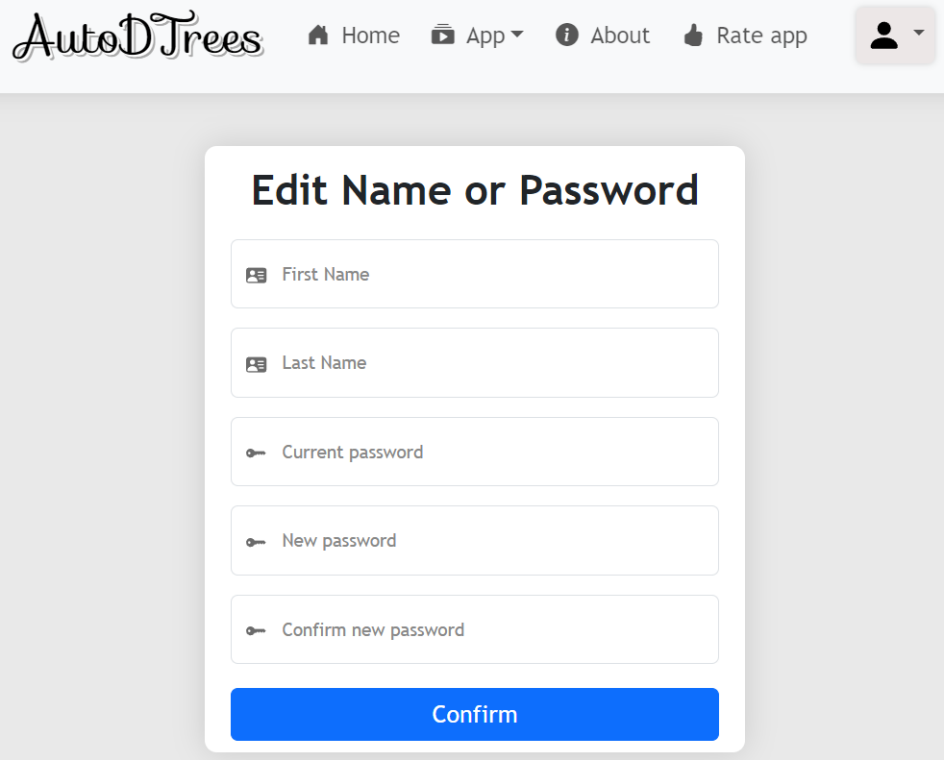


Σχήμα 5.6: Σελίδα για την αίτηση ανάκτησης λογαριασμού



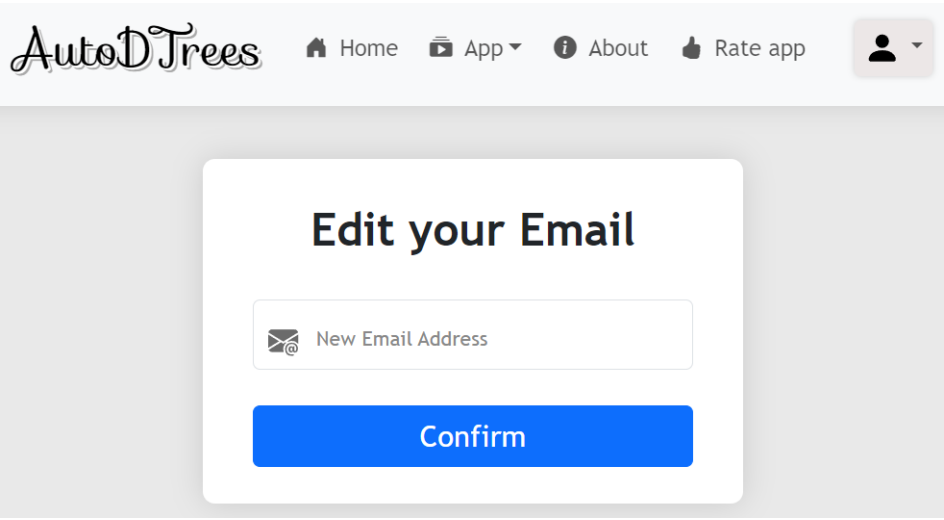
Σχήμα 5.7: Σελίδα εισαγωγής νέου κωδικού για την ανάκτηση λογαριασμού

Κάνοντας κλικ σε κάποια από τις επιλογές διαχείρισης λογαριασμού «Edit Name or Password», «Edit Email» και «Delete Account» (Σχήμα 5.5 που προηγήθηκε), ο χρήστης οδηγείται στην αντίστοιχη σελίδα. Έτσι, στην πρώτη περίπτωση (Σχήμα 5.8) μπορεί να τροποποιήσει όποιο στοιχείο επιθυμεί, συμπεριλαμβανομένου του ονόματος, του επιθέτου ή και του κωδικού πρόσβασης. Εφόσον ζητηθεί η ενημέρωση του κωδικού, θα πρέπει πρώτα να γίνει επιβεβαίωση του τρέχοντος.



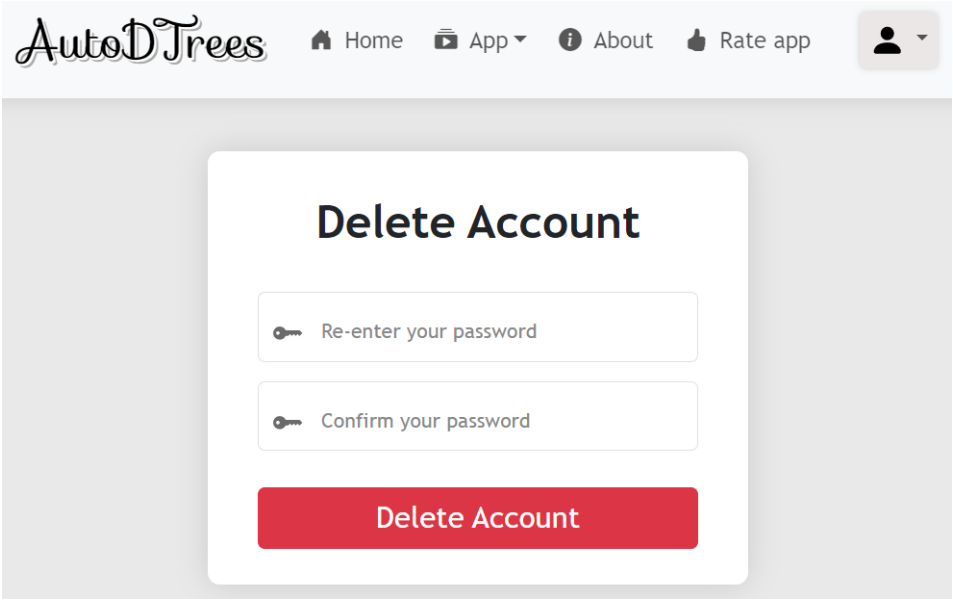
Σχήμα 5.8: Σελίδα για την τροποποίηση ονόματος, επιθέτου ή κωδικού

Στη δεύτερη περίπτωση (Σχήμα 5.9), ο χρήστης μπορεί να τροποποιήσει τη διεύθυνση Email του, με τη διαδικασία να ολοκληρώνεται κατόπιν επιβεβαίωσης μέσω αποστολής ηλεκτρονικού μηνύματος.

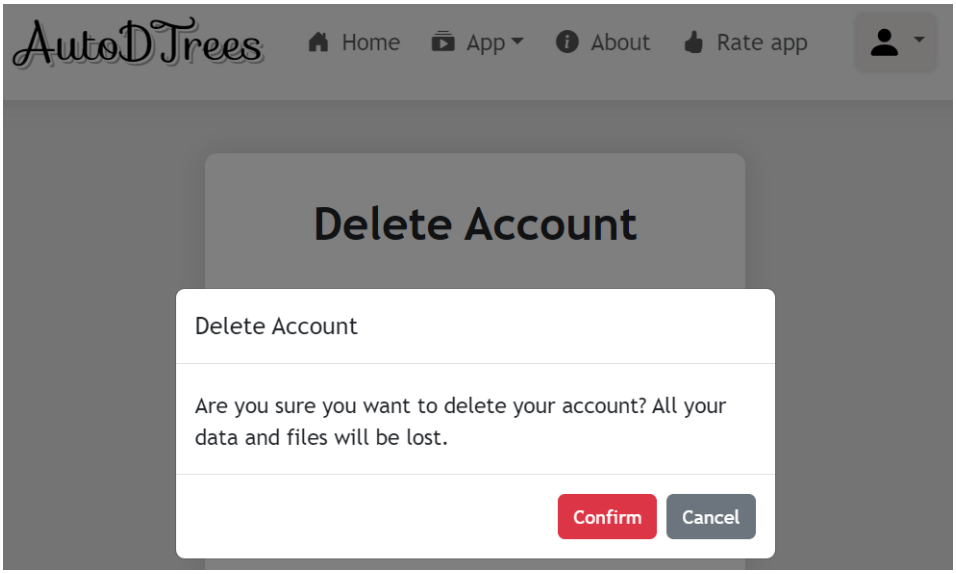


Σχήμα 5.9: Σελίδα για την τροποποίηση του Email

Στην τελευταία περίπτωση (Σχήμα 5.10) δίνεται η επιλογή στον χρήστη, εφόσον το επιθυμεί, να προχωρήσει στην οριστική διαγραφή του λογαριασμού του. Για να ολοκληρωθεί η διαδικασία, θα πρέπει πρώτα να εισαχθεί εκ νέου ο κωδικός πρόσβασης και στη συνέχεια να επιλεγεί το κουμπί επιβεβαίωσης (Σχήμα 5.11).



Σχήμα 5.10: Σελίδα για τη διαγραφή λογαριασμού



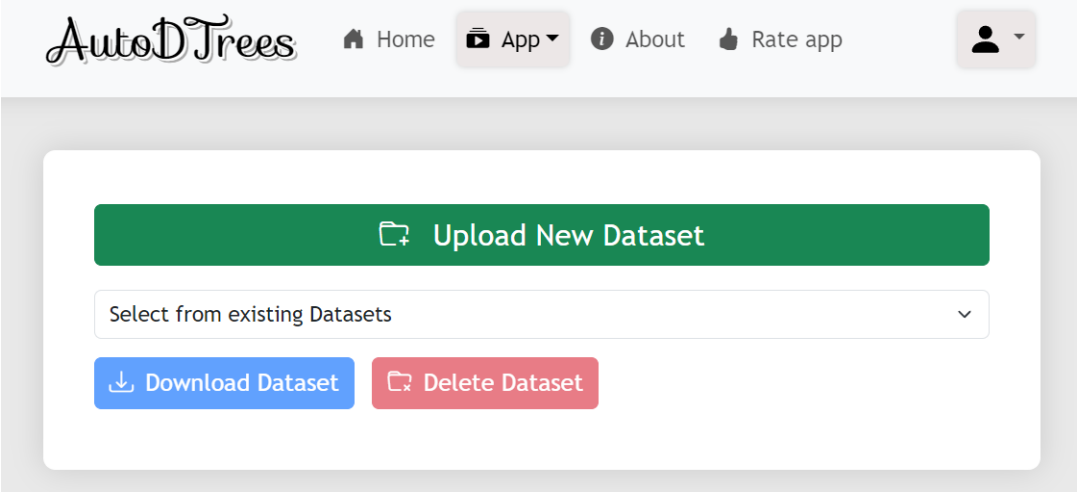
Σχήμα 5.11: Μήνυμα επιβεβαίωσης διαγραφής λογαριασμού

5.4 Σελίδα δημιουργίας μοντέλων

Κάνοντας κλικ στην επιλογή «App» από το μενού, εμφανίζονται δύο επιπλέον επιλογές ανάλογα με την ενέργεια που επιθυμεί να εκτελέσει ο χρήστης. Η πρώτη επιλογή αφορά τη σελίδα για τη δημιουργία νέου μοντέλου Δέντρου Απόφασης, ενώ η δεύτερη σχετίζεται με τη χρήση των προεκπαιδευμένων.

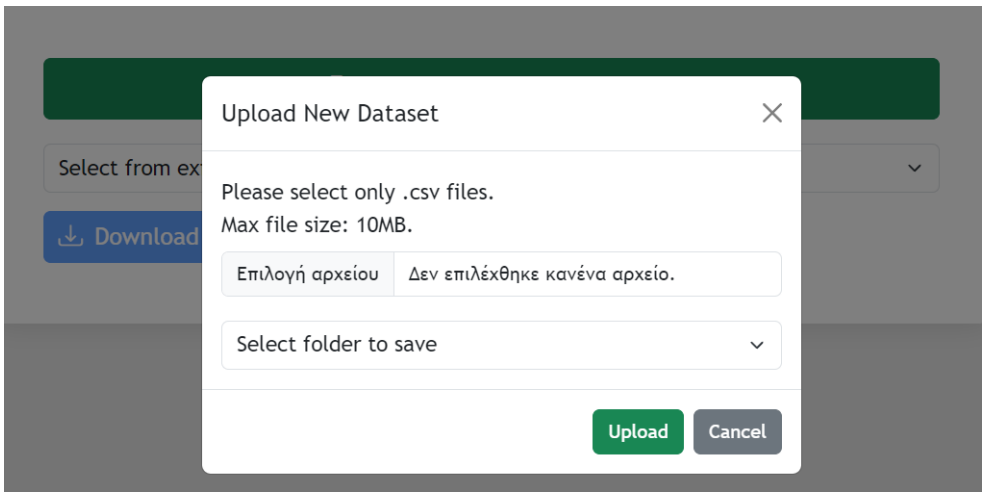
Κατά την είσοδο του χρήστη στη σελίδα δημιουργίας νέου μοντέλου, αυτή έχει αρχικά την μορφή του

Σχήματος 5.12.



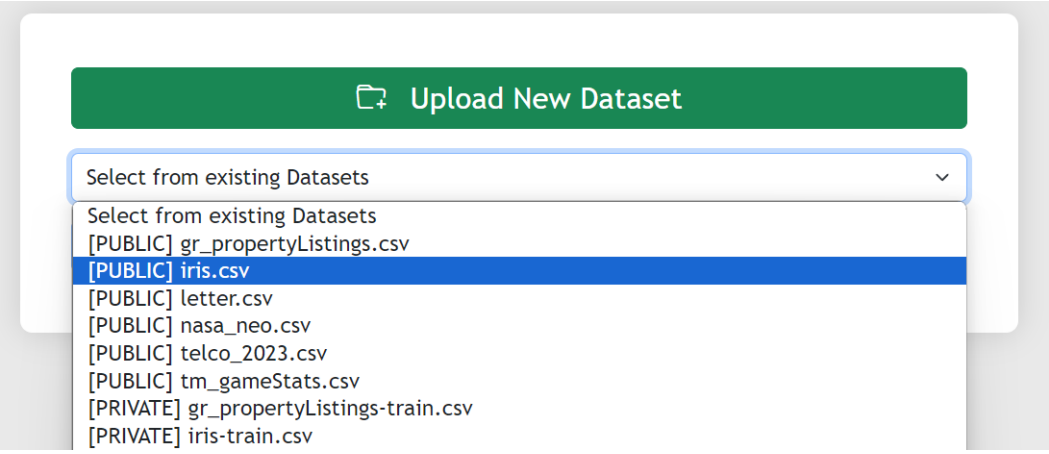
Σχήμα 5.12: Σελίδα δημιουργίας νέου μοντέλου

Ο χρήστης μπορεί είτε να ανεβάσει ένα νέο, είτε να επιλέξει ένα ήδη υπάρχον σύνολο δεδομένων εκπαίδευσης. Στην πρώτη περίπτωση, όπως φαίνεται στο Σχήμα 5.13, εμφανίζεται ένα μενού για την επιλογή του επιθυμητού αρχείου από τον προσωπικό υπολογιστή του χρήστη, καθώς και του φακέλου, ιδιωτικού ή δημόσιου, όπου θα αποθηκευτεί το αρχείο. Εφόσον δεν έχει το δικαίωμα χειρισμού δημόσιων συνόλων, θα εμφανιστεί το κατάλληλο μήνυμα.

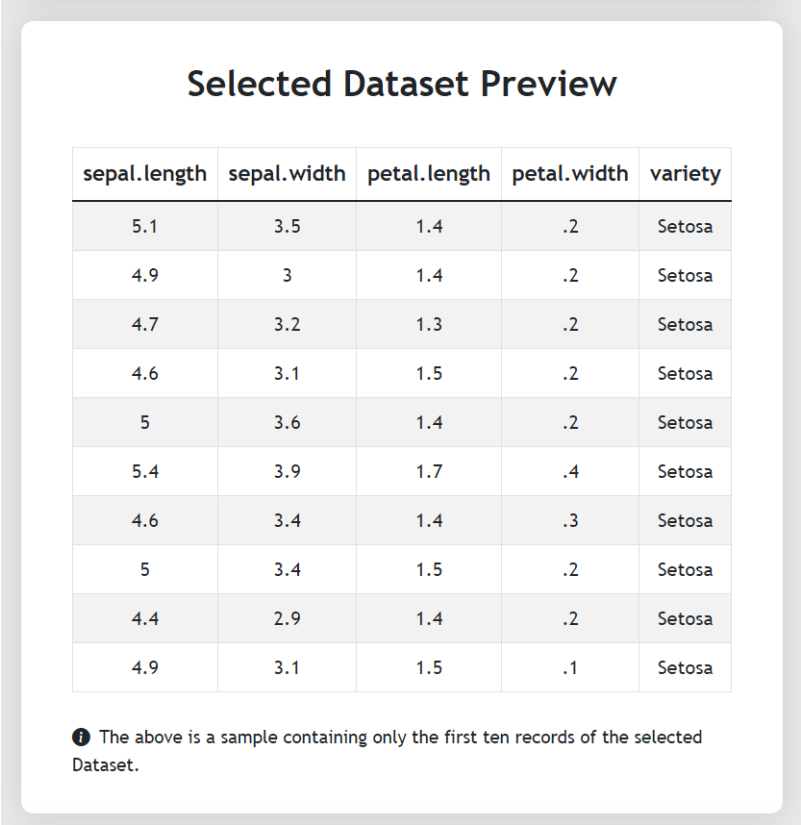


Σχήμα 5.13: Μενού για το ανέβασμα συνόλου δεδομένων εκπαίδευσης

Η μορφή της λίστας με τα αποθηκευμένα σύνολα φαίνεται στο Σχήμα 5.14. Αφότου επιλεγεί ένα από αυτά, εμφανίζεται μία προεπισκόπησή του (Σχήμα 5.15), ενώ ο χρήστης μπορεί επίσης να προχωρήσει στη λήψη ή τη διαγραφή του αρχείου. Αν πρόκειται για διαγραφή δημόσιου αρχείου, εκτελείται πρώτα ο σχετικός έλεγχος και εμφανίζεται κατάλληλο μήνυμα.

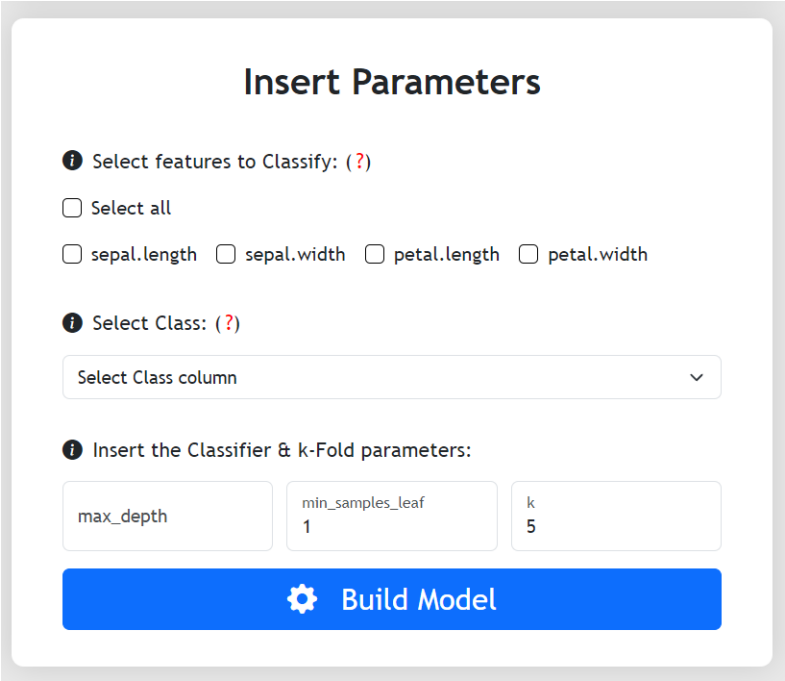


Σχήμα 5.14: Λίστα αποθηκευμένων συνόλων δεδομένων εκπαίδευσης

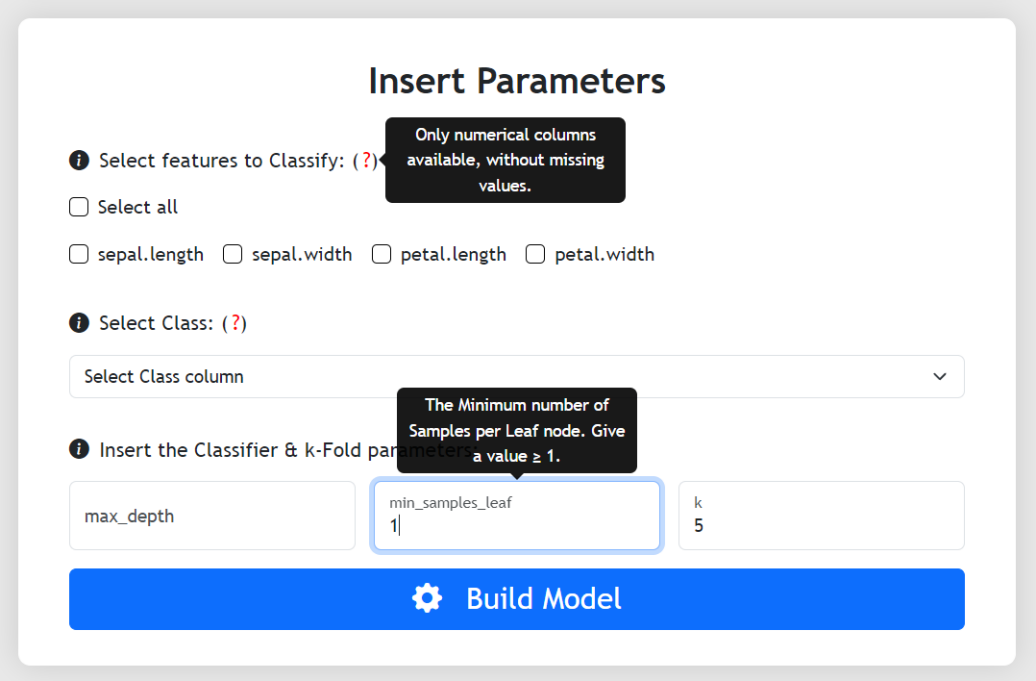


Σχήμα 5.15: Προεπισκόπηση συνόλου δεδομένων εκπαίδευσης

Όπως φαίνεται και στο Σχήμα 5.16, στο κάτω μέρος της σελίδας, μετά από την προεπισκόπηση, εμφανίζεται ένα ακόμη τμήμα που αφορά των καθορισμό των παραμέτρων της δημιουργίας μοντέλου Δέντρου Απόφασης. Σε αυτό, ο χρήστης μπορεί αρχικά να επιλέξει τα πεδία του dataset, τα οποία επιθυμεί να χρησιμοποιηθούν ως features, καθώς και το πεδίο του class. Στη συνέχεια, θα πρέπει να ορίσει τις παραμέτρους του κατηγοριοποιητή και της μεθόδου k-fold cross-validation, είτε εισάγοντας τις επιθυμητές τιμές, είτε χρησιμοποιώντας τις προεπιλεγμένες. Για την καλύτερη πληροφόρηση και κατανόηση του τρόπου λειτουργίας, παρέχεται καθοδήγηση με την εμφάνιση σχετικών μηνυμάτων (tooltips), κάτι που φαίνεται στο Σχήμα 5.17.

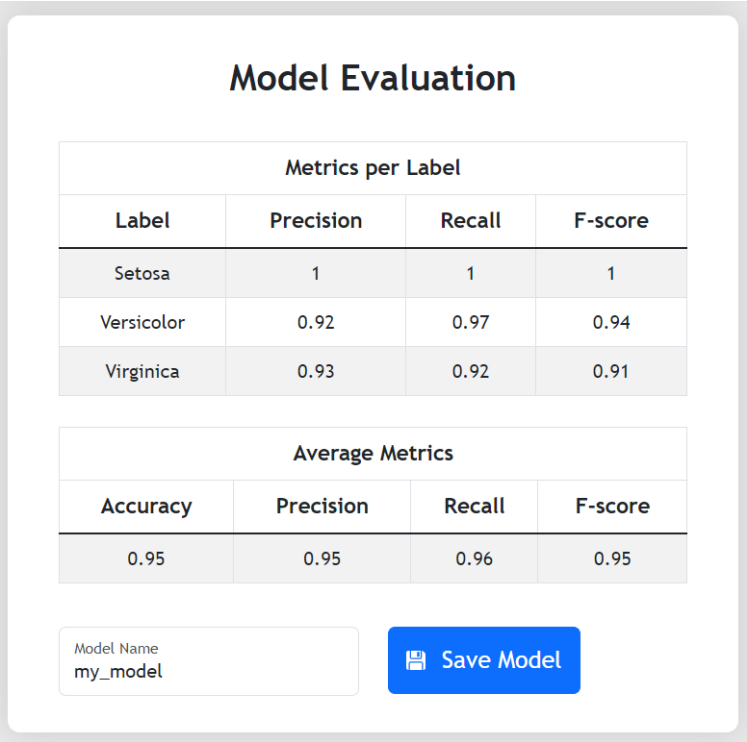


Σχήμα 5.16: Επιλογή παραμέτρων για τη δημιουργία μοντέλου



Σχήμα 5.17: Καθοδήγηση χρήστη μέσω tooltips

Μετά την συμπλήρωση των παραμέτρων, επιλέγοντας το κουμπί «Build Model», εκτελείται η διαδικασία δημιουργίας και αξιολόγησης του μοντέλου, μέσω της μεθόδου k-fold cross-validation. Με την ολοκλήρωσή της, επιστρέφονται τα αποτελέσματα των μετρικών απόδοσης, τόσο ανά label, όσο και συνολικά για το μοντέλο, κάτι που φαίνεται και στο Σχήμα 5.18. Τελικά, ο χρήστης μπορεί αν το επιθυμεί να αποθηκεύσει το μοντέλο στον λογαριασμό του, για να το αξιοποιήσει μελλοντικά. Για αυτόν το σκοπό, θα πρέπει πρώτα να δώσει ένα όνομα και να επιλέξει το κουμπί «Save Model».

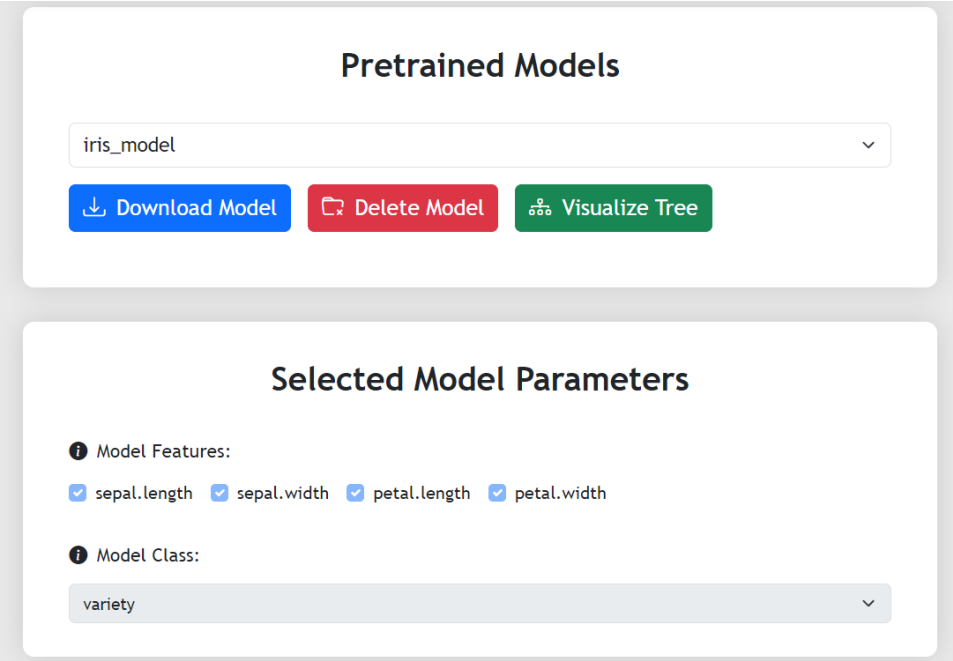


Σχήμα 5.18: Εμφάνιση μετρικών απόδοσης και αποθήκευση μοντέλου

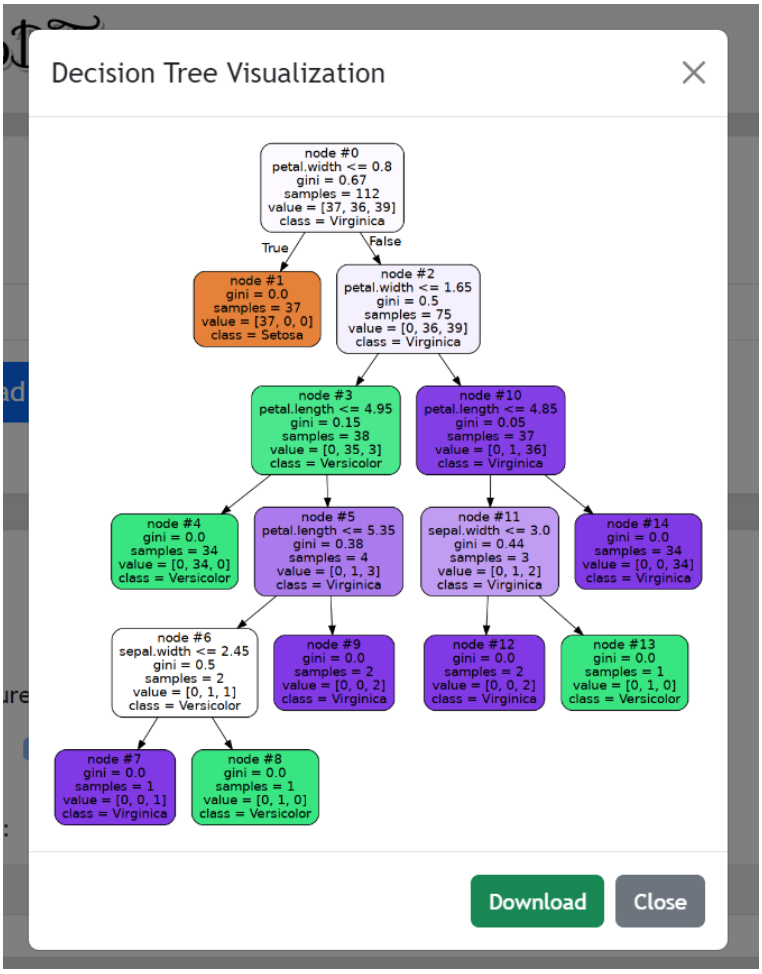
5.5 Σελίδα χρήσης προεκπαιδευμένων μοντέλων

Αρχικά, με την είσοδο στη συγκεκριμένη σελίδα, εμφανίζεται μία λίστα που περιέχει όλα τα προεκπαιδευμένα μοντέλα που έχει δημιουργήσει ο συγκεκριμένος χρήστης. Επιλέγοντας ένα από αυτά, εμφανίζεται το περιεχόμενο του μοντέλου, δηλαδή τα features και το class από τα οποία αποτελείται, ενώ υπάρχουν και τρία κουμπιά. Μέσω του πρώτου, δίνεται η δυνατότητα λήψης του μοντέλου σε μορφή αρχείου `'.pkl'`, το δεύτερο επιτρέπει την διαγραφή του, ενώ το τελευταίο χρησιμεύει στην οπτικοποίηση του Δέντρου Απόφασης (Σχήμα 5.19).

Ειδικότερα, επιλέγοντας το κουμπί «Visualize Tree» θα εμφανιστεί το αναλυτικό διάγραμμα Δέντρου Απόφασης του μοντέλου, επιτρέποντας στον χρήστη να το αποθηκεύσει σε μορφή εικόνας `'.png'` (Σχήμα 5.20).

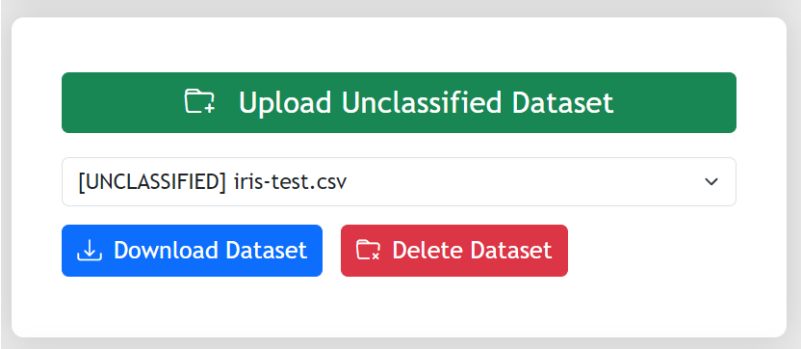


Σχήμα 5.19: Επιλογή μοντέλου και εμφάνιση του περιεχομένου του

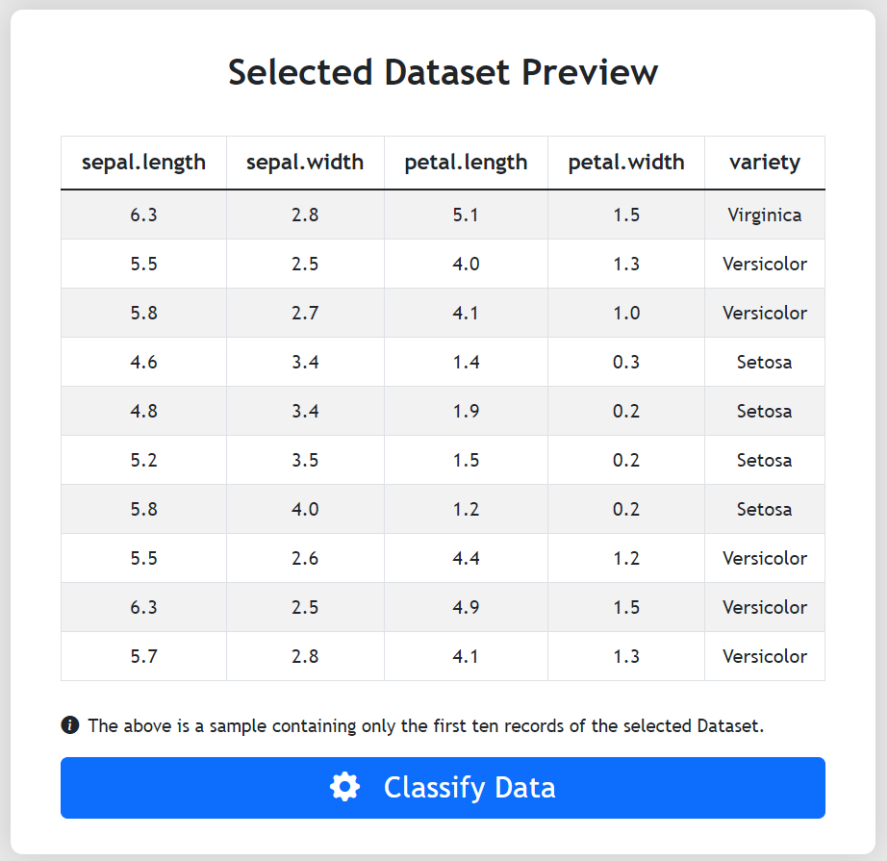


Σχήμα 5.20: Εμφάνιση διαγράμματος Δέντρου Απόφασης

Στη συνέχεια, ο χρήστης καλείται να προχωρήσει στη μεταφόρτωση και την επιλογή ενός συνόλου δεδομένων που περιέχει μη κατηγοριοποιημένα στιγμιότυπα (Σχήμα 5.21). Μετά την προεπισκόπηση του dataset (Σχήμα 5.22), επιλέγοντας το κουμπί «Classify Data», είναι εφικτή η εκτέλεση της διαδικασίας κατηγοριοποίησης των δεδομένων.



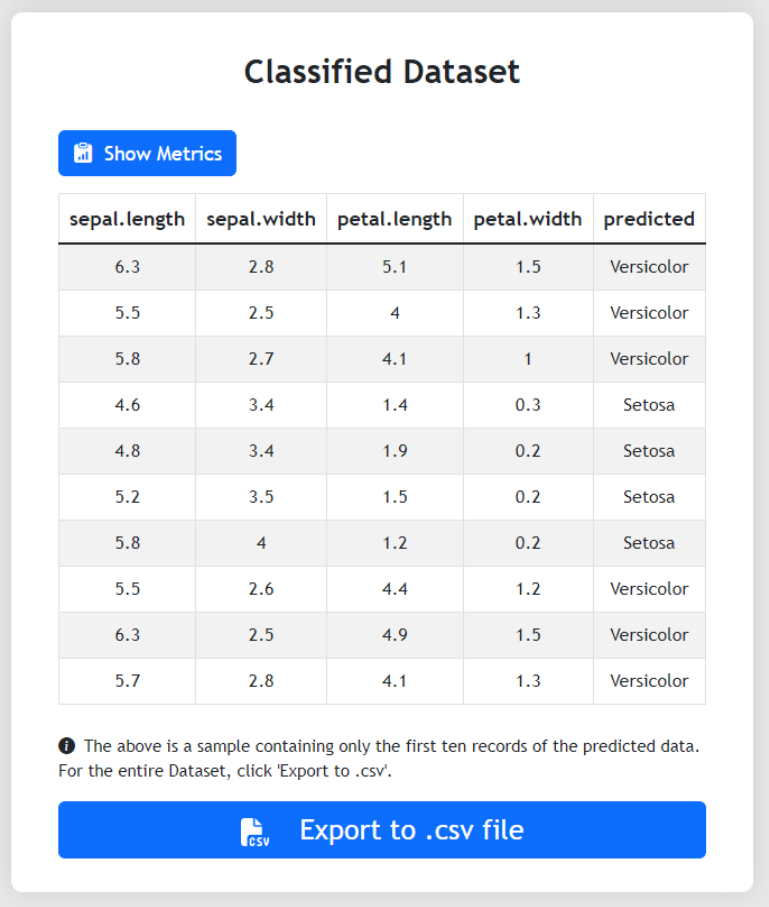
Σχήμα 5.21: Μενού χειρισμού μη κατηγοριοποιημένου dataset



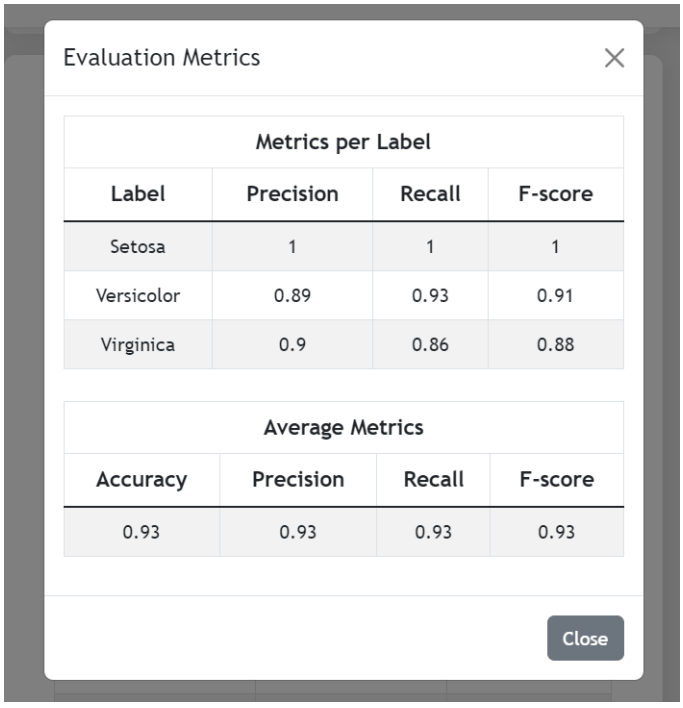
Σχήμα 5.22: Προεπισκόπηση μη κατηγοριοποιημένου dataset

Αφότου ολοκληρωθεί η εν λόγω διαδικασία, εμφανίζεται εκ νέου ο προηγούμενος πίνακας, έχοντας πλέον προστεθεί μία επιπλέον στήλη που περιέχει τα αποτελέσματα της κατηγοριοποίησης. Ο χρήστης μπορεί στη συνέχεια να εξάγει τα πλήρη αποτελέσματα σε μορφή αρχείου `'.csv'` (Σχήμα 5.23). Επιπρόσθετα, σε περίπτωση που το μη κατηγοριοποιημένο dataset που επιλέχθηκε, περιείχε ήδη και τη στήλη με την κλάση των παρατηρήσεων, τότε κάνοντας κλικ στο κουμπί «Show Metrics» είναι εφικτή η προβολή μετρικών, για την εκτίμηση της ποιότητας των αποτελεσμάτων (Σχήμα 5.24). Εφόσον δεν ικανοποιείται

η προαναφερθείσα συνθήκη, τότε το συγκεκριμένο κουμπί παραμένει απενεργοποιημένο.



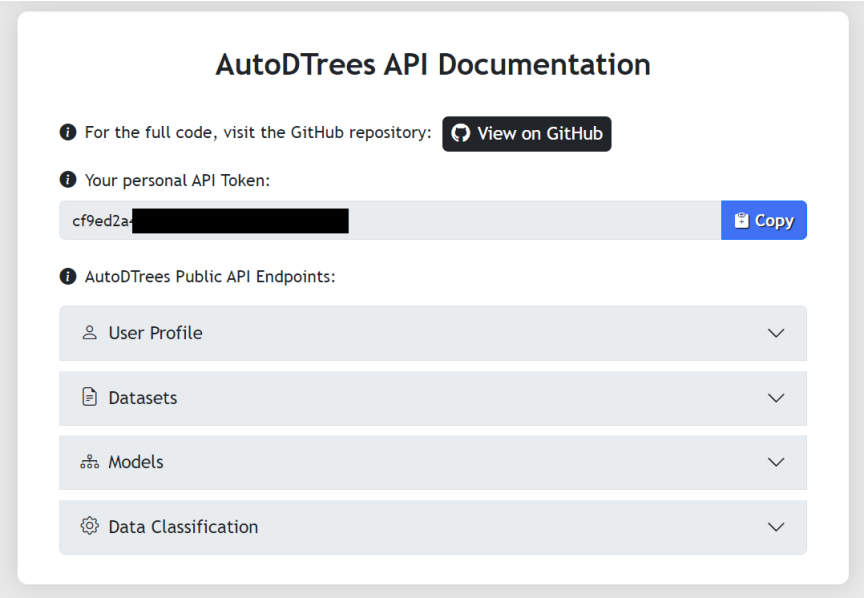
Σχήμα 5.23: Εμφάνιση αποτελεσμάτων κατηγοριοποίησης



Σχήμα 5.24: Προβολή μετρικών ποιότητας των αποτελεσμάτων

5.6 Σελίδα τεκμηρίωσης του Web API

Στο Σχήμα 5.25, φαίνεται η σελίδα που έχει δημιουργηθεί με σκοπό την τεκμηρίωση του Web API της εφαρμογής. Αρχικά, ο χρήστης μπορεί επιλέγοντας το σχετικό κουμπί να μεταφερθεί στο περιβάλλον της πλατφόρμας GitHub, όπου και φιλοξενείται το repository με τον πλήρη κώδικα της εφαρμογής. Ακολουθεί το πεδίο, από όπου μπορεί να προβάλλει και να αντιγράψει το προσωπικό Token, που του επιτρέπει την πρόσβαση στο Web API, ενώ στο κάτω τμήμα της σελίδας εμφανίζεται μία λίστα με όλα τα endpoints της εφαρμογής, ομαδοποιημένα ανάλογα με το είδος της λειτουργίας που εκτελούν. Κάνοντας κλικ σε κάποιο από αυτά, όπως φαίνεται και από το Σχήμα 5.26, εμφανίζεται μία σύντομη περιγραφή του, καθώς και παραδείγματα του τρόπου κλήσης και της απόκρισής του.



Σχήμα 5.25: Η σελίδα τεκμηρίωσης του Web API

Σχήμα 5.26: Προβολή λεπτομερειών endpoint

Κεφάλαιο 6ο: Αξιολόγηση του AutoDTrees

6.1 Εισαγωγή

Σε αυτό το Κεφάλαιο, θα γίνει παρουσίαση και ανάλυση των αποτελεσμάτων που προέκυψαν από τη διενέργεια αξιολόγησης της εφαρμογής, ως προς τον τομέα της εμπειρίας χρήστη.

Για τις ανάγκες αυτής της διαδικασίας, δημιουργήθηκε ένα σχετικό ερωτηματολόγιο με χρήση της υπηρεσίας Google Forms και κοινοποιήθηκε στους χρήστες της εφαρμογής AutoDTrees. Όσον αφορά τους τελευταίους, πρόκειται για άτομα που κατά κύριο λόγο είναι φοιτητές του Τμήματος Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων, με την πλειοψηφία αυτών να παρακολουθούν ή να έχουν ήδη παρακολουθήσει, κατά την περίοδο συμπλήρωσης του ερωτηματολογίου, το μάθημα Οργάνωση Δεδομένων και Εξόρυξη Πληροφορίας.

Το εν λόγω ερωτηματολόγιο, βασίζεται στην Κλίμακα Ευχρηστίας Συστήματος (System Usability Scale - SUS). Πρόκειται για μία μέθοδο, που επιτρέπει με έναν εύκολο και γρήγορο τρόπο την αξιολόγηση της ευχρηστίας ενός συστήματος. Αποτελείται από δέκα ερωτήσεις, κάθε μία από τις οποίες διαθέτει πέντε διαθέσιμες απαντήσεις που ακολουθούν τη διαβάθμιση «Διαφωνώ Απόλυτα», «Διαφωνώ», «Ούτε συμφωνώ ούτε διαφωνώ», «Συμφωνώ» και «Συμφωνώ απόλυτα» ή αντίστοιχα την αριθμητική κλίμακα 1 – 5. Πιο συγκεκριμένα, οι δέκα ερωτήσεις που συμπεριλαμβάνει η μέθοδος είναι οι ακόλουθες [41]:

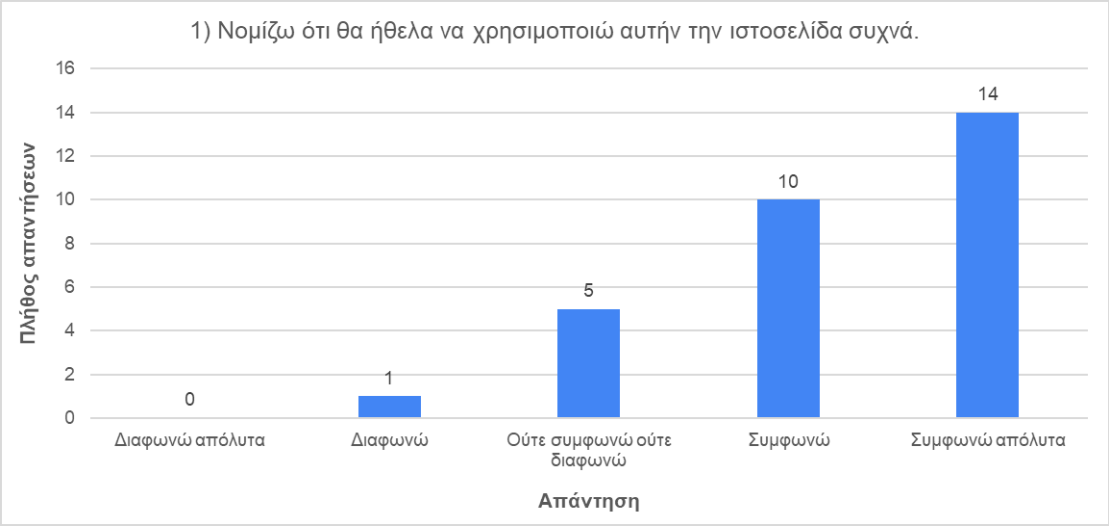
1. Νομίζω ότι θα ήθελα να χρησιμοποιώ αυτήν την ιστοσελίδα συχνά.
2. Βρήκα αυτήν την ιστοσελίδα αδικαιολόγητα περίπλοκη.
3. Σκέφτηκα ότι αυτή η ιστοσελίδα ήταν εύκολη στη χρήση.
4. Νομίζω ότι θα χρειαστώ βοήθεια για να είμαι σε θέση να χρησιμοποιήσω αυτήν την ιστοσελίδα.
5. Βρήκα τις διάφορες λειτουργίες σε αυτήν την ιστοσελίδα καλά ολοκληρωμένες.
6. Σκέφτηκα ότι υπήρχε μεγάλη ασυνέπεια σε αυτήν την ιστοσελίδα.
7. Φαντάζομαι ότι τα περισσότερα άτομα θα μάθουν να χρησιμοποιούν αυτήν την ιστοσελίδα πολύ γρήγορα.
8. Βρήκα αυτήν την ιστοσελίδα πολύ δύσκολη στη χρήση.
9. Ένιωσα σιγουριά χρησιμοποιώντας αυτήν την ιστοσελίδα.
10. Χρειάστηκε να μάθω πολλά πράγματα πριν να μπορέσω να ξεκινήσω με αυτήν την ιστοσελίδα.

6.2 Παρουσίαση αποτελεσμάτων

Όσον αφορά τα αποτελέσματα του ερωτηματολογίου της εφαρμογής, συνολικά υποβλήθηκαν απαντήσεις από 30 άτομα, με τα αναλυτικά στοιχεία που προέκυψαν να παρατίθενται ακολούθως.

Ερώτηση 1

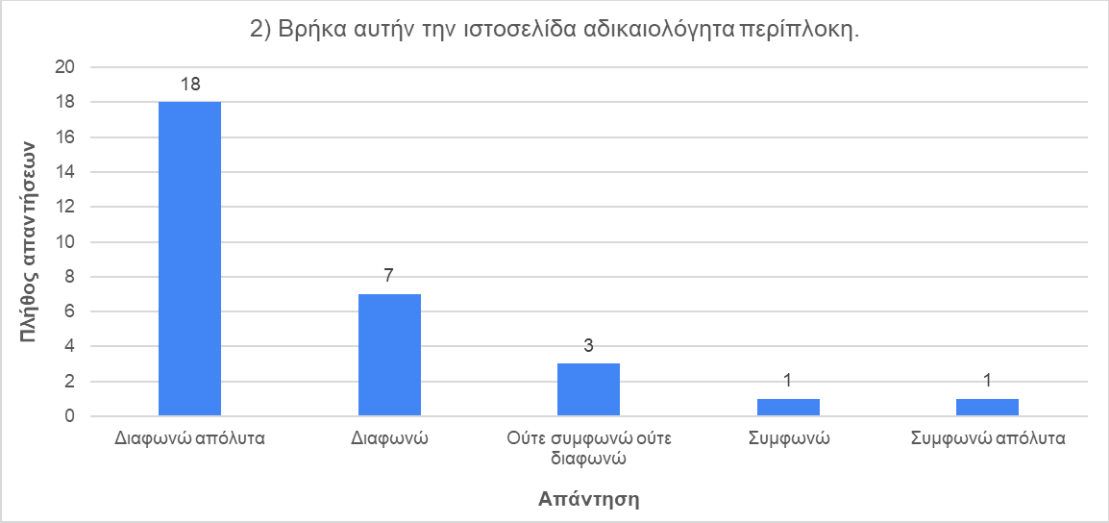
Στο Σχήμα 6.1 φαίνονται τα αποτελέσματα της Ερώτησης 1. Σύμφωνα με τους χρήστες, η πλειοψηφία αυτών δηλώνει ότι θα ήθελε να χρησιμοποιεί συχνά την εφαρμογή AutoDTrees, επιλέγοντας κυρίως τις απαντήσεις «Συμφωνώ» και «Συμφωνώ απόλυτα».



Σχήμα 6.1: Διάγραμμα απαντήσεων Ερώτησης 1

Ερώτηση 2

Όσον αφορά την Ερώτηση 2, οι χρήστες δηλώνουν κυρίως ότι «Διαφωνούν απόλυτα» ή απλώς «Διαφωνούν», στο αν βρίσκουν την ιστοσελίδα αδικαιολόγητα περίπλοκη, κάτι που φαίνεται και στο Σχήμα 6.2.



Σχήμα 6.2: Διάγραμμα απαντήσεων Ερώτησης 2

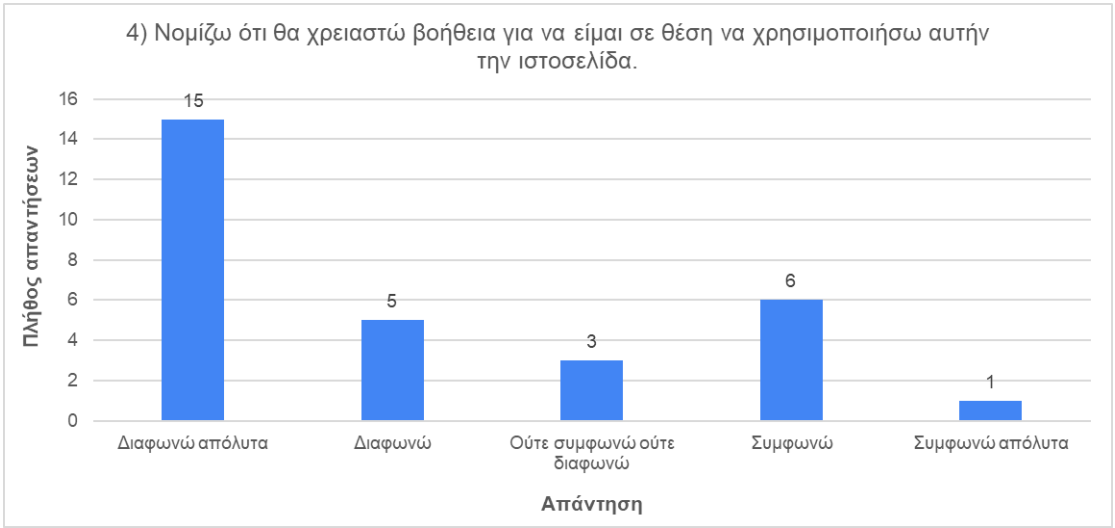
Ερώτηση 3

Στο Σχήμα 6.3 φαίνονται οι απαντήσεις για την Ερώτηση 3. Σύμφωνα με τα αποτελέσματα, οι χρήστες θεωρούν, σχεδόν κατά πλειοψηφία, ότι η εφαρμογή AutoDTrees είναι εύκολη στη χρήση.

Σχήμα 6.3: Διάγραμμα απαντήσεων Ερώτησης 3

Ερώτηση 4

Όσον αφορά την Ερώτηση 4 (Σχήμα 6.4), αρκετοί χρήστες δηλώνουν ότι διαφωνούν στο αν χρειάζονται βοήθεια για να χρησιμοποιήσουν την εφαρμογή, ωστόσο υπάρχει μία σημαντική μερίδα που είτε είναι αναποφάσιστοι είτε συμφωνούν.



Σχήμα 6.4: Διάγραμμα απαντήσεων Ερώτησης 4

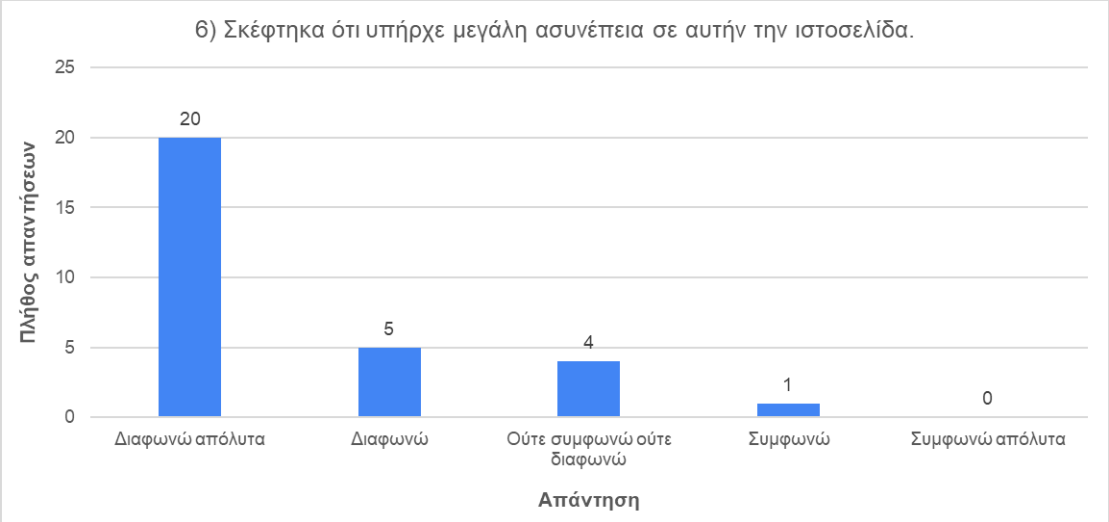
Ερώτηση 5

Στο Σχήμα 6.5 φαίνονται τα αποτελέσματα από την Ερώτηση 5. Σύμφωνα με αυτά, οι διάφορες λειτουργίες της ιστοσελίδας κρίνεται ότι είναι καλά ολοκληρωμένες.

Σχήμα 6.5: Διάγραμμα απαντήσεων Ερώτησης 5

Ερώτηση 6

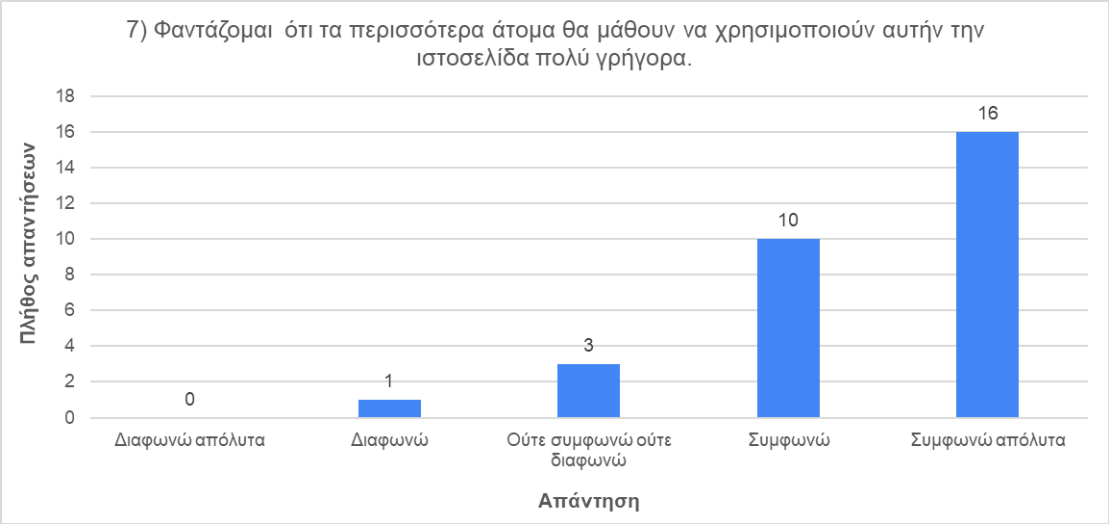
Όπως φαίνεται και από το Σχήμα 6.6, οι χρήστες κλήθηκαν στην Ερώτηση 6 να απαντήσουν, εάν κατά τη γνώμη τους υπήρχε μεγάλη ασυνέπεια στην ιστοσελίδα. Η πλειοψηφία δήλωσε ότι «Διαφωνεί απόλυτα», ενώ ακολουθούν τα άτομα που απλώς «Διαφωνούν» ή είναι αναποφάσιστα.



Σχήμα 6.6: Διάγραμμα απαντήσεων Ερώτησης 6

Ερώτηση 7

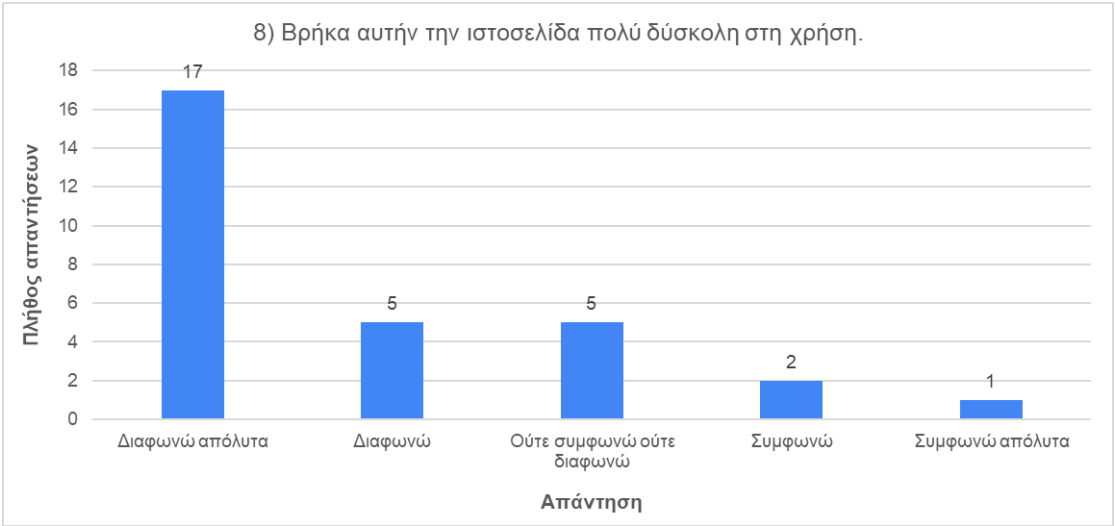
Στην Ερώτηση 7, ζητήθηκε από τους χρήστες να απαντήσουν, στο αν θεωρούν ότι τα περισσότερα άτομα θα μάθουν γρήγορα να χρησιμοποιούν την εφαρμογή. Σύμφωνα με το Σχήμα 6.7, το μεγαλύτερο μέρος των ερωτηθέντων «Συμφωνεί απόλυτα» ή απλώς «Συμφωνεί».



Σχήμα 6.7: Διάγραμμα απαντήσεων Ερώτησης 7

Ερώτηση 8

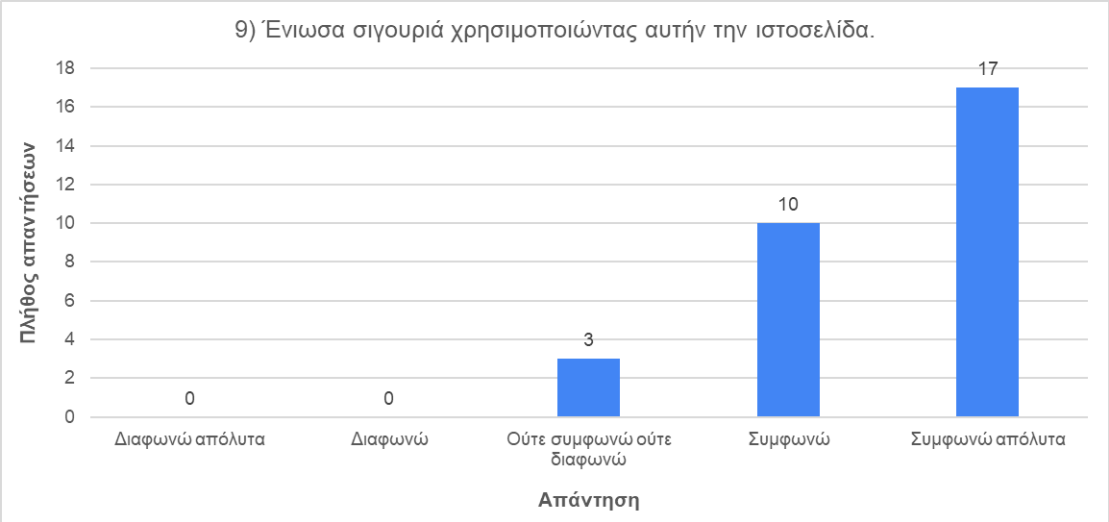
Σε ότι αφορά την Ερώτηση 8, εξετάζεται το κατά πόσο η ιστοσελίδα κρίνεται πολύ δύσκολη στη χρήση. Όπως φαίνεται και στο Σχήμα 6.8, οι ερωτηθέντες δηλώνουν σε μεγάλο βαθμό ότι «Διαφωνούν απόλυτα», ενώ ακολουθεί και μία σημαντική μερίδα ατόμων που απλώς «Διαφωνούν» ή είναι αναποφάσιστοι.



Σχήμα 6.8: Διάγραμμα απαντήσεων Ερώτησης 8

Ερώτηση 9

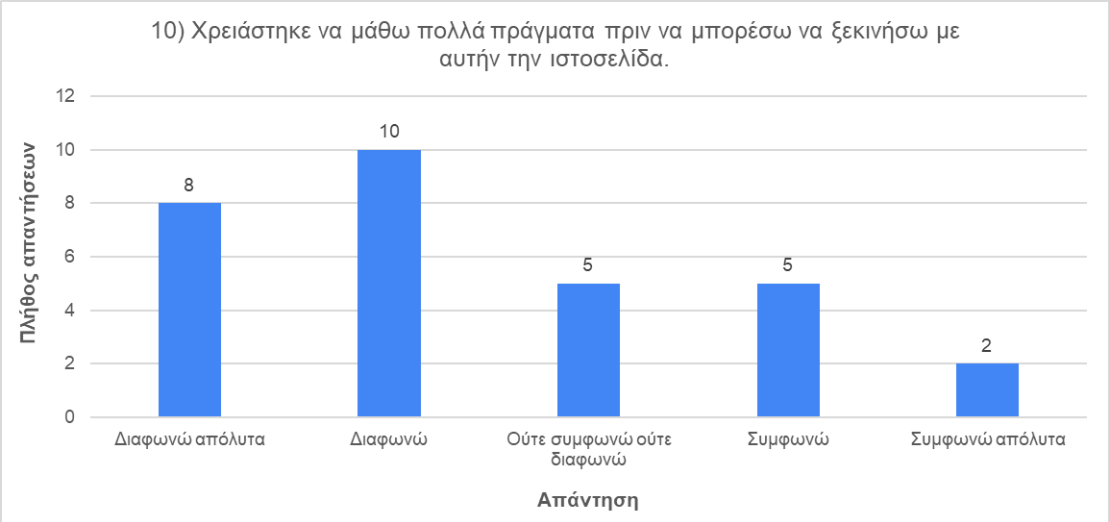
Στην Ερώτηση 9 (Σχήμα 6.9), οι χρήστες κλήθηκαν να απαντήσουν εάν ένιωσαν σίγουροι από τη χρήση του AutoDTrees, με την πλειοψηφία αυτών να εκφράζονται θετικά για την εφαρμογή.



Σχήμα 6.9: Διάγραμμα απαντήσεων Ερώτησης 9

Ερώτηση 10

Όσον αφορά την Ερώτηση 10, σε αυτήν τίθεται το ζήτημα του κατά πόσο απαιτείται η ύπαρξη προαπαιτούμενων γνώσεων, για την αποτελεσματική χρήση της εφαρμογής. Όπως φαίνεται και στο Σχήμα 6.10, πρόκειται για ένα ερώτημα, όπου οι απόψεις των χρηστών διίστανται. Ειδικότερα, τα περισσότερα άτομα «Διαφωνούν», με αρκετούς επίσης να «Διαφωνούν απόλυτα», ωστόσο μία αξιοσημείωτη μερίδα αυτών δηλώνουν είτε αναποφάσιστοι, είτε ότι «Συμφωνούν». Τα αποτελέσματα κρίνονται απόλυτα λογικά, καθώς σε ένα βαθμό οι χρήστες απαιτείται να έχουν βασικές γνώσεις στους τομείς της Εξόρυξης Δεδομένων και της Μηχανικής Μάθησης.



Σχήμα 6.10: Διάγραμμα απαντήσεων Ερώτησης 10

Τελική βαθμολογία

Για τον υπολογισμό της τελικής βαθμολογίας, που προκύπτει από το ερωτηματολόγιο SUS, ακολουθείται η εξής διαδικασία: Αρχικά, οι απαντήσεις μετατρέπονται στην κλίμακα 1 – 5 (1: Διαφωνώ Απόλυτα, 5: Συμφωνώ Απόλυτα). Στη συνέχεια, στις ερωτήσεις με μονό αριθμό (1,3,5,7,9) αφαιρείται ο αριθμός 1 από την απάντηση του χρήστη, ενώ σε εκείνες με ζυγό αριθμό (2,4,6,8,10) αφαιρείται η απάντηση του χρήστη από τον αριθμό 5. Ακολούθως, το άθροισμα των παραπάνω, πολλαπλασιάζεται επί 2,5 και προκύπτει ο βαθμός του κάθε χρήστη στην κλίμακα 0 – 100 [42]. Η τελική βαθμολογία, προκύπτει από τον μέσο όρο των επιμέρους βαθμών. Ένα σύστημα θεωρείται αποδεκτό, εφόσον η τελική του βαθμολογία είναι τουλάχιστον 68/100 [41].

Στον Πίνακα 6.1, φαίνεται ο συνολικός βαθμός του κάθε χρήστη, καθώς και η τελική βαθμολογία που συγκέντρωσε η εφαρμογή.

Πίνακας 6.1: Τελική βαθμολογία εφαρμογής

Χρήστης	**Βαθμολογία**	**Χρήστης**	**Βαθμολογία**
1	100	16	85
2	97,5	17	85
3	85	18	92,5
4	77,5	19	87,5
5	75	20	90
6	100	21	62,5
7	100	22	77,5
8	72,5	23	90
9	100	24	80
10	52,5	25	47,5
11	50	26	80
12	60	27	75
13	67,5	28	87,5
14	100	29	87,5
15	85	30	95
Τελικός Βαθμός:		**81,5**	

Κεφάλαιο 7ο: Συμπεράσματα και Μελλοντικές επεκτάσεις

7.1 Συμπεράσματα

Τα Δέντρα Απόφασης αποτελούν έναν εκ των δημοφιλέστερων αλγορίθμων κατηγοριοποίησης, κάτι που οφείλεται στην εύκολα κατανοητή δομή και στον απλό τρόπο λειτουργίας τους. Ωστόσο, κατόπιν σχετικής έρευνας διαπιστώθηκε ότι η χρήση τους καθίσταται αρκετές φορές δύσκολη, καθώς οι ενδιαφερόμενοι θα πρέπει είτε να διαθέτουν εξειδικευμένες γνώσεις, είτε να αξιοποιήσουν λύσεις λογισμικού, οι οποίες όμως είναι περιορισμένες και συχνά απαιτούν την απόκτηση συνδρομής.

Για την αντιμετώπιση αυτών των ζητημάτων, στα πλαίσια της παρούσας πτυχιακής εργασίας, αναπτύχθηκε η διαδικτυακή εφαρμογή «AutoDTrees». Πρόκειται για ένα ελεύθερο και ανοικτού κώδικα λογισμικό που επιτρέπει την εύκολη αξιοποίηση των Δέντρων Απόφασης. Ειδικότερα, παρέχει δυνατότητες όπως η δημιουργία μοντέλων με βάση τις προτιμήσεις του χρήστη, η εκτίμηση της απόδοσης, καθώς και η αποθήκευση των μοντέλων για μελλοντική αξιοποίηση, είτε για την πρόβλεψη μη κατηγοριοποιημένων στιγμιοτύπων, είτε για την οπτικοποίηση του Δέντρου Απόφασης. Η πρόσβαση στο AutoDTrees είναι εφικτή τόσο μέσω της διαθέσιμης γραφικής διεπαφής, όσο και μέσω ενός ελεύθερου Διαδικτυακού API, ενώ φιλοξενείται από server του Τμήματος Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων. Συμπερασματικά, η εφαρμογή μπορεί να αποτελέσει ένα χρήσιμο εργαλείο για μία ευρεία ομάδα ενδιαφερόμενων, ανεξαρτήτως της εμπειρίας που διαθέτουν, γεγονός που διαπιστώνεται και από τα αποτελέσματα της διαδικασίας αξιολόγησης από την πλευρά των χρηστών.

7.2 Μελλοντικές επεκτάσεις

Η διαδικτυακή εφαρμογή AutoDTrees, αποτελεί ένα ολοκληρωμένο περιβάλλον που προσφέρει ένα ευρύ φάσμα δυνατοτήτων και επιλογών, για την εκτέλεση του αλγορίθμου των Δέντρων Αποφάσεων. Ωστόσο, κρίνεται δυνατή η περαιτέρω βελτίωση και εξέλιξη της εφαρμογής, με την μελλοντική προσθήκη μιας σειράς από επιπλέον δυνατότητες. Μερικές από αυτές είναι οι ακόλουθες:

Ανάπτυξη εφαρμογής για κινητές συσκευές

Στα πλαίσια του γενικότερου σχεδιασμού ενός γραφικού περιβάλλοντος, φιλικού προς τον χρήστη, μία μελλοντική επέκταση θα μπορούσε να αποτελέσει η ανάπτυξη μίας έκδοσης του AutoDTrees για κινητές συσκευές (smartphone και tablet). Κάτι τέτοιο, θα διευκολύνει περαιτέρω τους χρήστες, ώστε να μπορούν να χρησιμοποιούν την εφαρμογή από κάθε είδος συσκευής που διαθέτουν.

Υποστήριξη διαφορετικών υλοποιήσεων των Δέντρων Απόφασης

Όπως σημειώθηκε σε προηγούμενο Κεφάλαιο, η παρούσα εφαρμογή αξιοποιεί τα Δέντρα Απόφασης μέσα από την υλοποίηση που παρέχει η βιβλιοθήκη Scikit-learn της Python, η οποία όμως δεν υποστηρίζει την χρήση ονομαστικών δεδομένων (categorical data). Ως εκ τούτου, μία μελλοντική επέκταση της εφαρμογής θα μπορούσε να αφορά την προσθήκη και άλλων υλοποιήσεων, όπως είναι οι αλγόριθμοι ID3 και C4.5 που επιτρέπουν τη χρήση ονομαστικών δεδομένων.

ΒΙΒΛΙΟΓΡΑΦΙΑ

[1] S. Akinola and O. Oyabugbe, "Accuracies and training times of data mining classification algorithms: An empirical comparative study," *Journal of Software Engineering and Applications*, pp. 470–477, 2015.

[2] H. A. Edelstein, *Introduction to Data Mining and Knowledge Discovery Third Edition*. Two Crows Corporation, 1999.

[3] Ε. Κύρκος, *Επιχειρηματική ευφυΐα και εξόρυξη δεδομένων*. Κάλλιπος, Ανοικτές Ακαδημαϊκές Εκδόσεις, 2015.

[4] P.-N. Tan, M. Steinbach, A. Karpatne, and V. Kumar, *Introduction to Data Mining (2nd Edition) - Instructor's Solution Manual*. Pearson Education, 2020.

[5] J. Han, M. Kamber, and J. Pei, *Data Mining: Concepts and Techniques (Third Edition)*. Morgan Kaufmann Series in Data Management Systems, Morgan Kaufmann publications, 2011.

[6] J. Delua, "Supervised vs. unsupervised learning: What's the difference?." `https://www.ibm.com/blog/supervised-vs-unsupervised-learning`. Accessed: 06-12-2023.

[7] Z. Keita, "Classification in machine learning: An introduction." `https://www.datacamp.com/blog/classification-machine-learning`. Accessed: 11-12-2023.

[8] A. Kumar, "Difference: Binary vs multiclass vs multilabel classification." `https://vitalflux.com/difference-binary-multi-class-multi-label-classification`. Accessed: 13-12-2023.

[9] D. D. Bhavani, A. Vasavi, and P. T. Keshava, "Machine learning: A critical review of classification techniques," *International Journal of Advanced Research in Computer and Communication Engineering (IJARCCE)*, vol. 5, no. 3, 2016.

[10] P.-N. Tan, M. Steinbach, and V. Kumar, *Introduction to Data Mining*. Pearson Education, 2014.

[11] A. A. Awan, "What is eager learning?." `https://www.datacamp.com/blog/what-is-eager-learning`. Accessed: 11-12-2023.

[12] A. A. Awan, "What is lazy learning?." `https://www.datacamp.com/blog/what-is-lazy-learning`. Accessed: 12-12-2023.

[13] I. Nti, O. Nyarko-Boateng, and J. Aning, "Performance of machine learning algorithms with different k values in k-fold cross-validation," *International Journal of Information Technology and Computer Science*, vol. 6, pp. 61–71, 12 2021.

[14] G. Cawley and N. Talbot, "Efficient leave-one-out cross-validation of kernel fisher discriminant classifiers," *Pattern Recognition*, vol. 36, pp. 2585–2592, 04 2003.

[15] T. Bahrynovska, "What is automated machine learning? going beyond the primer." `https://forbytes.com/blog/what-is-automl`. Accessed: 19-12-2023.

[16] V. Kanade, "What is automated machine learning (automl)? meaning, types, and functions." https://www.spiceworks.com/tech/artificial-intelligence/articles/what-is-automl. Accessed: 20-12-2023.

[17] B. Lutkevich, "Automated machine learning (automl)." https://www.techtarget.com/searchenterpriseai/definition/automated-machine-learning-AutoML. Accessed: 19-12-2023.

[18] IBM, "What is a decision tree?." https://www.ibm.com/topics/decision-trees. Accessed: 28-12-2023.

[19] T. Plapinger, "What is a decision tree?." https://towardsdatascience.com/what-is-a-decision-tree-22975f00f3e1. Accessed: 28-12-2023.

[20] Scikit-Learn, "Decision trees." https://scikit-learn.org/stable/modules/tree.html. Accessed: 29-12-2023.

[21] Π. Συμεωνίδης, *Ευφυή Συστήματα Συστάσεων*. Κάλλιπος, Ανοικτές Ακαδημαϊκές Εκδόσεις, 2023.

[22] Β. Βερύκιος, Β. Καγκλής και Η. Σταυρόπουλος, *Η επιστήμη των δεδομένων μέσα από τη γλώσσα R*. Κάλλιπος, Ανοικτές Ακαδημαϊκές Εκδόσεις, 2015.

[23] N. Mantri, "Using id3 algorithm to build a decision tree to predict the weather." https://iq.opengenus.org/id3-algorithm. Accessed: 02-01-2024.

[24] S. Singh and M. Giri, "Comparative study id3, cart and c4.5 decision tree algorithm: A survey," *International Journal of Advanced Information Science and Technology (IJAIST)*, vol. 3, pp. 47–52, 07 2014.

[25] B. Hssina, A. Merbouha, H. Ezzikouri, and M. Erritali, "A comparative study of decision tree id3 and c4.5," *International Journal of Advanced Computer Science and Applications*, vol. 4, no. 2, pp. 13–19, 2014.

[26] IBM, "What is an api?." https://www.ibm.com/topics/api. Accessed: 05-01-2024.

[27] IBM, "What is a rest api?." https://www.ibm.com/topics/rest-apis. Accessed: 05-01-2024.

[28] R. Toal, "A comprehensive guide to php programming: What you need to know." https://codeinstitute.net/global/blog/what-is-php-programming. Accessed: 06-01-2024.

[29] PHP.net, "PHP Documentation." https://www.php.net/manual/en. Accessed: 06-01-2024.

[30] Composer, "Composer Documentation." https://getcomposer.org/doc/00-intro.md. Accessed: 06-01-2024.

[31] MySQL, "MySQL Documentation." https://dev.mysql.com/doc/refman/8.0/en/what-is-mysql.html. Accessed: 07-01-2024.

[32] Amazon Web Services, "What is sql (structured query language)?." https://aws.amazon.com/what-is/sql. Accessed: 07-01-2024.

[33] Python.org, "Python Documentation." https://docs.python.org/3/index.html. Accessed: 07-01-2024.

[34] Coursera.org, "What is python used for? a beginner's guide." https://www.coursera.org/articles/what-is-python-used-for-a-beginners-guide-to-using-python. Accessed: 07-01-2024.

[35] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and Édouard Duchesnay, "Scikit-learn: Machine learning in python," *Journal of Machine Learning Research*, vol. 12, no. 85, pp. 2825–2830, 2011.

[36] Mozilla Developer Network (MDN) Web Docs, "HTML basics." https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/HTML_basics. Accessed: 08-01-2024.

[37] Mozilla Developer Network (MDN) Web Docs, "CSS basics." https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/CSS_basics. Accessed: 09-01-2024.

[38] J. Alexandrea, "What is bootstrap?." https://www.hostinger.com/tutorials/what-is-bootstrap. Accessed: 09-01-2024.

[39] S. Miller, "What is javascript used for?." https://www.codecademy.com/resources/blog/what-is-javascript-used-for. Accessed: 09-01-2024.

[40] J. Alexandrea, "What is jQuery? a beginner's introduction to the jQuery library." https://www.hostinger.com/tutorials/what-is-jquery. Accessed: 10-01-2024.

[41] Usability.gov, "System Usability Scale (SUS)." https://www.usability.gov/how-to-and-tools/methods/system-usability-scale.html. Accessed: 17-01-2024.

[42] J. Brooke, "SUS: A quick and dirty usability scale," *Usability Eval. Ind.*, vol. 189, 11 1995.