

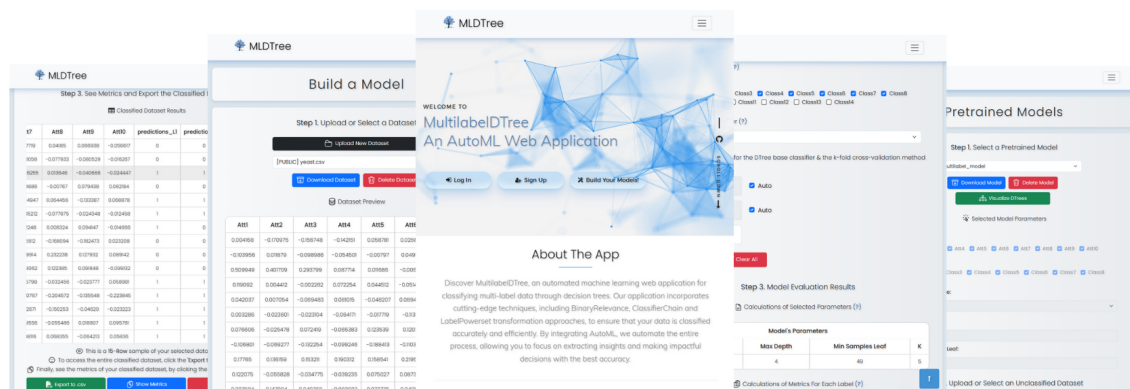


ΔΙΕΘΝΕΣ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΤΗΣ ΕΛΛΑΔΟΣ

Διεθνές Πανεπιστήμιο Ελλάδος
Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

MultilabelDTree: Εφαρμογή ιστού αυτοματοποιημένης μηχανικής μάθησης για την κατηγοριοποίηση δεδομένων πολλαπλών ετικετών μέσω δέντρων απόφασης



Φοιτητής:
Αλέξανδρος Οικονόμου
Αριθμός Μητρώου: 2019119

Επιβλέπων:
Στέφανος Ουγιάρογλου

31 Ιανουαρίου 2025

MultilabelDTree: Εφαρμογή ιστού αυτοματοποιημένης μηχανικής μάθησης για την κατηγοριοποίηση δεδομένων πολλαπλών ετικετών μέσω δέντρων απόφασης

Κωδικός Δ.Ε.: 24215

Όνοματεπώνυμο φοιτητή: Αλέξανδρος Οικονόμου

Όνοματεπώνυμο εισηγητή: Στέφανος Ουγιάρογλου

Ημερομηνία ανάληψης: 28-06-2024

Ημερομηνία περάτωσης: 31-01-2025

Βεβαιώνω ότι είμαι ο συγγραφέας αυτής της εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, έχω καταγράψει τις όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών, εικόνων και κειμένου, είτε αυτές αναφέρονται ακριβώς είτε παραφρασμένες. Επιπλέον, βεβαιώνω ότι αυτή η εργασία προετοιμάστηκε από εμένα προσωπικά, ειδικά ως διπλωματική εργασία, στο Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του ΔΙ.ΠΑ.Ε.

Η παρούσα εργασία αποτελεί πνευματική ιδιοκτησία του φοιτητή Αλέξανδρου Οικονόμου που την εκτόνησε. Στο πλαίσιο της πολιτικής ανοικτής πρόσβασης, ο συγγραφέας/δημιουργός εκχωρεί στο Διεθνές Πανεπιστήμιο της Ελλάδος άδεια χρήσης του δικαιώματος αναπαραγωγής, δανεισμού, παρουσίασης στο κοινό και ψηφιακής διάχυσης της εργασίας διεθνώς, σε ηλεκτρονική μορφή και σε οποιοδήποτε μέσο, για διδακτικούς και ερευνητικούς σκοπούς, άνευ ανταλλάγματος. Η ανοικτή πρόσβαση στο πλήρες κείμενο της εργασίας, δεν σημαίνει καθ' οιονδήποτε τρόπο παραχώρηση δικαιωμάτων διανοητικής ιδιοκτησίας του συγγραφέα/δημιουργού, ούτε επιτρέπει την αναπαραγωγή, αναδημοσίευση, αντιγραφή, πώληση, εμπορική χρήση, διανομή, έκδοση, μεταφόρτωση (downloading), ανάρτηση (uploading), μετάφραση, τροποποίηση με οποιονδήποτε τρόπο, τμηματικά ή περιληπτικά της εργασίας, χωρίς τη ρητή προηγούμενη έγγραφη συναίνεση του συγγραφέα/δημιουργού.

Η έγκριση της διπλωματικής εργασίας από το Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του Διεθνούς Πανεπιστημίου της Ελλάδος, δεν υποδηλώνει απαραίτητως και αποδοχή των απόψεων του συγγραφέα, εκ μέρους του Τμήματος.

Αφιέρωση

Αρχικά, θα ήθελα να ευχαριστήσω την οικογένειά μου για την τεράστια ενθάρρυνση και υποστήριξη που μου πρόσφερε καθ'όλη τη διάρκεια των σπουδών μου, αλλά και εκείνους που μου στάθηκαν ως φίλοι όλο το διάστημα των υπέροχων φοιτητικών μου χρόνων.

Τέλος, ιδιαίτερες ευχαριστίες οφείλονται στον επιβλέποντα της διπλωματικής μου εργασίας, τον Επίκουρο Καθηγητή κ. Στέφανο Ουγιάρογλου, για την πολύτιμη καθοδήγηση και εμπιστοσύνη που μου έδειξε κατά τη διάρκεια εκπόνησης της εργασίας.

Abstract

Classification is a fundamental technique in Machine Learning and Data Mining, aimed at grouping data into predefined categories. Multi-label classification extends this concept by allowing a data instance to belong to multiple categories simultaneously. To address this need, the «MultilabelDTree» application was developed as an open-source tool that facilitates the creation of the multi-label classification models, even for users with limited experience. Users can upload their own data by selecting the desired transformation technique (BinaryRelevance, LabelPowerset, or ClassifierChain), and by configuring appropriate parameters. The application also enables model performance evaluation through the k-fold cross-validation method, providing detailed results both overall and per category. Additionally, it supports saving models for future use, visualizing Decision Trees, and activating Auto Mode to automatically determine the best transformation method and Decision Tree parameters. Furthermore, the application operates via a Web API, enabling integration and extension into other applications. With a user-friendly interface, it is hosted on the server at the Department of Information and Electronic Engineering, and its source code is available on GitHub, addressing the needs of various users and organizations.

Περίληψη

Η κατηγοριοποίηση (classification) είναι μια βασική τεχνική της Μηχανικής Μάθησης και Εξόρυξης Δεδομένων, που αποσκοπεί στην ομαδοποίηση δεδομένων σε προκαθορισμένες κατηγορίες. Η κατηγοριοποίηση πολλαπλών ετικετών (multi-label classification) επεκτείνει αυτή την έννοια, επιτρέποντας σε ένα στιγμιότυπο να ανήκει ταυτόχρονα σε περισσότερες από μία κατηγορίες. Έτσι, για τον σκοπό αυτό δημιουργήθηκε η εφαρμογή «MultilabelDTree», η οποία είναι ένα εργαλείο ανοικτού κώδικα που διευκολύνει τη δημιουργία μοντέλων κατηγοριοποίησης πολλαπλών ετικετών, ακόμη και από χρήστες με περιορισμένη εμπειρία. Οι χρήστες μπορούν να ανεβάζουν τα δικά τους δεδομένα, να επιλέγουν την τεχνική μετασχηματισμού που επιθυμούν (BinaryRelevance, LabelPowerset ή ClassifierChain), καθώς και να ρυθμίζουν τις κατάλληλες παραμέτρους. Παρέχεται επίσης η δυνατότητα αξιολόγησης της απόδοσης του μοντέλου μέσω της μεθόδου k-fold cross-validation, προσφέροντας αναλυτικά αποτελέσματα, τόσο συνολικά όσο και ανά κατηγορία. Επιπρόσθετα, η εφαρμογή υποστηρίζει αποθήκευση των μοντέλων για μελλοντική χρήση, οπτικοποίηση των Δέντρων Απόφασης, καθώς και ενεργοποίηση της αυτόματης λειτουργίας (Auto Mode) για την εύρεση της καλύτερης μεθόδου μετασχηματισμού αλλά και των παραμέτρων του δέντρου απόφασης. Επιπλέον, η εφαρμογή λειτουργεί μέσω ενός Web API, επιτρέποντας ενσωμάτωση και επέκταση σε άλλες εφαρμογές. Η εφαρμογή διαθέτει και μία φιλική προς τον χρήστη διεπαφή και φιλοξενείται σε server του Τμήματος Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων, ενώ ο κώδικάς της είναι διαθέσιμος στο GitHub, καλύπτοντας ανάγκες διαφόρων χρηστών και οργανισμών.

Περιεχόμενα

Αφιέρωση	iii
Περίληψη	v
Περιεχόμενα	v
Κατάλογος Σχημάτων	2
Κατάλογος Πινάκων	3
1 Εισαγωγή	5
1.1 Εισαγωγή στην Κατηγοριοποίηση Πολλαπλών Ετικετών	5
1.2 Αυτοματοποιημένη Μηχανική Μάθηση (AutoML)	9
1.3 Κίνητρο	12
1.4 Συνεισφορά	13
1.5 Οργάνωση της Διπλωματικής	14
2 Κατηγοριοποίηση Πολλαπλών Ετικετών	16
2.1 Μέθοδοι και Αλγόριθμοι	16
2.2 Binary Relevance	19
2.3 Label Powerset	20
2.4 Classifier Chain	22
2.5 Μέτρα Απόδοσης	23
3 Τεχνολογίες	25
3.1 Backend	25
3.1.1 REST API	25
3.1.2 PHP	26
3.1.3 MySQL	27
3.1.4 Python	28
3.1.5 Composer και PHPMailer	29
3.1.6 Scikit-learn και Scikit-Multilearn	30
3.2 Frontend	31
3.2.1 HTML	31
3.2.2 CSS	32
3.2.3 Bootstrap	33
3.2.4 JavaScript	34
3.2.5 JQuery	34
4 Σχεδίαση και Υλοποίηση του MultilabelDTree	36

4.1	Λειτουργικές Απαιτήσεις	36
4.2	Αρχιτεκτονική της Εφαρμογής	38
4.3	Τύποι Συνόλων Δεδομένων	42
4.4	Χρήστες της Εφαρμογής	42
4.5	Υλοποίηση του Backend	43
4.5.1	Βάση Δεδομένων	43
4.5.2	Web API	47
4.5.3	Χειρισμός Multi-label Μοντέλων	55
4.5.4	Χειρισμός Προεκπαιδευμένων Μοντέλων	60
4.6	Υλοποίηση του Frontend	64
5	Παρουσίαση του MultilabelDTree	69
5.1	Αρχική Σελίδα της Εφαρμογής	69
5.2	Sign Up και Log In του Χρήστη	71
5.3	Ανάκτηση Κωδικού και Επεξεργασία Λογαριασμού	72
5.4	Δημιουργία Multi-label Μοντέλων	75
5.5	Χρήση Προεκπαιδευμένων Μοντέλων	81
6	Συμπεράσματα και Μελλοντικές Επεκτάσεις	87
6.0.1	Συμπεράσματα	87
6.0.2	Μελλοντικές Επεκτάσεις	87

Κατάλογος Σχημάτων

1.1	Παράδειγμα Κατηγοριοποίησης Πολλαπλών Ετικετών.	6
1.2	Ροή Εργασίας Εφαρμογών Μηχανικής Μάθησης με AutoML.	10
2.1	Ταξινόμηση Μεθόδων Κατηγοριοποίησης Πολλαπλών Ετικετών.	17
2.2	Σύνολο Δεδομένων Πολλαπλών Ετικετών.	19
2.3	BR Μετασχηματισμός του Συνόλου Δεδομένων.	20
2.4	LP Μετασχηματισμός του Συνόλου Δεδομένων	21
3.1	Τα Πλεονεκτήματα της PHP.	27
3.2	Τα Χαρακτηριστικά της MySQL.	28
3.3	Τα Βασικά Χαρακτηριστικά της Python.	29
4.1	Διάγραμμα Ροής του MultilabelDTree.	40
4.2	Αρχιτεκτονική του MultilabelDTree.	41
4.3	ER Diagram της Βάσης Δεδομένων.	46
4.4	Αρχεία HTML Κώδικα για τις Σελίδες της Εφαρμογής	64
4.5	Αρχεία CSS Κώδικα για τις Σελίδες της Εφαρμογής	65
4.6	Αρχεία JavaScript Κώδικα της Εφαρμογής	66
5.1	Αρχική Σελίδα του «MultilabelDTree»	69
5.2	Δυνατότητες και Λειτουργίες του «MultilabelDTree»	70
5.3	Φωτογραφίες και Υποσέλιδο του «MultilabelDTree»	70
5.4	Σελίδα Εγγραφής του «MultilabelDTree»	71
5.5	Σελίδα Εισόδου του «MultilabelDTree»	72
5.6	Σελίδα Ανάκτησης Κωδικού Λογαριασμού του «MultilabelDTree»	73
5.7	Σελίδα Επεξεργασίας Στοιχείων Λογαριασμού του «MultilabelDTree»	73
5.8	Σελίδα Επεξεργασίας Email Λογαριασμού του «MultilabelDTree»	74
5.9	Σελίδα Διαγραφής Λογαριασμού του «MultilabelDTree»	74
5.10	Σελίδα Δημιουργίας Multi-label Μοντέλων του «MultilabelDTree»	75
5.11	Λίστα για την Επιλογή ενός Multi-label Συνόλου Δεδομένων	76
5.12	Modal για το Ανέβασμα ενός Multi-label Συνόλου Δεδομένων	76
5.13	Πίνακας για το Preview ενός Multi-label Συνόλου Δεδομένων	77
5.14	Επιλογή Παραμέτρων για τη Δημιουργία του Multi-label Μοντέλου	78
5.15	Παράδειγμα Βοηθητικού Μηνύματος (tooltip) για τον Χρήστη	78
5.16	Αποτελέσματα Αξιολόγησης του Multi-label Μοντέλου	79
5.17	Αποθήκευση του Multi-label Μοντέλου	80

5.18	Μήνυμα Προτροπής για τον Χρήστη	80
5.19	Σελίδα Προεκπαιδευμένων Μοντέλων του MultilabelIDTree	81
5.20	Λίστα για την Επιλογή ενός Προεκπαιδευμένου Μοντέλου	82
5.21	Προβολή Επιλεγμένων Παραμέτρων του Μοντέλου	82
5.22	Οπτικοποίηση Δέντρων Απόφασης μέσω Modal	83
5.23	Modal για το Ανέβασμα ενός Αταξινόμητου Συνόλου Δεδομένων	84
5.24	Λίστα για την Επιλογή ενός Αταξινόμητου Συνόλου Δεδομένων	84
5.25	Πίνακας για το Preview ενός Αταξινόμητου Συνόλου Δεδομένων	85
5.26	Πίνακας Ταξινομημένων Στιγμιότυπων του Συνόλου Δεδομένων	86
5.27	Modal με τις Μετρικές Αξιολόγησης του Ταξινομημένου Συνόλου	86

Κατάλογος Πινάκων

1.1	Διαφορές μεταξύ Single-Label και Multi-Label Classification	7
4.1	Πίνακας users της Βάσης Δεδομένων	44
4.2	Πίνακας verify_account της Βάσης Δεδομένων	45
4.3	Πίνακας models της Βάσης Δεδομένων	45
4.4	Πίνακας labels της Βάσης Δεδομένων	46
4.5	Public API Endpoints της Εφαρμογής	48

Κεφάλαιο 1

Εισαγωγή

1.1 Εισαγωγή στην Κατηγοριοποίηση Πολλαπλών Ετικετών

Η ταξινόμηση ή κατηγοριοποίηση (classification) αποτελεί μία από τις πιο σημαντικές τεχνικές στον τομέα της Μηχανικής Μάθησης (Machine Learning) και της Εξόρυξης Δεδομένων (Data Mining). Κύριος στόχος της είναι η ομαδοποίηση των δεδομένων σε προκαθορισμένες κατηγορίες με τρόπο που κάθε στοιχείο του συνόλου αποδίδεται στη σωστή ομάδα. Με άλλα λόγια, η ταξινόμηση επιδιώκει την ακριβή πρόβλεψη της κατάλληλης κατηγορίας για κάθε περίπτωση στα διαθέσιμα δεδομένα, στηριζόμενη σε μοτίβα και σχέσεις που εντοπίζονται στα χαρακτηριστικά των δειγμάτων. Ένα παράδειγμα εφαρμογής αυτής της τεχνικής είναι η αξιολόγηση αιτήσεων δανείων, όπου ένα μοντέλο ταξινόμησης θα μπορούσε να εκπαιδευτεί έτσι ώστε να εντοπίζει και να κατατάσσει τους υποψήφιους δανειολήπτες σε κατηγορίες χαμηλού, μεσαίου ή υψηλού πιστωτικού κινδύνου [1]. Ένας μαθηματικός ορισμός της κατηγοριοποίησης δεδομένων μπορεί να διατυπωθεί ως εξής:

Ορισμός 1 (classification). *Δεδομένης μιας βάσης δεδομένων D με n εγγραφές (γραμμές) και d χαρακτηριστικά (στήλες), καθώς και μίας ετικέτας κλάσης (class-label) που αντιστοιχεί σε κάθε μία από τις n εγγραφές της βάσης D , ζητούμε να κατασκευαστεί ένα μοντέλο εκπαίδευσης M που θα μπορεί να χρησιμοποιηθεί για την πρόβλεψη της ετικέτας-κλάσης μιας νέας εγγραφής X , η οποία διαθέτει επίσης d -διαστάσεις [2].*

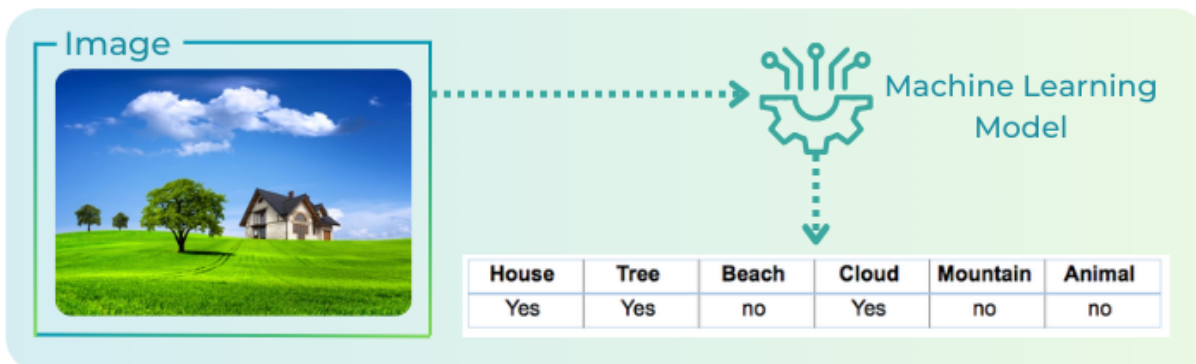
Στην παραδοσιακή ταξινόμηση με μία ετικέτα (single-label classification), κάθε στοιχείο μπορεί να ανήκει σε μία μόνο κατηγορία από ένα προκαθορισμένο σύνολο κατηγοριών. Αυτός ο τύπος ταξινόμησης χωρίζεται ευρέως σε δύο βασικές μεθόδους [3]:

- **Διαδική Κατηγοριοποίηση (Binary Classification):** Σε αυτή την μέθοδο, το μοντέλο μπορεί να αποφασίσει μεταξύ δύο πιθανών αποτελεσμάτων (0 ή 1). Ένα χαρακτηριστικό παράδειγμα είναι η ανίχνευση ανεπιθύμητων email, όπου ένα σύστημα αποφασίζει εάν το μήνυμα είναι ανεπιθύμητο (1) ή όχι (0).
- **Κατηγοριοποίηση Πολλαπλών Κατηγοριών (Multi-class Classification):** Εδώ το μοντέλο καλείται να επιλέξει μία μόνο κατηγορία από πολλές αμοιβαία αποκλειόμενες. Για παράδειγμα, ένα σύστημα αναγνώρισης φρούτων θα μπορούσε να ταξινομήσει μια εικόνα ως "μήλο", "πορτοκάλι" ή "μπανάνα", επιλέγοντας δηλαδή αποκλειστικά μία από αυτές τις κατηγορίες.

Παρόλο που οι παραδοσιακές μέθοδοι ταξινόμησης αποδίδουν ικανοποιητικά σε ορισμένα προβλήματα, πολλά σενάρια του πραγματικού κόσμου απαιτούν την ταυτόχρονη αντιστοίχιση ενός αντικειμένου σε περισσότερες από μία κατηγορίες. Σε αυτές τις περιπτώσεις, η κατηγοριοποίηση πολλαπλών ετικετών καθύσταται αναγκαία.

Η **Κατηγοριοποίηση Πολλαπλών Ετικετών (Multi-label Classification)** αποτελεί μία επέκταση της παραδοσιακής ταξινόμησης και επιτρέπει σε ένα στιγμιότυπο ενός συνόλου δεδομένων να ανήκει ταυτόχρονα σε πολλαπλές ετικέτες-κατηγορίες. Αυτό αντανακλά την πολυπλοκότητα των δεδομένων του πραγματικού κόσμου, όπου οι κατηγορίες δεν είναι πάντα αμοιβαία αποκλειστικές. Για παράδειγμα, μια ταινία θα μπορούσε να χαρακτηριστεί τόσο ως "Δράση" όσο και ως "Κωμωδία", ενώ μια ιατρική διάγνωση θα μπορούσε να ενδείξει ότι ένας ασθενής πάσχει ταυτόχρονα και από "Διαβήτη" και από "Υπέρταση". Αξίζει να σημειωθεί, πως η ανάγκη για μεθόδους πολλαπλών ετικετών γίνεται όλο και πιο εμφανής στις σύγχρονες εφαρμογές και τεχνολογίες, καθιστώντας την ταξινόμηση έναν ακρογωνιαίο λίθο της σύγχρονης ανάλυσης δεδομένων. Ενδεικτικά παραδείγματα σύγχρονων εφαρμογών, που χρησιμοποιούν την τεχνική της κατηγοριοποίησης πολλαπλών ετικετών, είναι η ταξινόμηση λειτουργιών των πρωτεϊνών στον κλάδο της βιοπληροφορικής, η κατηγοριοποίηση μουσικών κομματιών στον τομέα της μουσικής πληροφορικής καθώς και η αναγνώριση σημασιολογικών σκηνών στον χώρο της υπολογιστικής όρασης [4].

Στο Σχήμα 1.1 παρουσιάζεται ένα παράδειγμα κατηγοριοποίησης πολλαπλών ετικετών, όπου δίνεται μια εικόνα και ζητείται να απαντηθεί το εξής ερώτημα: "Ποιες είναι όλες οι ετικέτες (ή κατηγορίες) που σχετίζονται με την συγκεκριμένη εικόνα;" [5].



Σχήμα 1.1: Παράδειγμα Κατηγοριοποίησης Πολλαπλών Ετικετών.

Όπως διακρίνεται στο παραπάνω σχήμα, το μοντέλο μηχανικής μάθησης που εφαρμόζει κατηγοριοποίηση πολλαπλών ετικετών κατάφερε να αναγνωρίσει με ακρίβεια ότι η εικόνα περιλαμβάνει τρεις διαφορετικές ετικέτες: Σπίτι (House), Δέντρο (Tree) και Σύννεφο (Cloud). Αντίθετα, το μοντέλο απέρριψε σωστά τις κατηγορίες Παραλία (Beach), Βουνό (Mountain) και Ζώο (Animal), καθώς τα στοιχεία αυτά δεν υπήρχαν στην εικόνα. Η δυνατότητα αυτή του μοντέλου όχι μόνο να εντοπίζει τα παρόντα χαρακτηριστικά αλλά και να αποκλείει τα άσχετα είναι κρίσιμη για την ακρίβεια των προβλέψεων.

Επιπλέον, είναι σημαντικό να αποσαφηνιστούν ορισμένες διαφορές που διακρίνουν τις δυνατότητες της σύγχρονης κατηγοριοποίησης σε σχέση με τις απλές τεχνικές. Ο Πίνακας 1.1, που ακολουθεί, συνοψίζει τις κύριες διαφορές μεταξύ της παραδοσιακής κατηγοριοποίησης μίας ετικέτας (single-label classification) και της κατηγοριοποίησης πολλαπλών ετικετών (multi-label classification), εστιάζοντας στα βασικά χαρακτηριστικά τους, όπως ο αριθμός ετικετών ανά δείγμα, η σχέση μεταξύ των ετικετών, η μορφή εξόδου, και τα μέτρα αξιολόγησης [6], [7].

Πίνακας 1.1: Διαφορές μεταξύ Single-Label και Multi-Label Classification

Χαρακτηριστικό / Κατηγοριοποίηση	Binary Classification	Multi-Class Classification	Multi-Label Classification
Αριθμός Ετικετών ανά Δείγμα	Μία ετικέτα από δύο δυνατές κατηγορίες (π.χ., Αληθές/Ψευδές)	Μία ετικέτα από πολλές κατηγορίες	Μία ή περισσότερες ετικέτες από πολλές κατηγορίες
Σχέση Ετικετών	Οι ετικέτες είναι αμοιβαία αποκλειόμενες	Οι ετικέτες είναι αμοιβαία αποκλειόμενες	Οι ετικέτες μπορούν να επικαλύπτονται και να συνυπάρχουν
Μορφή Εξόδου	Μοναδική δυαδική τιμή	Μοναδική κατηγορική τιμή	Σετ ή διάνυσμα δυαδικών τιμών
Παράδειγμα Πρόβλεψης	Email ταξινομημένο ως "Spam" ή "Όχι Spam"	Φρούτο ταξινομημένο ως "Μήλο" ή "Μπανάνα"	Εικόνα με ετικέτες: "Παραλία", "Νερό", "Ηλιοβασίλεμα"
Κύρια Μέτρα Αξιολόγησης	Accuracy, Precision, Recall, F1-Score	Accuracy, Precision, Recall, F1-Score	Hamming Loss, Accuracy, Precision, Recall, F1-Score
Πολυπλοκότητα	Απλή	Μέτρια	Υψηλή (λόγω εξαρτήσεων ετικετών)

Όπως προαναφέρθηκε, η κατηγοριοποίηση πολλαπλών ετικετών έχει πλέον εξελιχθεί πέρα από τις παραδοσιακές εφαρμογές. Με την πρόοδο των τεχνολογιών και την αυξανόμενη πολυπλοκότητα των δεδομένων, αναδύονται νέες εφαρμογές σε τομείς όπως η Υπολογιστική Όραση (Computer Vision), η Επεξεργασία Φυσικής Γλώσσας (Natural Language Processing) και η Εξόρυξη Δεδομένων (Data Mining), οι οποίες αξιοποιούν τις δυνατότητες της σύγχρονης κατηγοριοποίησης προκειμένου να αντιμετωπίσουν τις αυξημένες απαιτήσεις. Παρακάτω παρουσιάζονται ορισμένες από τις πλέον σύγχρονες και καινοτόμες εφαρμογές, οι οποίες αξιοποιούν πρωτοποριακές τεχνολογίες και χρησιμοποιούν μεθόδους κατηγοριοποίησης πολλαπλών ετικετών, όπως αυτές περιγράφονται λεπτομερώς στην πηγή [8]:

Εφαρμογές στην Υπολογιστική Όραση

- **Σχολιασμός Βίντεο:** Η ραγδαία αύξηση των βίντεο σε πλατφόρμες όπως το YouTube και το Instagram καθιστά την κατηγοριοποίησή τους απαραίτητη. Η κατηγοριοποίηση πολλαπλών ετικετών επιτρέπει την ταυτόχρονη κατανομή πολλών κατηγοριών σε ένα βίντεο, π.χ., για σκηνές, αντικείμενα ή γεγονότα. Έτσι, με τον συνδυασμό των ετικετών βελτιώνεται η ακρίβεια αναζήτησης και η κατανόηση του περιεχομένου των βίντεο.
- **Τμηματοποίηση Εγκεφάλου:** Η σωστή κατηγοριοποίηση των λευκών ουσιών του εγκεφάλου, βοηθάει στην ανίχνευση ανωμαλιών και την κατανόηση της ανάπτυξής τους. Οι μέθοδοι πολλαπλών ετικετών αποδίδουν πολλαπλές κατηγορίες-ετικέτες σε ένα σημείο του εγκεφάλου, προσφέροντας ακριβείς ανατομικές πληροφορίες για πιθανή πρόγνωση ψυχικών διαταραχών.

Εφαρμογές στην Επεξεργασία Φυσικής Γλώσσας

- **Κινητές Συσκευές:** Η κατηγοριοποίηση πολλαπλών ετικετών αποτελεί ένα βασικό εργαλείο για τα συστήματα σύστασης στις κινητές συσκευές, επιτρέποντας στις εφαρμογές να προσφέρουν εξατομικευμένο περιεχόμενο στους χρήστες. Για παράδειγμα, βοηθάει στην πρόβλεψη της «αξιολόγησης» ή «προτίμησης» που θα έδινε ο χρήστης σε ένα αντικείμενο. Επίσης, οι βιομετρικές εφαρμογές, όπως η τμηματοποίηση της ίριδας, εκμεταλλεύονται τις δυνατότητες των συνελκτικών νευρωνικών δικτύων για την ανάλυση εικόνων χαμηλής ποιότητας, ενώ οι μέθοδοι ενεργούς μάθησης συμβάλλουν στην ακριβέστερη ανάλυση των σχόλιων στις εφαρμογές, ενισχύοντας έτσι την συνολική εμπειρία του χρήστη.
- **Εξόρυξη Νομικών Κειμένων:** Η κατηγοριοποίηση νομικών εγγράφων, όπως αυτών της βάσης δεδομένων EUR-Lex, επιτρέπει την αντιστοίχιση κειμένων σε πολλαπλές νομικές έννοιες. Η ενημερωμένη έκδοση EURLEX-57K περιλαμβάνει περισσότερα δεδομένα, διευκολύνοντας έτσι την αναζήτηση και ανάλυση νομικών πληροφοριών. Παράλληλα, οι αλγόριθμοι MLC χρησιμοποιούνται για να προβλέπουν δικαστικές αποφάσεις, βασιζόμενοι σε περιγραφές υποθέσεων, σχετικούς νόμους και κατηγορίες.

Εφαρμογές στην Εξόρυξη Δεδομένων

- **Συστήματα Σύστασης:** Τα συστήματα σύστασης λειτουργούν ως μοντέλα πολλαπλών ετικετών (MLC) διότι έχουν τη δυνατότητα να προτείνουν στους χρήστες πολλά αντικείμενα ταυτόχρονα. Για παράδειγμα, σε πλατφόρμες όπως η Amazon, ένα μοντέλο MLC θα μπορούσε να προβλέψει ποια προϊόντα είναι πιθανόν να ενδιαφέρουν τον χρήστη, με βάση το περιεχόμενο του καλαθιού αγορών του.
- **Προφίλ Χρηστών:** Η δημιουργία προφίλ χρηστών είναι απαραίτητη για την παροχή εξατομικευμένων υπηρεσιών σε τομείς όπως τα social media και το ηλεκτρονικό εμπόριο. Έτσι, πολλές εφαρμογές και πλατφόρμες κοινωνικής δικτύωσης, όπως το Facebook, χρησιμοποιούν μοντέλα μηχανικής μάθησης για να αναλύσουν διάφορα δεδομένα των χρηστών και να εξάγουν χαρακτηριστικά, ενδιαφέροντα και προτιμήσεις τους. Αυτές οι πληροφορίες συμβάλλουν στη βελτίωση των προφίλ και επιτρέπουν πιο στοχευμένες προτάσεις και διαφημίσεις, ενισχύοντας έτσι την συνολική εμπειρία του χρήστη.

1.2 Αυτοματοποιημένη Μηχανική Μάθηση (AutoML)

Η Αυτοματοποιημένη Μηχανική Μάθηση (Automated Machine Learning ή AutoML) είναι ένας τομέας που στοχεύει να αυτοματοποιήσει, σε σημαντικό βαθμό, τα διάφορα στάδια ανάπτυξης ενός συστήματος μηχανικής μάθησης. Η AutoML αποτελεί ένα από τα πιο δημοφιλή και επίκαιρα θέματα στον τομέα της μηχανικής μάθησης, προσελκύοντας έντονο ενδιαφέρον από τη βιομηχανία, την ακαδημαϊκή κοινότητα, ακόμη και το ευρύ κοινό [9]. Κύριος στόχος της είναι να διευκολύνει τους μη ειδικούς στη δημιουργία και ανάπτυξη μοντέλων μηχανικής μάθησης, παρέχοντάς τους μια απλή, φιλική και εύχρηστη διεπαφή που απλοποιεί την όλη διαδικασία [10]. Συνεπώς, η AutoML προσφέρει τα εξής τρία σημαντικά πλεονεκτήματα [11]:

1. Μειώνει και απαλλάσσει την κοινότητα από χρονοβόρες και επίμονες διαδικασίες ρύθμισης, εξοικονομώντας χρόνο και πόρους.
2. Ενισχύει τη χρήση και την ανάπτυξη της μηχανικής μάθησης σε οργανισμούς και επιχειρήσεις, διευκολύνοντας την ενσωμάτωσή της σε διάφορους τομείς.
3. Καθιστά τη μηχανική μάθηση προσβάσιμη σε όλους τους επιστήμονες, ενισχύοντας την καινοτομία και τη διεπιστημονική συνεργασία.

Για την χρήση τεχνικών μηχανικής μάθησης με υψηλή απόδοση, απαιτείται συνήθως η ανθρώπινη παρέμβαση σε στάδια όπως η συλλογή δεδομένων, η επιλογή χαρακτηριστικών και η επιλογή κατάλληλων μοντέλων και αλγορίθμων. Το Σχήμα 1.2 απεικονίζει μια τυπική ροή εργασίας των εφαρμογών μηχανικής μάθησης, ενώ ταυτόχρονα δείχνει πώς η AutoML μπορεί να ενσωματωθεί σε αυτή τη διαδικασία, μειώνοντας σημαντικά την ανάγκη για ανθρώπινη εμπλοκή [12]. Η διαδικασία AutoML περιλαμβάνει μια σειρά από διακριτά και αλληλένδετα στάδια, καθένα από τα οποία παίζουν κρίσιμο ρόλο στην επίτευξη του επιθυμητού στόχου. Αναλυτικότερα, τα κύρια στάδια της AutoML περιλαμβάνουν τα εξής:

Συλλογή και Προεπεξεργασία Δεδομένων (Data Collection and Preprocessing): Η συλλογή δεδομένων αποτελεί πρωταρχικό βήμα για τη δημιουργία ή επέκταση ενός συνόλου δεδομένων, ώστε να χρησιμοποιηθεί μετέπειτα στη διαδικασία αυτοματοποιημένης μηχανικής μάθησης [13]. Στη συνέχεια, η AutoML αναλαμβάνει την προεπεξεργασία των δεδομένων, διασφαλίζοντας ότι αυτά είναι κατάλληλα και έτοιμα για χρήση. Η προεπεξεργασία περιλαμβάνει διάφορες τεχνικές, όπως την Συμπλήρωση Ελλιπών Δεδομένων (Data Imputing), την Εξισορρόπηση Δεδομένων (Data Balancing) και την Κωδικοποίηση Δεδομένων (Data Encoding) [14].

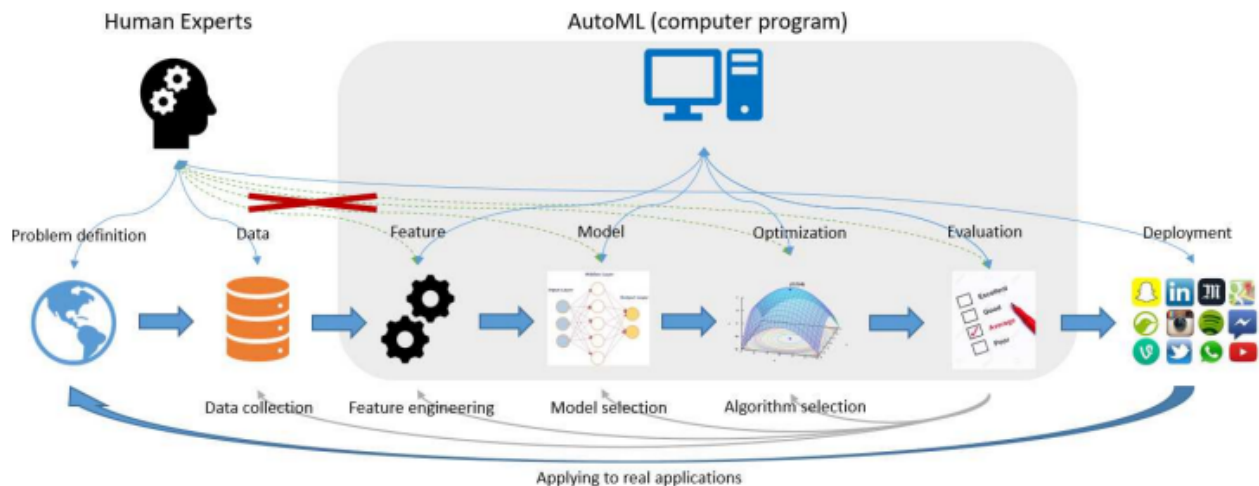
Μηχανική Χαρακτηριστικών (Feature Engineering): Εφόσον έχουν δοθεί τα κατάλληλα δεδομένα από τους χρήστες, ακολουθεί η διαδικασία μηχανικής χαρακτηριστικών. Στόχος της είναι να εξάγει όσο το δυνατόν περισσότερα χρήσιμα χαρακτηριστικά από τα ακατέργαστα δεδομένα, προκειμένου να αξιοποιηθούν από τους αλγορίθμους και τα μοντέλα μηχανικής μάθησης [13]. Εκτός από την βασική λειτουργία, δηλαδή την Επιλογή Χαρακτηριστικών (Feature Selection), επιπρόσθετες δυνατότητες προσφέρουν η Εξόρυξη Χαρακτηριστικών (Feature Mining) καθώς και η Δημιουργία Χαρακτηριστικών (Feature Generation) [14].

Επιλογή Αλγορίθμου (Algorithm Selection): Σε αυτό το στάδιο, η AutoML εξετάζει και συγκρίνει πολλούς αλγορίθμους, επιλέγοντας τον καταλληλότερο με βάση συγκεκριμένα κριτήρια αξιολόγη-

σης. Χρησιμοποιεί μεθόδους, όπως την Διασταυρούμενη Επικύρωση (Cross-Validation Method), και διάφορα μέτρα απόδοσης, όπως την Ακρίβεια (Accuracy), την Ευαισθησία (Precision) και την Ανάκληση (Recall), για να καθορίσει ποιος αλγόριθμος είναι ο πιο αποδοτικός για το συγκεκριμένο πρόβλημα και το εκάστοτε σύνολο δεδομένων [15].

Βελτιστοποίηση Παραμέτρων (Parameters Optimization): Η επιλογή των κατάλληλων παραμέτρων είναι ζωτικής σημασίας για την επίτευξη της μέγιστης απόδοσης και αποτελεσματικότητας του μοντέλου. Η AutoML αυτοματοποιεί αυτή τη διαδικασία, δοκιμάζοντας διάφορες τιμές παραμέτρων και στο τέλος επιλέγει εκείνες που προσφέρουν τα βέλτιστα αποτελέσματα, συνθέτοντας τον ιδανικό συνδυασμό [15].

Δημιουργία και Αξιολόγηση Μοντέλου (Model Creation and Evaluation): Μετά την ολοκλήρωση των προηγούμενων σταδίων, ακολουθεί η δημιουργία του προεκπαιδευμένου μοντέλου με τον καλύτερο αλγόριθμο και τις αντίστοιχες βέλτιστες παραμέτρους. Έπειτα, η AutoML προχωράει στο τελικό στάδιο, που είναι η εκτίμηση και αξιολόγηση της απόδοσης του μοντέλου, προσφέροντας στον χρήστη μια πλήρη ανάλυση των πληροφοριών και χαρακτηριστικών του, ώστε να αποφασίσει εάν επιθυμεί να το χρησιμοποιήσει στο περιβάλλον εργασίας [15].



Σχήμα 1.2: Ροή Εργασίας Εφαρμογών Μηχανικής Μάθησης με AutoML.

Είναι γεγονός πως την τελευταία δεκαετία έχουν σημειωθεί σημαντικές εξελίξεις στον τομέα της AutoML, με την ανάπτυξη προηγμένων Εργαλείων (Tools) και Πλαισίων (Frameworks) που στοχεύουν στην αυτοματοποίηση κρίσιμων σταδίων, όπως η επιλογή των κατάλληλων αλγορίθμων και η βελτιστοποίηση των παραμέτρων. Παρακάτω παρουσιάζονται μερικά από αυτά τα χρήσιμα εργαλεία και πλαίσια, τα οποία διακρίνονται σε τρεις κύριες κατηγορίες [16], [17]:

1. **Κεντριοποιημένα (Centralized):** Είναι χρήσιμα κυρίως για χειρισμό μεσαίων ή μικρών συνόλων δεδομένων (datasets) και όλες οι διαδικασίες εκτελούνται τοπικά.

Auto-Weka: Θεωρείται το πρώτο πλαίσιο αυτοματοποιημένης μηχανικής μάθησης. Αναπτύχθηκε σε Java και βασίζεται στο Weka, μια δημοφιλή βιβλιοθήκη που προσφέρει μεγάλη ποικιλία αλγορίθμων. Το Auto-Weka συνδυάζει την επιλογή αλγορίθμων και τη βελτιστοποίηση

υπερ-παραμέτρων, ενώ παράλληλα χρησιμοποιεί την τεχνική Bayesian Optimization μέσω του SMAC (Sequential Model-based Algorithm Configuration) και του TPE (Tree-Structured Parzen Estimator). Στόχος του είναι να βελτιώσει την απόδοση, ακρίβεια και παραγωγικότητα στην ανάπτυξη μοντέλων [18].

[<https://github.com/automl/autoweka>]

Auto-Sklearn: Βασίζεται στην ευρέως χρησιμοποιούμενη βιβλιοθήκη Python, δηλαδή την scikit-learn, ενώ αποτελεί και ένα εργαλείο ανοιχτού κώδικα όπου η ανάπτυξή του γίνεται ενεργά στο GitHub. Όπως και το Auto-Weka, χρησιμοποιεί το SMAC ως τεχνική Bayesian Optimization. Εισάγει την έννοια της μετα-μάθησης (meta-learning) και προσφέρει μια ξεχωριστή προσέγγιση στη μη επιβλεπόμενη μάθηση, διαχωρίζοντας την προεπεξεργασία σε προεπεξεργασία δεδομένων και προεπεξεργασία χαρακτηριστικών, παρέχοντας μεγαλύτερη ευελιξία σε διάφορες εφαρμογές μηχανικής μάθησης [19].

[<https://github.com/automl/auto-sklearn>]

2. **Κατανεμημένα (Distributed)**: Αξιοποιούν πολλαπλούς κόμβους για επεξεργασία, αντιμετωπίζοντας την υπολογιστική πολυπλοκότητα των προβλημάτων, και μπορούν να υποστηρίξουν μεγαλύτερα σύνολα δεδομένων.

MLBox: Αποτελεί μία από τις κορυφαίες AutoML βιβλιοθήκες Python που προσφέρει δυνατότητες και λειτουργίες για την προετοιμασία δεδομένων, την επιλογή μοντέλων και την βελτιστοποίηση υπερ-παραμέτρων. Ένα από τα βασικά του χαρακτηριστικά είναι η συσσώρευση μοντέλων (model stacking), όπου ένα νέο μοντέλο εκπαιδεύεται με βάση τον συνδυασμό των προβλέψεων πολλών προγενέστερων μοντέλων. Επιπλέον, το MLBox χρησιμοποιεί τη βιβλιοθήκη Hyperopt, η οποία προσφέρει μία κατανεμημένη και ασύγχρονη διαδικασία βελτιστοποίησης των υπερπαραμέτρων [20].

[<https://github.com/AxeldeRomblay/MLBox>]

TransmogriAI: Πρόκειται για ένα απ'τα πιο σύγχρονα και αρθρωτά AutoML εργαλεία. Το TransmogriAI έχει σχεδιαστεί πάνω σε Spark, είναι γραμμένο σε Scala και χρησιμοποιεί ροές εργασίας που περιλαμβάνουν προεπεξεργασία χαρακτηριστικών και επιλογή μοντέλων, με ελάχιστη ανθρώπινη παρέμβαση. Ένα σημαντικό πλεονέκτημά του είναι ότι επαναχρησιμοποιεί αυτές τις ροές εργασίας. Επιπλέον, υποστηρίζει οκτώ διαφορετικούς δυαδικούς ταξινομητές και πέντε αλγόριθμους παλινδρόμησης [21].

[<https://github.com/salesforce/TransmogriAI>]

3. **Βασισμένα στο Cloud (Cloud-Based)**: Προσφέρουν ευελιξία και επεκτασιμότητα μέσω των υποδομών cloud, εξασφαλίζοντας αποδοτική διαχείριση μεγάλων όγκων δεδομένων, ταχύτερη επεξεργασία και μειωμένο κόστος συντήρησης.

Google Cloud AutoML: Χρησιμοποιεί ενισχυτική μάθηση (reinforcement learning) και διαθέτει την ικανότητα να εκπαιδεύει μοντέλα για ποικίλες εφαρμογές με σύνθετα δομημένα δεδομένα, όπως επεξεργασία εικόνας και βίντεο, καθώς και ανάλυση κειμένου. Όλα αυτά επιτυγχάνονται σχετικά εύκολα χωρίς να χρειάζονται ιδιαίτερες τεχνικές γνώσεις από τον χρήστη. Επιπρόσθετα, η Google AutoML προσφέρει προηγμένες επιλογές διαμόρφωσης που μπορούν να βελτιώσουν και να απλοποιήσουν τις υπάρχουσες ροές εργασίας μηχανικής μάθησης [22].

[<https://cloud.google.com/automl>]

Azure AutoML: Είναι μια υπηρεσία που προσφέρει αυτοματοποιημένες λειτουργίες μηχανικής μάθησης τόσο για προβλήματα ταξινόμησης (classification) όσο και παλινδρόμησης (regression). Η Azure AutoML είναι διαθέσιμη μέσω του Python SDK της Microsoft Azure και αξιοποιεί τις δυνατότητες των αλγορίθμων της βιβλιοθήκης scikit-learn. Τέλος, δίνει την δυνατότητα στους χρήστες να την χρησιμοποιήσουν είτε τοπικά είτε απομακρυσμένα, εκμεταλλευόμενοι την ισχύ και την επεκτασιμότητα των υπηρεσιών Azure Cloud [23].

[<https://learn.microsoft.com/en-us/azure/machine-learning/>]

1.3 Κίνητρο

Μέχρι στιγμής, έχουν παρουσιαστεί οι βασικές έννοιες και εισαγωγικές πληροφορίες που αφορούν την κατηγοριοποίηση πολλαπλών ετικετών (multi-label classification) και την αυτοματοποιημένη μηχανική μάθηση (automated machine learning). Είναι κατανοητό ότι ο κεντρικός άξονας αυτής της διπλωματικής εργασίας είναι η κατηγοριοποίηση των δεδομένων πολλαπλών ετικετών, μια διαδικασία που θα εξεταστεί εκτενώς στο επόμενο κεφάλαιο, με την εφαρμογή των Δέντρων Απόφασης (Decision Trees) καθώς και τη χρήση Μεθόδων Μετασχηματισμού Προβλήματος (Problem Transformation Methods).

Τα Δέντρα Απόφασης αποτελούν έναν από τους πλέον διαδεδομένους αλγόριθμους για κατηγοριοποίηση, γνωστά για την ευανάγνωστη και κατανοητή δομή τους. Η διαφάνειά τους τα καθιστά ιδιαίτερα προσιτά και εύκολα στη χρήση, διευκολύνοντας σημαντικά τη διαδικασία λήψης αποφάσεων σε ποικίλα επιστημονικά πεδία που αξιοποιούν τις δυνατότητές τους. Στο πλαίσιο της κατηγοριοποίησης πολλαπλών ετικετών, θα γίνει χρήση των προσεγγίσεων μετασχηματισμού προβλήματος, όπως είναι οι μέθοδοι BinaryRelevance, ClassifierChain και LabelPowerset. Αυτές οι μέθοδοι μετατρέπουν το πρόβλημα πολλαπλών ετικετών σε μικρότερα, διαχειρίσιμα υπό-προβλήματα, καθιστώντας δυνατή την αποτελεσματική ανάλυση και τον καλύτερο έλεγχο των δεδομένων.

Παρά τη δημοφιλία τους, η χρήση αυτών των αλγορίθμων παρουσιάζει αρκετές προκλήσεις. Το βασικό εμπόδιο είναι ότι απαιτείται εξειδικευμένη γνώση και εμπειρία στη Μηχανική Μάθηση και την Εξόρυξη Δεδομένων, καθώς και εξοικείωση στον προγραμματισμό. Ως αποτέλεσμα, πολλοί χρήστες αναζητούν ψηφιακά εργαλεία που απλοποιούν την όλη διαδικασία. Ωστόσο, η αγορά εξακολουθεί να στερείται από εξειδικευμένα λογισμικά που να προσφέρουν άμεση πρόσβαση σε αυτούς τους αλγόριθμους και να διευκολύνουν την διεξαγωγή πειραμάτων, αναλύσεων, καθώς και την εκτέλεση των εργασιών που απαιτούνται.

Παρότι υπάρχουν κάποιες διαθέσιμες λύσεις στον χώρο της Μηχανικής Μάθησης και της Εξόρυξης Δεδομένων, που χρησιμοποιούν δέντρα απόφασης και κατηγοριοποιούν δεδομένα, οι εφαρμογές αυτές συχνά εμφανίζουν μειονεκτήματα που περιορίζουν την ευρεία χρήση τους. Ένα συνηθισμένο πρόβλημα είναι το υψηλό κόστος, καθώς πολλές από αυτές τις εφαρμογές απαιτούν συνδρομή, προσφέροντας μόνο βασικές λειτουργίες στις δωρεάν εκδόσεις τους. Επιπλέον, ορισμένα εργαλεία είναι διαθέσιμα αποκλειστικά ως προγράμματα εγκατάστασης για υπολογιστές, γεγονός που απαιτεί την εγκατάσταση λογισμικού ή βιβλιοθηκών, με αποτέλεσμα να επιβαρύνεται το σύστημα του χρήστη και να περιορίζεται η ευελιξία του. Αξιοσημείωτο είναι επίσης ότι δεν υπάρχει κάποιο

εργαλείο ή εφαρμογή που να συνδυάζει τα Δέντρα Απόφασης με τις μεθόδους μετασχηματισμού προβλήματος για την κατηγοριοποίηση των δεδομένων πολλαπλών ετικετών, το οποίο παράλληλα να παρέχει μια ολοκληρωμένη και φιλική προς τον χρήστη λύση, καλύπτοντας αποτελεσματικά τις ανάγκες του.

1.4 Συνεισφορά

Όλα τα παραπάνω στοχεύει να καλύψει η παρούσα διπλωματική εργασία, μέσα από την ανάπτυξη μίας διαδικτυακής εφαρμογής που ονομάζεται «MultilabelDTree». Η εφαρμογή αυτή αποτελεί ένα χρήσιμο εργαλείο ανοικτού κώδικα και απευθύνεται τόσο σε άτομα με εμπειρία στο εν λόγω αντικείμενο όσο και σε χρήστες με περιορισμένη γνώση στον τομέα.

Πιο συγκεκριμένα, η εφαρμογή περιλαμβάνει και ενσωματώνει τεχνικές αυτοματοποιημένης μηχανικής μάθησης (AutoML), επιτρέποντας στους χρήστες να ανεβάζουν τα δικά τους σύνολα δεδομένων πολλαπλών ετικετών (multi-label) και να δημιουργούν προεκπαιδευμένα μοντέλα μέσω της τεχνικής μετασχηματισμού προβλήματος που επιθυμούν, όπως BinaryRelevance, LabelPowerset ή ClassifierChain, καθώς και των κατάλληλων παραμέτρων για το δέντρο απόφασης. Επιπρόσθετα, η εφαρμογή επιτρέπει την αξιολόγηση της απόδοσης των μοντέλων, η οποία πραγματοποιείται μέσω της τεχνικής επικύρωσης k-fold cross-validation, παρέχοντας λεπτομερείς πληροφορίες για κάθε κατηγορία, καθώς και μια συνολική εικόνα της αποτελεσματικότητας του μοντέλου. Βάσει των αποτελεσμάτων, οι χρήστες έχουν τη δυνατότητα να αποθηκεύσουν το μοντέλο για μελλοντική χρήση σε προβλέψεις αλλά και να οπτικοποιήσουν τα αντίστοιχα δέντρα απόφασης. Το «MultilabelDTree» παρέχει στους χρήστες την επιπλέον δυνατότητα ενεργοποίησης της αυτόματης επιλογής (Auto Mode) της καλύτερης μεθόδου μετασχηματισμού προβλήματος και των καλύτερων τιμών παραμέτρων, ο προσδιορισμός των οποίων γίνεται βάσει τις υψηλότερη ακρίβειας που επιτυγχάνουν.

Τέλος, η εφαρμογή λειτουργεί μέσω ενός Web API, διευκολύνοντας την ενσωμάτωσή της σε άλλες εφαρμογές ή την περαιτέρω επέκτασή της από προγραμματιστές που επιθυμούν να αξιοποιήσουν τις λειτουργίες και δυνατότητές της. Η εφαρμογή διαθέτει μία σύγχρονη και φιλική προς τον χρήστη γραφική διεπαφή, επιτρέποντας σε κάθε ενδιαφερόμενο να αποκτήσει πρόσβαση στις λειτουργίες της χωρίς να έχει ιδιαίτερες γνώσεις. Φιλοξενείται σε server του Τμήματος Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων, ενώ η πλήρης υλοποίηση είναι διαθέσιμη μέσω του GitHub, καλύπτοντας έτσι τις ανάγκες διάφορων χρηστών και οργανισμών.

Συνολικά, το «MultilabelDTree» επιδιώκει να καλύψει ένα κενό στον τομέα της μηχανικής μάθησης και εξόρυξης δεδομένων, αποτελώντας ένα χρήσιμο εργαλείο για ερευνητές, φοιτητές, επαγγελματίες, αλλά και για οποιονδήποτε ενδιαφέρεται για εφαρμογές που σχετίζονται με το εν λόγω επιστημονικό αντικείμενο.

1.5 Οργάνωση της Διπλωματικής

Σε αυτήν την ενότητα, αναλύουμε την οργάνωση των επτά κεφαλαίων της διπλωματικής εργασίας, επιδιώκοντας να προσφέρουμε μια συνοπτική και σαφή εικόνα των βασικών στόχων και κύριων σημείων που καλύπτει το καθένα. Έχοντας ολοκληρώσει το Κεφάλαιο 1, είδαμε τις θεμελιώδεις εισαγωγικές έννοιες που αποτελούν τη βάση της μελέτης μας. Αρχικά, έγινε αναφορά στην Κατηγοριοποίηση Πολλαπλών Ετικετών (Multi-label Classification), όπου εξετάσαμε τον ορισμό της, μερικά παραδείγματα χρήσης, τις διαφορές που εμφανίζει σε σχέση με την κατηγοριοποίηση μίας ετικέτας, καθώς και ορισμένες σύγχρονες εφαρμογές που αναδεικνύουν την καινοτομία της σήμερα. Στη συνέχεια, μελετήσαμε την Αυτοματοποιημένη Μηχανική Μάθηση (AutoML), καλύπτοντας τον ορισμό, τα πλεονεκτήματα, τα κύρια στάδιά της, καθώς και τα πιο σύγχρονα εργαλεία και πλαίσια που χρησιμοποιούνται σε αυτόν τον τομέα.

Στο Κεφάλαιο 2, εστιάζουμε στους κύριους αλγορίθμους και μεθόδους κατηγοριοποίησης πολλαπλών ετικετών, εξετάζοντας τους διαφορετικούς τρόπους με τους οποίους μπορούμε να διαχειριστούμε προβλήματα με περισσότερες από μία κλάσεις ταυτόχρονα. Αρχικά, γίνεται μια ταξινόμηση των μεθόδων multi-label classification, αναλύοντας τις πιο διαδεδομένες τεχνικές. Στη συνέχεια, εμβαθύνουμε στους τρεις σημαντικούς αλγορίθμους μετασχηματισμού προβλήματος: τον Binary Relevance, τον Label Powerset και τον Classifier Chain, μιας και χρησιμοποιούνται πλήρως στην εφαρμογή μας. Κάθε αλγόριθμος περιγράφεται διεξοδικά μέσα από παράδειγμα και απεικόνιση του αντίστοιχου ψευδοκώδικα προκειμένου να γίνουν κατανοητοί. Στο τέλος, παρουσιάζονται τα κύρια μέτρα απόδοσης που χρησιμοποιούνται για την αξιολόγηση της ποιότητας ενός μοντέλου κατηγοριοποίησης πολλαπλών ετικετών, προσφέροντας ένα ολοκληρωμένο πλαίσιο.

Στο Κεφάλαιο 3, παρουσιάζουμε τις τεχνολογίες που χρησιμοποιήθηκαν για την ανάπτυξη της εφαρμογής. Για την υλοποίηση του Frontend και τον σχεδιασμό της διεπαφής χρήστη, επιλέχθηκαν οι τεχνολογίες HTML, CSS, και Bootstrap, οι οποίες συνέβαλλαν στη δημιουργία ενός ευχάριστου και δυναμικού περιβάλλοντος. Επιπλέον, χρησιμοποιήθηκε η γλώσσα JavaScript με την βιβλιοθήκη JQuery, για την δημιουργία της λογικής, καθώς και των διαδραστικών λειτουργιών για την αλληλεπίδραση του χρήστη με την εφαρμογή. Από την άλλη, για την υλοποίηση του Backend, χρησιμοποιήθηκε η γλώσσα PHP, η οποία αλληλεπιδρά με τη MySQL βάση δεδομένων για την αποθήκευση, ανάκτηση και επεξεργασία των δεδομένων. Επίσης, επιλέχθηκε η προσέγγιση REST API για μια ομαλή επικοινωνία μεταξύ του frontend και του backend, διευκολύνοντας την απομακρυσμένη και ασφαλή μετάδοση δεδομένων. Παράλληλα, για την ενσωμάτωση τεχνικών μηχανικής μάθησης για multi-label κατηγοριοποίηση, χρησιμοποιήθηκε η Python με τις βιβλιοθήκες scikit-learn και scikit-multilearn. Τέλος, για την ασφαλή και αξιόπιστη αποστολή email, χρησιμοποιήθηκε η βιβλιοθήκη PHPMailer, μέσω του Composer.

Στο Κεφάλαιο 4, προσφέρουμε μια εκτενή αναφορά σχετικά με την σχεδίαση και υλοποίηση του «MultilabelDTree». Ειδικότερα, αναλύουμε τις λειτουργικές απαιτήσεις της εφαρμογής, την αρχιτεκτονική της μέσα από κατάλληλα διαγράμματα και σχήματα, καθώς και τους τύπους συνόλων δεδομένων που χρησιμοποιούνται από τους χρήστες. Επιπλέον, συζητάμε και κατηγοριοποιούμε τους διαφορετικούς τύπους χρηστών που υποστηρίζει η εφαρμογή μας. Στη συνέχεια εξετάζουμε διεξοδικά την υλοποίηση τόσο του backend όσο και του frontend. Δίνοντας ιδιαίτερη έμφαση στο backend, περιγράφουμε τη βάση δεδομένων που δημιουργήθηκε, το Web API με τα endpoints, κα-

θώς και τον χειρισμό των multi-label και προεκπαιδευμένων μοντέλων που προσφέρουν οι κύριες σελίδες της εφαρμογής.

Στο Κεφάλαιο 5, παρουσιάζουμε την εφαρμογή «MultilabelDTree» και τις δυνατότητές της μέσα από σχήματα και αναλυτικές επεξηγήσεις, με στόχο την πλήρη κατανόηση της από τον χρήστη. Το παρόν κεφάλαιο λειτουργεί ως εγχειρίδιο χρήσης (manual) της γραφικής διεπαφής και των βασικών λειτουργιών της εφαρμογής, περιγράφοντας αναλυτικά την Αρχική Σελίδα της Εφαρμογής, τη Διαδικασία Εγγραφής (Sign Up) και Εισόδου (Log In), την Ανάκτηση Κωδικού και Επεξεργασία Λογαριασμού, τη Δημιουργία Multi-label Μοντέλων, καθώς και τη Χρήση Προεκπαιδευμένων Μοντέλων. Συνεπώς, η εφαρμογή «MultilabelDTree» έχει αναπτυχθεί με σύγχρονες τεχνολογίες, προσφέροντας μια ολοκληρωμένη εμπειρία χρήστη, καθώς αποτελεί μία ολοκληρωμένη full-stack web εφαρμογή με προηγμένες AutoML λειτουργίες.

Στο Κεφάλαιο 7, παρουσιάζονται τα συμπεράσματα που προέκυψαν από την ανάπτυξη της διαδικτυακής εφαρμογής «MultilabelDTree», η οποία παρέχει ένα φιλικό και λειτουργικό περιβάλλον για την κατηγοριοποίηση δεδομένων πολλαπλών ετικετών μέσω Δέντρων Απόφασης. Παρά την αποτελεσματική δημιουργία της, διαπιστώθηκαν περιθώρια βελτίωσης και ανάπτυξης. Μελλοντικές επεκτάσεις θα μπορούσαν να περιλαμβάνουν την ανάπτυξη έκδοσης για κινητές συσκευές, την ενσωμάτωση μεθόδων όπως οι Μέθοδοι Προσαρμογής Αλγορίθμων (Algorithm Adaptation Methods) και οι Μέθοδοι Συνόλου (Ensemble Methods). Οι προσθήκες αυτές θα ενισχύσουν περαιτέρω τη χρηστικότητα και την αποτελεσματικότητα του εργαλείου, καθιστώντας το ακόμα πιο πολύτιμο για ερευνητές και επαγγελματίες.

Κεφάλαιο 2

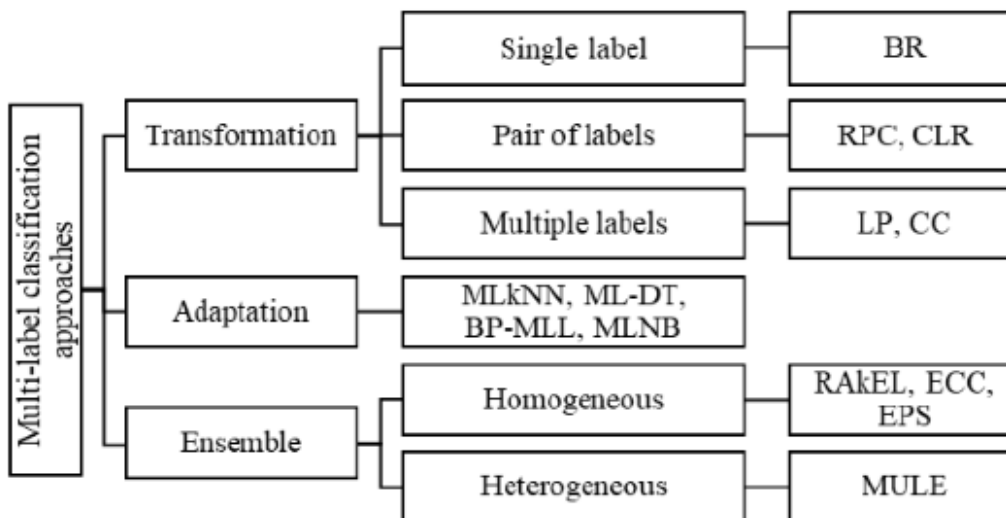
Κατηγοριοποίηση Πολλαπλών Ετικετών

2.1 Μέθοδοι και Αλγόριθμοι

Όπως αναφέρθηκε στην εισαγωγή, η κατηγοριοποίηση πολλαπλών ετικετών αποτελεί έναν εξειδικευμένο τομέα της μηχανικής μάθησης που επιτρέπει μια πιο ευέλικτη και λεπτομερή ταξινόμηση. Σε αντίθεση με την κλασική ταξινόμηση, όπου κάθε στιγμιότυπο ανήκει σε μία μόνο κατηγορία, η κατηγοριοποίηση πολλαπλών ετικετών δίνει τη δυνατότητα σε ένα στιγμιότυπο να ανήκει ταυτόχρονα σε πολλές ετικέτες-κατηγορίες. Για την αντιμετώπιση της πολυπλοκότητας του multi-label classification, έχουν αναπτυχθεί διάφοροι μέθοδοι και αλγόριθμοι. Σύμφωνα με το [4], οι μέθοδοι πολλαπλών ετικετών χωρίζονται στις παρακάτω δύο βασικές ομάδες:

1. **Μέθοδοι Μετασχηματισμού Προβλήματος (Problem Transformation Methods)**
2. **Μέθοδοι Προσαρμογής Αλγορίθμων (Algorithm Adaption Methods)**

Οι μέθοδοι μετασχηματισμού προβλήματος, που εφαρμόζονται πλήρως στην εφαρμογή μας, μετατρέπουν το πρόβλημα πολλαπλών ετικετών σε ένα ή περισσότερα προβλήματα ταξινόμησης με μία ετικέτα ή παλινδρόμησης, διευκολύνοντας τη χρήση παραδοσιακών αλγορίθμων. Από την άλλη, οι μέθοδοι προσαρμογής αλγορίθμων επεκτείνουν συγκεκριμένους αλγορίθμους μάθησης, ώστε να μπορούν να χειρίζονται απευθείας τα δεδομένα πολλαπλών ετικετών [4]. Αξίζει να σημειωθεί πως υπάρχει και μια τρίτη ομάδα, οι Μέθοδοι Συνόλου (Ensemble Methods), οι οποίες συνδυάζουν πολλαπλές μεθόδους με διαφορετικούς τρόπους έτσι ώστε να ενισχύσουν την απόδοση και να διαχειριστούν πιο αποτελεσματικά τις συσχετίσεις μεταξύ των ετικετών. Επιπλέον, οι μέθοδοι πολλαπλών ετικετών μπορούν να χωριστούν και με βάση την εξάρτησή τους από τον αλγόριθμο. Συγκεκριμένα, διακρίνονται σε ομάδες που βασίζονται στον αλγόριθμο (Algorithm Dependent) και σε εκείνες που είναι ανεξάρτητες από αυτόν (Algorithm Independent) [24]. Κάθε μία από αυτές τις στρατηγικές έχει τα δικά της πλεονεκτήματα και περιορισμούς, και η επιλογή της κατάλληλης προσέγγισης εξαρτάται συχνά από τις απαιτήσεις του έργου, όπως ο αριθμός των ετικετών, η αλληλεξάρτησή τους και το μέγεθος του συνόλου δεδομένων. Στο Σχήμα 2.1 απεικονίζεται αναλυτικά ο διαχωρισμός των μεθόδων κατηγοριοποίησης πολλαπλών ετικετών στις βασικές τους ομάδες.



Σχήμα 2.1: Ταξινόμηση Μεθόδων Κατηγοριοποίησης Πολλαπλών Ετικετών.

Όπως διακρίνεται στο παραπάνω σχήμα, οι μέθοδοι μετασχηματισμού προβλήματος με την σειρά τους χωρίζονται ανάλογα με τον αριθμό των ετικετών που λαμβάνεται υπόψιν από τον ταξινομητή. Υπάρχουν μέθοδοι που επεξεργάζονται μόνο μία ετικέτα (Single label) κάθε φορά, όπως ο αλγόριθμος Binary Relevance (BR) [25], ενώ άλλοι, όπως οι RPC [26] και CLR [27], εξετάζουν ταυτόχρονα ζεύγη ετικετών (Pair of labels), δηλαδή δύο ετικέτες μαζί. Στην περίπτωση πολλαπλών ετικετών (Multiple labels), χρησιμοποιούνται όλες οι ετικέτες μιας περίπτωσης ή συγκεκριμένες ομάδες ετικετών, όπως στους αλγόριθμους Label Powerset (LP) [28], RAkEL [29], Classifier Chain (CC) και Ensemble of Classifier Chains (ECC) [30]. Ο LP ενσωματώνει όλες τις ετικέτες μιας περίπτωσης, ενώ οι CC, ECC και RAkEL χρησιμοποιούν ομάδες ετικετών [24].

Οι μέθοδοι προσαρμογής αλγορίθμων, όπως προαναφέρθηκε, προσαρμόζουν αλγορίθμους για να μπορούν να επεξεργάζονται δεδομένα με πολλαπλές ετικέτες απευθείας. Η προσέγγιση αυτή, βασισμένη στην εποπτευόμενη μάθηση, έχει οδηγήσει στη δημιουργία αλγορίθμων που χρησιμοποιούν τα Δέντρα Απόφασης (Decision Trees), τις Μηχανές Υποστηρικτικών Διανυσμάτων (Support Vector Machines), το Naïve Bayes, τα Νευρωνικά Δίκτυα (Neural Networks) καθώς και τους k πλησιέστερους γείτονες (k -nearest neighbors), ώστε να χειρίζονται δεδομένα πολλαπλών ετικετών χωρίς την ανάγκη κάποιας μετατροπής. Αν και οι μέθοδοι αυτοί είναι ευρέως χρησιμοποιούμενοι, συνεχίζεται η έρευνα για τη βελτίωση της αποδοτικότητάς τους στην ανάλυση σύνθετων δεδομένων [24].

Τέλος, οι μέθοδοι συνόλου, που αξιοποιούν πολλαπλούς ταξινομητές για τη βελτίωση της ακρίβειας των προβλέψεων, χρησιμοποιούν αλγορίθμους όπως τους RAkEL, ECC, EPS και MULE. Αυτές οι μέθοδοι διακρίνονται περαιτέρω σε δύο κατηγορίες: τις ομοιογενείς (Homogeneous) και τις ετερογενείς (Heterogeneous). Οι αλγόριθμοι RAkEL, ECC και EPS κατατάσσονται ως ομοιογενείς επειδή χρησιμοποιούν τον ίδιο τύπο βασικού ταξινομητή, ενώ ο MULE κατατάσσεται ως ετερογενής, καθώς συνδυάζει διαφορετικούς τύπους ταξινομητών. Η ετερογένεια αυτή επιτρέπει την αξιοποίηση των πλεονεκτημάτων ποικίλων αλγορίθμων για καλύτερη ανάλυση και πρόβλεψη σε πιο σύνθετα προβλήματα. [24].

Παρακάτω παρουσιάζονται συνοπτικά ορισμένοι από τους πιο δημοφιλείς αλγόριθμους κατηγοριοποίησης πολλαπλών ετικετών, όπως περιγράφονται στο [31]. Αυτοί οι αλγόριθμοι έχουν αναπτυχθεί για την αποτελεσματική διαχείριση περιπτώσεων όπου το κάθε δείγμα δεδομένων μπορεί να ανήκει σε περισσότερες από μία κατηγορίες, αποτελώντας έτσι βασικά εργαλεία για την βελτίωση της ακρίβειας και της αξιοπιστίας σε σύνθετα προβλήματα κατηγοριοποίησης.

Binary Relevance (BR): Ο αλγόριθμος Binary Relevance αποτελεί μια δημοφιλή μέθοδο μετασχηματισμού προβλήματος. Η προσέγγιση αυτή μετατρέπει το πρόβλημα των πολλαπλών ετικετών σε μια σειρά δυαδικών ταξινομήσεων, όπου για κάθε ετικέτα δημιουργείται ένας ξεχωριστός δυαδικός ταξινομητής. Έτσι, για ένα σύνολο δεδομένων με 70 ετικέτες, θα χρειαστούν 70 δυαδικοί ταξινομητές, καθένας εκ των οποίων προβλέπει την παρουσία (1) ή απουσία (0) μιας συγκεκριμένης ετικέτας. Κατά την ταξινόμηση μιας νέας περίπτωσης, ο BR συγκεντρώνει τις προβλέψεις όλων των ταξινομητών για να αποδώσει τις κατάλληλες ετικέτες [31].

Classifier Chain (CC): Ο Classifier Chain είναι μια πιο εξελιγμένη προσέγγιση που διατηρεί την απλότητα του Binary Relevance, προσθέτοντας όμως τη δυνατότητα συνυπολογισμού των συσχετίσεων μεταξύ των ετικετών. Σε αυτή την τεχνική, οι ταξινομητές διατάσσονται σε μια αλυσίδα, με κάθε έναν να επεξεργάζεται τη δική του ετικέτα και να ενσωματώνει τις πληροφορίες-προβλέψεις που παράγονται από τους προηγούμενους ταξινομητές. Αυτό επιτρέπει στον CC να λαμβάνει υπόψη τις συσχετίσεις των ετικετών και να αποφεύγει το πρόβλημα της ανεξαρτησίας τους [31].

Label Powerset (LP): Η μέθοδος Label Powerset μετασχηματίζει ένα πρόβλημα πολλαπλών ετικετών σε ένα απλό πρόβλημα μίας ετικέτας, δημιουργώντας όλα τα πιθανά υποσύνολα ετικετών (unique combinations) του συνόλου και αντιμετωπίζοντας το καθένα ως μία ξεχωριστή νέα ετικέτα. Εκπαιδεύει έναν μόνο ταξινομητή σε όλους τους μοναδικούς συνδυασμούς. Παρόλο που η LP είναι χρήσιμη σε ορισμένα προβλήματα, γίνεται ιδιαίτερα χρονοβόρα και απαιτητική από άποψη υπολογιστικού κόστους και προκαλεί ανισορροπίες, ειδικά όταν υπάρχουν πολλές ετικέτες [31].

Random k-Label Set (RAkEL): Αποτελεί μία βελτιωμένη εκδοχή της μεθόδου LP, η οποία επιλύει το πρόβλημα ανισορροπίας που εμφανίζεται. Η RAkEL δημιουργεί τυχαία σύνολα ετικετών συγκεκριμένου μεγέθους, εφαρμόζει σε αυτά την τεχνική LP, και στο τέλος προβλέπει αθροίζοντας το αποτέλεσμα όλων των εκπαιδευμένων ταξινομητών. Έτσι μειώνεται το υπολογιστικό κόστος και βελτιώνεται την απόδοση του μοντέλου [31].

Multi-Label K-Nearest Neighbor (MLkNN): Ο MLkNN αποτελεί επέκταση του κλασικού αλγορίθμου των k πλησιέστερων γειτόνων (kNN), προσαρμοσμένο για προβλήματα πολλαπλών ετικετών. Συγκεκριμένα, για κάθε νέα αταξινομήτη περίπτωση, ο αλγόριθμος εντοπίζει τους k πλησιέστερους γείτονες από το σύνολο εκπαίδευσης. Έπειτα, υπολογίζει τις προγενέστερες πιθανότητες με βάση αυτές τις k πλησιέστερες περιπτώσεις και στο τέλος προσδιορίζει το σύνολο των ετικετών της υπό εξέταση περίπτωσης, χρησιμοποιώντας τη μέγιστη εκ των υστέρων πιθανότητα [31].

Στις επόμενες τρεις ενότητες του κεφαλαίου (Ενότητες 2.2, 2.3 και 2.4) θα εξετάσουμε λεπτομερώς τις μεθόδους Binary Relevance (BR), Label Powerset (LP) και Classifier Chain (CC), καθώς χρησιμοποιούνται πλήρως στην εφαρμογή «MultilabelDTree» και αποτελούν τους βασικούς αλγόριθμους για την κατηγοριοποίηση των δεδομένων πολλαπλών ετικετών.

2.2 Binary Relevance

Η μέθοδος Binary Relevance (BR) είναι ένας αλγόριθμος που ανήκει στην ομάδα μεθόδων μετασχηματισμού προβλήματος. Κύρια λειτουργία του είναι η διάσπαση του προβλήματος κατηγοριοποίησης πολλαπλών ετικετών (multi-label classification) σε πολλά ανεξάρτητα προβλήματα δυαδικής ταξινόμησης (binary classification), καθένα από τα οποία σχετίζεται αποκλειστικά με μία ετικέτα από το σύνολο $L = \{y_1, y_2, \dots, y_q\}$ [32].

Αρχικά, το multi-label σύνολο εκπαίδευσης διασπάται σε q δυαδικά σύνολα D_{y_j} (όπου $j = 1, \dots, q$), δηλαδή τόσα όσα είναι και ο αριθμός των ετικετών ή στηλών-στόχων, και το κάθε D_{y_j} περιέχει όλα τα στιγμιότυπα από το αρχικό σύνολο δεδομένων με αποκλειστικά μόνο μια θετική ή αρνητική ένδειξη για την συγκεκριμένη ετικέτα y_j . Η ένδειξη αυτή γίνεται θετική αν το σύνολο των πραγματικών ετικετών του στιγμιότυπου περιλαμβάνει την ετικέτα y_j , και αρνητική αν δεν την περιλαμβάνει. Μετά την διάσπαση των δεδομένων, δημιουργείται ένα σύνολο από q δυαδικούς ταξινομητές H_j , οι οποίοι εκπαιδεύονται στα αντίστοιχα δυαδικά σύνολα D_{y_j} , χρησιμοποιώντας έναν αλγόριθμο κατηγοριοποίησης B . Συνοπτικά, η μέθοδος BR κατασκευάζει έναν δυαδικό ταξινομητή ανά ετικέτα, σχηματίζοντας έτσι μία σειρά από q ταξινομητές, όπου ο καθένας εκπαιδεύεται με στόχο την πρόβλεψη της παρουσίας (1) ή της απουσίας (0) μιας συγκεκριμένης ετικέτας. Η προσέγγιση αυτή περιγράφεται στην Εξίσωση (2.1) [32].

$$H_{BR} = \{C_{y_j}((x, y_j)) \rightarrow y'_j \mid y_j \in L, j = 1, \dots, q\} \quad (2.1)$$

Για να γίνει κατανοητή η κεντρική ιδέα της διαδικασίας μετασχηματισμού του BR, παρακάτω παρουσιάζεται το αρχικό multi-label σύνολο δεδομένων (Σχήμα 2.2) με τα τέσσερα δυαδικά σύνολα δεδομένων που δημιουργήθηκαν μετά τον μετασχηματισμό (Σχήμα 2.3), υποθέτοντας ότι υπάρχουν 4 ετικέτες στο πρόβλημα. Οι πιθανές τιμές για την ετικέτα-κλάση Y του κάθε συνόλου είναι είτε «1» (θετική) είτε «0» (αρνητική), και σημειώνονται ως y_i και $\neg y_i$ αντίστοιχα. Προκειμένου να κατηγοριοποιηθεί ένα νέο στιγμιότυπο πολλαπλών ετικετών, η μέθοδος BR εξάγει τις ετικέτες που έχουν προβλεφθεί θετικά από όλους το δυαδικούς ταξινομητές και στο τέλος επιλέγει την καλύτερη ετικέτα για το συγκεκριμένο στιγμιότυπο. Ένα σημαντικό πλεονέκτημα της BR είναι η χαμηλή της υπολογιστική πολυπλοκότητα σε σχέση με άλλες μεθόδους ταξινόμησης πολλαπλών ετικετών [32].

	\mathbf{x}				Y
E_1	x_{11}	x_{12}	\dots	x_{1M}	Y_1
E_2	x_{21}	x_{22}	\dots	x_{2M}	Y_2
\vdots	\vdots	\vdots	\ddots	\vdots	\vdots
E_N	x_{N1}	x_{N2}	\dots	x_{NM}	Y_N

Σχήμα 2.2: Σύνολο Δεδομένων Πολλαπλών Ετικετών.

	Y		Y		Y		Y
E_1	$\neg y_1$	E_1	y_2	E_1	y_3	E_1	$\neg y_4$
E_2	y_1	E_2	$\neg y_2$	E_2	y_3	E_2	y_4
E_3	$\neg y_1$	E_3	$\neg y_2$	E_3	$\neg y_3$	E_3	y_4
E_4	$\neg y_1$	E_4	y_2	E_4	y_3	E_4	$\neg y_4$

Σχήμα 2.3: BR Μετασχηματισμός του Συνόλου Δεδομένων.

Παρακάτω παρατίθεται ο ψευδοκώδικας της μεθόδου κατηγοριοποίησης Binary Relevance, όπως περιγράφεται στις αντίστοιχες βιβλιογραφίες [25], [33], όπου εκπαιδεύεται ένας ξεχωριστός ταξινομητής για κάθε ετικέτα του συνόλου δεδομένων. Στόχος είναι να προβλεφθεί η παρουσία ή η απουσία κάθε ετικέτας για ένα δεδομένο στιγμιότυπο. Στον Αλγόριθμο 1 που ακολουθεί, αποτυπώνεται με σαφήνεια η διαδικασία εκπαίδευσης και πρόβλεψης της μεθόδου BR, με κάθε ετικέτα να αντιμετωπίζεται ανεξάρτητα μέσω ενός δυαδικού ταξινομητή, αναδεικνύονται έτσι τα βήματα που απαιτούνται για την υλοποίηση αυτής της τεχνικής.

Αλγόριθμος 1 Ψευδοκώδικας Binary Relevance

- 1: **Είσοδοι:**
 - 2: D : Σύνολο εκπαίδευσης πολλαπλών ετικετών, με (x_i, y_i) δείγματα.
 - 3: q : Αριθμός στηλών-στόχων.
 - 4: L : Σύνολο ετικετών $\{y_1, y_2, \dots, y_q\}$.
 - 5: E_i : Στιγμιότυπο για πρόβλεψη.
 - 6: **Έξοδοι:**
 - 7: Y^* : Σύνολο προβλεπόμενων ετικετών για το E_i ($Y^* \subseteq Y$).
 - 8: **Διαδικασία:**
 - 9: **Για j από 1 μέχρι q**
 - 10: Παράγω το δυαδικό σύνολο εκπαίδευσης D_{y_j} .
 - 11: Δημιουργώ τον δυαδικό ταξινομητή H_j .
 - 12: **Τέλος επανάληψης**
 - 13: **Επιστρέφω Y^* .**
-

2.3 Label Powerset

Η μέθοδος Label Powerset (LP) μετατρέπει το πρόβλημα κατηγοριοποίησης πολλαπλών ετικετών (multi-label classification) σε ένα πρόβλημα κατηγοριοποίησης πολλαπλών κλάσεων (multi-class classification), αποτελώντας και αυτή μια από τις δημοφιλέστερες τεχνικές μετασχηματισμού προβλήματος, καθώς επιτρέπει τη διαχείριση πολλών ετικετών ταυτόχρονα.

Συγκεκριμένα, η μέθοδος LP δημιουργεί όλες τις πιθανές τιμές ή μοναδικούς συνδυασμούς (unique combinations) που υπάρχουν στο αρχικό σύνολο εκπαίδευσης προκειμένου να χρησιμοποιηθούν στο νέο μετασχηματισμένο multi-class πρόβλημα. Αυτοί οι συνδυασμοί αντιπροσωπεύουν στην

ουσία όλα τα μοναδικά υποσύνολα ετικετών, καθένα από τα οποία μετατρέπεται σε μια ξεχωριστή κλάση-ετικέτα και χρησιμοποιείται ως πρόβλεψη για τα χαρακτηριστικά του συνόλου δεδομένων. Το Σχήμα 2.4 αποσαφηνίζει αυτή την προσέγγιση καθώς παρουσιάζει τον μετασχηματισμό του συνόλου με την μέθοδο LP, παραλείποντας τον χώρο χαρακτηριστικών, αφού η διαδικασία μετασχηματισμού επηρεάζει μόνο τον χώρο των ετικετών [32].

	Y
E_1	$Y_1 = \{y_2, y_3\}$
E_2	$Y_2 = \{y_1, y_3, y_4\}$
E_3	$Y_3 = \{y_4\}$
E_4	$Y_4 = \{y_2, y_3\}$

	Y
E_1	$y_{2,3}$
E_2	$y_{1,3,4}$
E_3	y_4
E_4	$y_{2,3}$

Σχήμα 2.4: LP Μετασχηματισμός του Συνόλου Δεδομένων

Ο αριστερά πίνακας απεικονίζει την αρχική μορφή του συνόλου δεδομένων πολλαπλών ετικετών, ενώ ο δεξιά πίνακας δείχνει το ίδιο σύνολο μετά τον μετασχηματισμό της μεθόδου Label Powerset (LP). Στο αρχικό Σχήμα 2.2, το σύνολο ετικετών L είναι $\{y_1, y_2, y_3, y_4\}$, ενώ μετά τον μετασχηματισμό γίνεται $\{y_{2,3}, y_{1,3,4}, y_4, y_{2,3}\}$. Επιπλέον, η συμβολή $y_{i,j,\dots,k}$ υποδηλώνει ότι το συγκεκριμένο στιγμιότυπο κατηγοριοποιήθηκε με τις ετικέτες $y_i \wedge y_j \wedge \dots \wedge y_k$. Για παράδειγμα, τα δεδομένα του στιγμιότυπου E_1 κατηγοριοποιήθηκαν με το σύνολο ετικετών $y_{2,3}$, υποδεικνύοντας ότι ανήκουν στις ετικέτες 2 και 3. Ωστόσο, αν και η μέθοδος LP λαμβάνει υπόψη τις συσχετίσεις μεταξύ των ετικετών, η εκθετική αύξηση των συνδυασμών μπορεί να δημιουργήσει δυσκολίες όταν ο αριθμός των ετικετών είναι μεγάλος [32]. Ο Αλγόριθμος 2, που ακολουθεί, παρουσιάζει αναλυτικά τα βήματα της διαδικασίας μετασχηματισμού και κατηγοριοποίησης της μεθόδου Label Powerset.

Αλγόριθμος 2 Ψευδοκώδικας Label Powerset

- 1: **Είσοδοι:**
 - 2: D : Σύνολο εκπαίδευσης πολλαπλών ετικετών, με (x_i, y_i) δείγματα.
 - 3: L : Σύνολο ετικετών $\{y_1, y_2, \dots, y_q\}$.
 - 4: E_i : Στιγμιότυπο για πρόβλεψη.
 - 5: **Έξοδοι:**
 - 6: Y^* : Σύνολο προβλεπόμενων ετικετών (unique combinations) για το E_i ($Y^* \subseteq Y$).
 - 7: **Διαδικασία:**
 - 8: **Βήμα 1: Μετασχηματισμός Συνόλου**
 - 9: Δημιουργώ το μοναδικό σύνολο ετικετών (unique combinations) στο D .
 - 10: **Βήμα 2: Εκπαίδευση Ταξινομητή**
 - 11: Χρησιμοποιώ το μετασχηματισμένο σύνολο για να εκπαιδεύσω έναν ταξινομητή H .
 - 12: **Βήμα 3: Πρόβλεψη Ετικετών**
 - 13: **Για κάθε** νέο στιγμιότυπο E_i :
 - 14: Χρησιμοποιώ τον H για να προβλέψω το σύνολο ετικετών για το νέο στιγμιότυπο.
 - 15: **Τέλος επανάληψης**
 - 16: **Επιστρέφω** Y^* .
-

2.4 Classifier Chain

Η μέθοδος Classifier Chain (CC) είναι μια τεχνική που χρησιμοποιείται για την επίλυση προβλημάτων κατηγοριοποίησης πολλαπλών ετικετών (multi-label classification). Αποτελεί την τρίτη μέθοδο μετασχηματισμού προβλήματος (problem transformation method). Η μέθοδος CC χρησιμοποιεί μια αλυσίδα ταξινομητών η οποία μετατρέπει το multi-label πρόβλημα σε μια ακολουθία δυαδικών προβλημάτων. Κάθε ταξινομητής σε αυτή την αλυσίδα εξαρτάται από τις προβλέψεις των προηγούμενων ταξινομητών προκειμένου να ενσωματώσει τις πληροφορίες συσχέτισης μεταξύ των ετικετών. Έτσι, λαμβάνοντας υπόψη αυτές τις συσχετίσεις, βελτιώνεται η ακρίβεια των προβλέψεων, ιδιαίτερα όταν υπάρχει ισχυρή εξάρτηση μεταξύ των ετικετών.

Αναλυτικότερα, στην αλυσίδα ταξινομητών δημιουργούνται συνολικά q δυαδικοί ταξινομητές όπου η σειρά τους καθορίζεται από μια συγκεκριμένη διάταξη ετικετών. Κάθε ταξινομητής κατασκευάζεται έτσι ώστε να εκτιμά την παρουσία (1) ή απουσία (0) μιας συγκεκριμένης ετικέτας-κατηγορίας, βασιζόμενος όχι μόνο στα χαρακτηριστικά των δεδομένων, αλλά και στις προβλέψεις των προηγούμενων ταξινομητών της αλυσίδας. Έστω το σύνολο ετικετών $Y = \{\lambda_1, \lambda_2, \dots, \lambda_q\}$ καθώς και μια συνάρτηση αναδιάταξης $\pi : \{1, 2, \dots, q\} \rightarrow \{1, 2, \dots, q\}$ που ορίζει τη σειρά στην αλυσίδα, δηλαδή $\lambda_{\pi(1)} > \lambda_{\pi(2)} > \dots > \lambda_{\pi(q)}$. Αρχικά, για την j -οστή ετικέτα $\lambda_{\pi(j)}$ στη διατεταγμένη λίστα, η μέθοδος δημιουργεί ένα αντίστοιχο δυαδικό σύνολο εκπαίδευσης $D_{\pi(j)}$ από το αρχικό σύνολο D , όπως φαίνεται και στην Σχέση 2.2 [25]:

$$D_{\pi(j)} = \{(x_i, y_{i\pi(1)}, \dots, y_{i\pi(j-1)}), y_{i\pi(j)} \mid 1 \leq i \leq m\}. \quad (2.2)$$

Πιο συγκεκριμένα, οι προβλέψεις των προηγούμενων ταξινομητών στην αλυσίδα, δηλαδή οι δυαδικές τιμές $y_{i\pi(1)}, \dots, y_{i\pi(j-1)}$, προστίθενται ως επιπλέον χαρακτηριστικά στην περίπτωση x_i . Έπειτα, εκπαιδεύεται ένας δυαδικός ταξινομητής $g_{\pi(j)} : X \times \{-1, +1\}^{j-1} \rightarrow \mathbb{R}$ από το σύνολο $D_{\pi(j)}$ χρησιμοποιώντας έναν αλγόριθμο μάθησης B , δηλαδή $g_{\pi(j)} = B(D_{\pi(j)})$. Με αυτόν τον τρόπο, ο ταξινομητής $g_{\pi(j)}$ καθορίζει τη συνάφεια της ετικέτας $\lambda_{\pi(j)}$, αξιοποιώντας τις σχέσεις με τις προηγούμενες ετικέτες $\lambda_{\pi(1)}, \dots, \lambda_{\pi(j-1)}$ στην αλυσίδα. Για μια νέα περίπτωση πρόβλεψης E_i , το σχετικό σύνολο προβλεπόμενων ετικετών Y^* καθορίζεται ελέγχοντας την έξοδο κάθε ταξινομητή διαδοχικά κατά μήκος της αλυσίδας [25].

Παρακάτω παρατίθεται αναλυτικά ο Αλγόριθμος 3 για τη μέθοδο Classifier Chain. Στη διαδικασία αυτή, κάθε ετικέτα προβλέπεται διαδοχικά σε μια προκαθορισμένη αλυσίδα, με κάθε ταξινομητή να χρησιμοποιεί όχι μόνο τα χαρακτηριστικά των δεδομένων αλλά και τις προβλέψεις των προηγούμενων ταξινομητών.

Αλγόριθμος 3 Ψευδοκώδικας Classifier Chain

- 1: **Είσοδοι:**
- 2: D : Σύνολο εκπαίδευσης πολλαπλών ετικετών, με (x_i, y_i) δείγματα.
- 3: π : Συνάρτηση αναδιάταξης που καθορίζει τη σειρά στην αλυσίδα.
- 4: E_i : Στιγμιότυπο για πρόβλεψη.
- 5: **Έξοδοι:**
- 6: Y^* : Σύνολο προβλεπόμενων ετικετών για το E_i ($Y^* \subseteq Y$).
- 7: **Διαδικασία:**
- 8: **Για** j από 1 μέχρι q
- 9: Δημιουργώ το δυαδικό σύνολο εκπαίδευσης $D_{\pi(j)}$.
- 10: Εκπαιδεύω τον δυαδικό ταξινομητή $g_{\pi(j)}$ χρησιμοποιώντας τον $B(D_{\pi(j)})$.
- 11: **Τέλος επανάληψης**
- 12: Προσδιορίζω τις δυαδικές τιμές (0 ή 1) με βάση τις προβλέψεις.
- 13: **Επιστρέφω** Y^* .

2.5 Μέτρα Απόδοσης

Τα συνήθη κριτήρια αξιολόγησης στις μεθόδους ταξινόμησης μίας ετικέτας βασίζονται σε δύο πιθανές εκβάσεις για κάθε δείγμα: σωστή ή λανθασμένη ταξινόμηση. Αντίθετα, στις ταξινομήσεις πολλαπλών ετικετών, ένα δείγμα μπορεί να ταξινομηθεί μερικώς σωστά, καθώς μπορεί να ανήκει σε πολλές κατηγορίες. Για αυτόν τον λόγο, οι μέθοδοι κατηγοριοποίησης multi-label απαιτούν διαφορετικά κριτήρια από την κλασική single-label ταξινόμηση. Κάποια από αυτά τα μέτρα αποτελούν προσαρμογές των παραδοσιακών μέτρων αξιολόγησης που χρησιμοποιούνται στα προβλήματα μίας ετικέτας, ενώ άλλα έχουν σχεδιαστεί ειδικά για την αξιολόγηση πολλαπλών ετικετών. Παρακάτω παρουσιάζονται τα βασικά μέτρα που εφαρμόστηκαν στην εφαρμογή μας, τα οποία αξιολογούν το μοντέλο κατηγοριοποίησης πολλαπλών ετικετών που διαμορφώνει ο χρήστης, με βάση τις παραμέτρους που έχει θέσει καθώς και την επιλογή μίας εκ των τριών μεθόδων μετασχηματισμού προβλήματος. Τα κύρια κριτήρια αξιολόγησης περιλαμβάνουν τα Hamming Loss, Accuracy, Precision, Recall και F1-Measure. Ιδιαίτερη έμφαση, επίσης, παρουσιάζει και το μέτρο Subset Accuracy (ή Classification Accuracy) το οποίο αξίζει να συζητηθεί [34].

Το μέτρο Hamming Loss υπολογίζει πόσο "λάθος" είναι οι προβλέψεις σε σχέση με το πραγματικό σύνολο ετικετών, μετρώντας πόσες ετικέτες δεν αντιστοιχούν σωστά μεταξύ δύο συνόλων για κάθε δείγμα. Στον τύπο (2.3), το σύμβολο Δ αντιπροσωπεύει τη συμμετρική διαφορά των δύο συνόλων Y_i, Z_i , δηλαδή το σύνολο των στοιχείων που βρίσκονται είτε στο ένα σύνολο είτε στο άλλο, αλλά όχι και στα δύο ταυτόχρονα. Η συμμετρική διαφορά είναι ουσιαστικά το ισοδύναμο της αποκλειστικής διάζευξης (XOR) στη λογική Boolean. Μια χαμηλή τιμή του Hamming Loss δείχνει μεγαλύτερη ακρίβεια καθώς σημαίνει ότι το μοντέλο προβλέπει σωστά τις περισσότερες ετικέτες [34].

$$HammingLoss = \frac{1}{m} \sum_{i=1}^m \frac{|Y_i \Delta Z_i|}{M} \quad (2.3)$$

Η Ακρίβεια Ταξινόμησης (Classification Accuracy) ή Ακρίβεια Υποσυνόλου (Subset Accuracy) (2.4), είναι ένα αυστηρό μέτρο αξιολόγησης της απόδοσης σε προβλήματα πολλαπλών ετικετών. Στην περίπτωση αυτή, απαιτείται το σύνολο των προβλεπόμενων ετικετών να είναι ακριβώς ίδιο με το πραγματικό σύνολο ετικετών για κάθε δείγμα, δηλαδή να υπάρχει πλήρης αντιστοιχία μεταξύ των δύο. Ουσιαστικά, αν το σύνολο των προβλεπόμενων ετικετών συμπίπτει απόλυτα με τις πραγματικές ετικέτες, τότε το αποτέλεσμα θεωρείται επιτυχές. Ο τύπος Classification Accuracy χρησιμοποιεί τη συνάρτηση ένδειξης I , όπου $I(\text{true}) = 1$ και $I(\text{false}) = 0$. Έτσι, όταν το σύνολο των προβλεπόμενων ετικετών Z_i είναι ακριβώς ίδιο με το πραγματικό σύνολο Y_i , η ένδειξη παίρνει την τιμή 1, διαφορετικά παίρνει τιμή 0. Στη συνέχεια, ο συνολικός βαθμός ακρίβειας υπολογίζεται ως ο μέσος όρος των αποτελεσμάτων σε όλα τα δείγματα. Το μέτρο αυτό ορίζεται ως εξής [34]:

$$\text{Classification Accuracy} = \frac{1}{m} \sum_{i=1}^m I(Z_i = Y_i) \quad (2.4)$$

Η Ακρίβεια (Accuracy) (2.5) εκφράζει το ποσοστό των σωστά προβλεπόμενων ετικετών σε σχέση με το συνολικό πλήθος των ετικετών (πραγματικών και προβλεπόμενων) για κάθε δείγμα. Η Ορθότητα (Precision) (2.6) δείχνει το ποσοστό των σωστά προβλεπόμενων ετικετών σε σχέση με το σύνολο των προβλεπόμενων ετικετών, υπολογιζόμενο κατά μέσο όρο για όλα τα δείγματα. Η Ανάκληση (Recall) (2.7) είναι το ποσοστό των σωστά προβλεπόμενων ετικετών σε σχέση με το σύνολο των πραγματικών ετικετών, επίσης μετρούμενο κατά μέσο όρο για όλα τα δείγματα. Τέλος, το F1-Measure (2.8) υπολογίζει τον αρμονικό μέσο μεταξύ της Ορθότητας (Precision) και της Ανάκλησης (Recall), προσφέροντας έναν ενιαίο δείκτη που συνδυάζει και τα δύο μέτρα [35], [36].

$$\text{Accuracy} = \frac{1}{m} \sum_{i=1}^m \frac{|Y_i \cap Z_i|}{|Y_i \cup Z_i|} \quad (2.5)$$

$$\text{Precision} = \frac{1}{m} \sum_{i=1}^m \frac{|Y_i \cap Z_i|}{|Z_i|} \quad (2.6)$$

$$\text{Recall} = \frac{1}{m} \sum_{i=1}^m \frac{|Y_i \cap Z_i|}{|Y_i|} \quad (2.7)$$

$$F1 = \frac{1}{m} \sum_{i=1}^m \frac{2|Y_i \cap Z_i|}{|Z_i| + |Y_i|} \quad (2.8)$$

Κεφάλαιο 3

Τεχνολογίες

3.1 Backend

3.1.1 REST API

Ο όρος «Διεπαφή Προγραμματισμού Εφαρμογών» (Application Programming Interface), γνωστός και ως API, είναι μία εφαρμογή λογισμικού που επιτρέπει την επικοινωνία και την ανταλλαγή πληροφοριών μεταξύ δύο εφαρμογών. Ένα API, δηλαδή, λειτουργεί ως μια γέφυρα επικοινωνίας μεταξύ διαφορετικών προγραμμάτων μέσω κώδικα. Τα APIs είναι χρήσιμα καθώς αποτελούν έναν εύκολο τρόπο πρόσβασης και κοινοποίησης δεδομένων τόσο εντός ενός οργανισμού όσο και μεταξύ διαφορετικών οργανισμών, συμβάλλοντας έτσι στην ανάπτυξη ποικίλων διαδικτυακών λύσεων και εφαρμογών σε διάφορους τομείς [37], [38].

Αξίζει να σημειωθεί, πως υπάρχουν διάφοροι τύποι APIs που εξυπηρετούν συγκεκριμένες ανάγκες και χρήσεις, παρέχοντας διαφορετικά επίπεδα λειτουργικότητας και προσβασιμότητας. Οι κύριοι τύποι APIs είναι τέσσερις: Open APIs, Partner APIs, Internal APIs και Composite APIs [37]. Συνοπτικά, τα Open APIs αποτελούν APIs ανοιχτού κώδικα στα οποία έχουν πρόσβαση τόσο οι προγραμματιστές όσο και οι τελικοί χρήστες χωρίς κανένα περιορισμό. Από την άλλη, τα Partner APIs δεν είναι προσβάσιμα από το κοινό και απαιτούνται συγκεκριμένες άδειες χρήσης για να αποκτήσει κάποιος πρόσβαση. Τα Internal APIs (ή Private APIs) είναι περιορισμένα στην εσωτερική χρήση μιας επιχείρησης, δεν διατίθενται σε εξωτερικούς χρήστες και χρησιμοποιούνται για την υποστήριξη εσωτερικών λειτουργιών και διαδικασιών. Τέλος, τα Composite APIs συνδυάζουν δύο ή περισσότερα APIs και έχουν τη δυνατότητα πρόσβασης σε πολλά συστήματα ταυτόχρονα, συγκεντρώνοντας δεδομένα από πολλαπλές πηγές με μία μόνο κλήση API.

Ένα σημαντικό είδος API είναι το REST API (ή RESTful API). Τα REST APIs παρουσιάζουν πολλά πλεονεκτήματα. Για παράδειγμα, μπορούν και αξιοποιούν τα υπάρχοντα πρωτόκολλα, κυρίως το HTTP, χωρίς να απαιτείται κάποιο επιπλέον λογισμικό ή βιβλιοθήκη για τη λειτουργία τους. Τα APIs καλούνται RESTful APIs εφόσον ακολουθούν τους παρακάτω έξι θεμελιώδεις κανόνες, που σχετίζονται με την λειτουργικότητα και την απόδοσή τους [39]:

1. **Client-Server:** Σε μία REST API εφαρμογή ο πελάτης (client) και ο διακομιστής (server) είναι ξεχωριστοί, επιτρέποντας την ανεξάρτητη λειτουργία τους χωρίς να επηρεάζουν ο ένας τον άλλο [39].
2. **Stateless:** Στα REST APIs, οι κλήσεις είναι ανεξάρτητες μεταξύ τους και κάθε αίτημα περιλαμβάνει όλα τα απαραίτητα δεδομένα χωρίς να χρειάζεται προηγούμενη αποθηκευμένη πληροφορία [39].
3. **Cache:** Για να μειωθεί ο φόρτος των αιτημάτων και να επιτυγχάνονται αποτελεσματικά, το REST API είναι σχεδιασμένο με τρόπο που να προάγει την αποθήκευση (cache) των δεδομένων [39].
4. **Uniform Interface:** Η ύπαρξη μιας ενιαίας διεπαφής διευκολύνει την αποσύνδεση πελάτη-διακομιστή, επιτρέποντας ομαλά την ανάπτυξη των υπηρεσιών και την καλύτερη συντήρηση του API [39].
5. **Layered System:** Η αρχιτεκτονική των REST APIs είναι πολυεπίπεδη, βοηθώντας στη δημιουργία πιο ευέλικτων και κλιμακούμενων εφαρμογών μέσω της χρήσης ιεραρχιών [39].
6. **Code on Demand:** Επιτρέπει τη μεταφορά κώδικα (ή applets) μέσω του API, τον οποίο μπορεί να χρησιμοποιήσει η εφαρμογή για επιπλέον λειτουργικότητες [39].

Αυτοί οι περιορισμοί διασφαλίζουν την αξιοπιστία και την αποτελεσματικότητα των RESTful APIs, καθιστώντας τα ιδανικά για εφαρμογές που απαιτούν ευελιξία και δυνατότητα κλιμάκωσης. Συνεπώς, η εφαρμογή «MultilabelDTree» ακολουθεί μια REST API προσέγγιση.

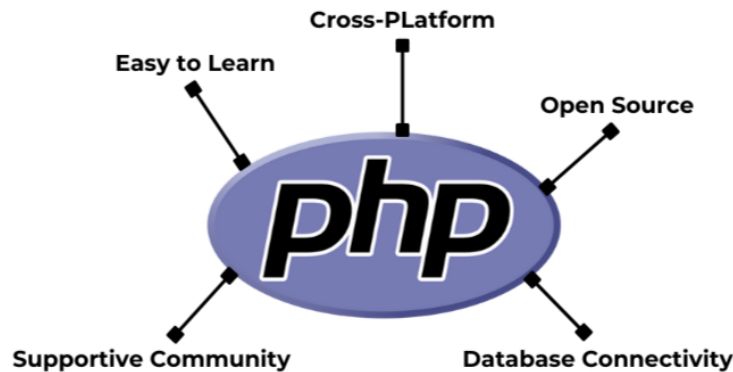
Στο REST API χρησιμοποιούνται 5 βασικές HTTP μέθοδοι που επιτρέπουν την επικοινωνία μεταξύ πελάτη (client) - διακομιστή (server) για τη διαχείριση των πόρων. Συγκεκριμένα, οι μέθοδοι αυτές είναι οι: GET, POST, PUT, PATCH και DELETE. Η μέθοδος GET χρησιμοποιείται για την ανάκτηση των δεδομένων, ενώ η POST επιτρέπει τη δημιουργία νέων δεδομένων, όπως εγγραφές σε μια βάση. Οι μέθοδοι PUT και PATCH χρησιμοποιούνται για την ενημέρωση υπάρχοντων δεδομένων, όπου το PUT ενημερώνει ολόκληρο τον πόρο ενώ το PATCH συγκεκριμένα μέρη του. Τέλος, το DELETE διαγράφει δεδομένα από τον διακομιστή. Αυτές οι μέθοδοι είναι τα θεμέλια του REST API, επιτρέποντας τη διαχείριση δεδομένων με δομημένο και ευέλικτο τρόπο.

3.1.2 PHP

Η PHP είναι μια γλώσσα προγραμματισμού και χρησιμοποιείται κυρίως για την ανάπτυξη διαδικτυακών εφαρμογών. Εκτελείται τόσο σε προγράμματα περιήγησης όσο και στη γραμμή εντολών, δίνοντας τη δυνατότητα στους προγραμματιστές να επιλέγουν πού θα εμφανίζονται τα αποτελέσματα του κώδικά τους. Αρχικά, ο όρος PHP σήμαινε "Personal Home Page" (Προσωπική Ιστοσελίδα), αλλά πλέον ονομάζεται "Hypertext Preprocessor" (Προεπεξεργαστής Υπερκειμένου). Σήμερα η PHP βρίσκεται στην έκδοση 8, αν και η έκδοση 7 παραμένει πιο σταθερή και διαδεδομένη, ενώ η πρώτη έκδοσή της κυκλοφόρησε πριν από 26 χρόνια [40].

Η PHP έχει γίνει πολύ δημοφιλής για τη δημιουργία ιστοσελίδων και εφαρμογών διακομιστή ως μία backend γλώσσα προγραμματισμού, λόγω των πλεονεκτημάτων που εμφανίζει. Αρχικά, η php

έχει το πλεονέκτημα της δια-λειτουργικότητας, καθώς είναι συμβατή με κάθε πλατφόρμα (Cross-Platform), λειτουργώντας εξίσου καλά σε Mac, Windows και Linux, χωρίς να απαιτείται κάποιο συγκεκριμένο λειτουργικό σύστημα. Επιπλέον, η PHP είναι ανοιχτού κώδικα (Open Source), δίνοντας την δυνατότητα στους προγραμματιστές να εξελίσσουν και να επεκτείνουν τον κώδικά της. Είναι εύκολη στην εκμάθηση, ειδικά για αρχάριους προγραμματιστές, και προσφέρει μια ομαλή καμπύλη μάθησης για όσους έχουν βασικές γνώσεις προγραμματισμού. Επίσης, η PHP συνδέεται εύκολα με κάθε είδους βάση δεδομένων, είτε σχεσιακή (όπως MySQL και PostgreSQL) είτε μη σχεσιακή (όπως MongoDB), ενώ σημαντικό είναι και το γεγονός ότι διαθέτει μία μεγάλη και ενεργή διαδικτυακή κοινότητα που παρέχει πλούσια τεκμηρίωση και βοηθητικούς οδηγούς για τη χρήση των εργαλείων της, διευκολύνοντας έτσι την επίλυση τυχόν προβλημάτων [40].



Σχήμα 3.1: Τα Πλεονεκτήματα της PHP.

Η σύνταξη της PHP περιλαμβάνει τις ετικέτες '`<?php`' και '`?>`', οι οποίες σηματοδοτούν την αρχή και το τέλος του κώδικα PHP, επιτρέποντας την ενσωμάτωση του σε ένα αρχείο HTML. Παρόλα αυτά, ο PHP κώδικας αποθηκεύεται συνήθως σε ένα ξεχωριστό αρχείο με την κατάληξη '`.php`'. Επιπρόσθετα, τα ονόματα των μεταβλητών ξεκινάνε με το σύμβολο '\$' και ο τύπος τους δεν καθορίζεται από τον προγραμματιστή, αλλά προσδιορίζεται αυτόματα κατά την εκτέλεση. Τέλος, η PHP υποστηρίζει τόσο διαδικαστική όσο και αντικειμενοστρεφή συγγραφή κώδικα, δίνοντας στους προγραμματιστές ευελιξία στον τρόπο οργάνωσης των προγραμμάτων τους [41], [42].

3.1.3 MySQL

Η MySQL είναι το πιο δημοφιλές και ανοιχτού κώδικα Σύστημα Διαχείρισης Σχεσιακών Βάσεων Δεδομένων (RDBMS) που χρησιμοποιείται για την αποθήκευση και οργάνωση δεδομένων. Χάρη στην αξιοπιστία, την υψηλή απόδοση, την επεκτασιμότητα και την ευκολία χρήσης της, η MySQL αποτελεί εύχρηστη επιλογή για προγραμματιστές. Μεγάλοι οργανισμοί, όπως το Facebook, το Netflix και το Booking, βασίζονται στη MySQL για να υποστηρίξουν τις υψηλές απαιτήσεις τους σε κίνηση δεδομένων. Παρότι η MySQL εμφανίστηκε σχεδόν πριν από 30 χρόνια, εξακολουθεί να αποτελεί ένα ισχυρό εργαλείο και κατατάσσεται στη 2η θέση ως η πιο δημοφιλής βάση δεδομένων παγκοσμίως, μετά την Oracle Database. Έτσι, η MySQL είναι αρκετά ευέλικτη ώστε να καλύπτει εφαρμογές όλων των μεγεθών, από προσωπικά έργα μέχρι μεγάλες επιχειρήσεις [43], [44].

Η MySQL διαθέτει ορισμένα χαρακτηριστικά που την καθιστούν ιδανική και κατάλληλη για τη διαχείριση των βάσεων δεδομένων. Πρώτον, η εγκατάσταση της MySQL είναι γρήγορη και απλή, ενώ η διαχείριση των βάσεων μέσω του RDBMS είναι σχετικά εύκολη για τους προγραμματιστές. Η MySQL είναι ένα από τα πιο αξιόπιστα συστήματα διαχείρισης βάσεων δεδομένων, εξασφαλίζοντας την απρόσκοπτη λειτουργία κρίσιμων εφαρμογών. Επιπλέον, η MySQL μπορεί να επεκταθεί προκειμένου να υποστηρίξει εφαρμογές με εκατομμύρια χρήστες, διότι περιλαμβάνει χαρακτηριστικά που την καθιστούν ιδανική για μεγάλες πλατφόρμες. Μέσω της MySQL, οι προγραμματιστές έχουν την ελευθερία να αναπτύξουν εφαρμογές με διαφορετικά μοντέλα δεδομένων, χρησιμοποιώντας εξίσου αποδοτικά SQL και NoSQL δεδομένα [43], [44].



Σχήμα 3.2: Τα Χαρακτηριστικά της MySQL.

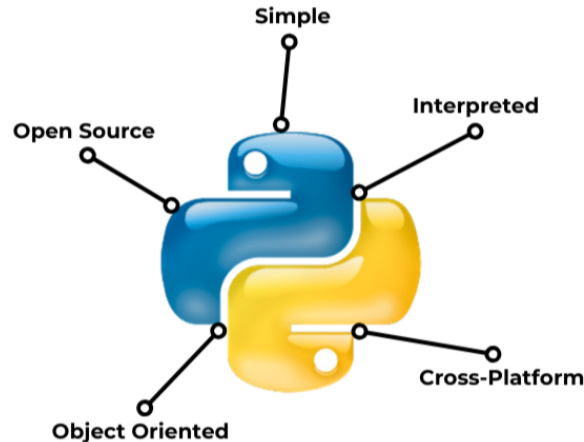
Τέλος, η MySQL χρησιμοποιεί τη γλώσσα SQL για να διαχειρίζεται και να αποθηκεύει τα δεδομένα σε μορφή πινάκων. Οι βασικές εντολές της SQL περιλαμβάνουν το 'SELECT' για την αναζήτηση των δεδομένων, το 'INSERT' για την εισαγωγή νέων εγγραφών, το 'UPDATE' για την ενημέρωση των δεδομένων και το 'DELETE' για την διαγραφή δεδομένων. Αυτές οι εντολές επιτρέπουν εύκολο και οργανωμένο χειρισμό των πληροφοριών μέσα στις βάσεις δεδομένων.

3.1.4 Python

Η Python είναι μια ευέλικτη, υψηλού επιπέδου γλώσσα προγραμματισμού, γνωστή για την απλότητα και τη δυναμική της φύση. Δημιουργήθηκε αρχικά ως χόμπι από τον σχεδιαστή της, τον Guido van Rossum, ο οποίος ήθελε να αναπτύξει μια γλώσσα εύκολη στην ανάγνωση και απλή στην χρήση. Για παράδειγμα, η Python, αντί για αγκύλες, χρησιμοποιεί εσοχές για να περιγράψει τα μπλοκ κώδικα, κάτι που την κάνει πιο όμορφα δομημένη και κατανοητή. Παρά την αρχική της έλλειψη δημοτικότητας, ιδιαίτερα στον τομέα της μηχανικής μάθησης και της τεχνητής νοημοσύνης, η Python αναδείχθηκε ως μία από τις πιο παραγωγικές και εύχρηστες γλώσσες προγραμματισμού, διευκολύνοντας σημαντικά την εργασία των προγραμματιστών [45].

Η Python ξεχωρίζει από τις άλλες γλώσσες προγραμματισμού λόγω των βασικών της χαρακτηριστικών. Αρχικά, η Python είναι εξαιρετικά απλή και εύκολη στην κατανόηση, ακόμα και για αρχάριους. Χάρη στη φιλική προς τον προγραμματιστή προσέγγισή της, ξεχωρίζει από άλλες γλώσσες προγραμματισμού. Επιπρόσθετα είναι μία ερμηνευόμενη γλώσσα (interpreted), δηλαδή ο κώδικας εκτελείται γραμμή προς γραμμή, το οποίο καθιστά την διαδικασία αποσφαλμάτωσης και κατανόησης πιο απλή και γρήγορη. Η Python φημίζεται για την δια-λειτουργικότητα που προσφέρει, καθώς μπορεί να τρέξει σε διάφορες πλατφόρμες, όπως Windows, Linux και Unix, καθιστώντας την μια

φορητή γλώσσα. Επίσης, είναι μία δωρεάν και ανοιχτού κώδικα γλώσσα, προσφέροντας ελευθερία χρήσης και ανάπτυξης εφαρμογών χωρίς την ανάγκη αδειοδότησης. Εξαιτίας της έτσι φτιαγμένης μορφής της, η Python μπορεί να χρησιμοποιηθεί σε δομικό και λειτουργικό προγραμματισμό, ενώ μπορεί να υποστηρίξει εξίσου καλά και αντικειμενοστραφή προγραμματισμό, διευκολύνοντας την επίλυση σύνθετων προβλημάτων μέσω κώδικα που βασίζεται σε αντικείμενα. Τέλος, είναι μια επεκτάσιμη γλώσσα, διότι επιτρέπει εύκολα την ενσωμάτωση άλλων γλωσσών στον κώδικά της, όπως C και C++, διευρύνοντας με αυτόν τον τρόπο τις δυνατότητες των εφαρμογών [45]. Παρακάτω, απεικονίζονται τα βασικά χαρακτηριστικά της Python μέσω ενός σχήματος (Σχήμα 3.3).



Σχήμα 3.3: Τα Βασικά Χαρακτηριστικά της Python.

3.1.5 Composer και PHPMailer

Το Composer είναι ένα εργαλείο για τη διαχείριση εξαρτήσεων (Dependency Management) που υποστηρίζεται μαζί με την γλώσσα προγραμματισμού PHP. Μέσω του Composer, οι προγραμματιστές έχουν την δυνατότητα να καθορίζουν ποιες βιβλιοθήκες χρειάζεται το έργο τους, και αυτό έπειτα αναλαμβάνει να τις εγκαθιστά ή να τις ενημερώνει για αυτούς. Οι βιβλιοθήκες που διαχειρίζεται εγκαθίστανται σε έναν κατάλογο μέσα στο έργο (συνήθως τον κατάλογο vendor). Σε αντίθεση με άλλους διαχειριστές πακέτων, το Composer λειτουργεί σε επίπεδο έργου και όχι σε παγκόσμιο επίπεδο. Παρόλα αυτά, μέσω της λειτουργίας "global" δίνεται η δυνατότητα σε όποιον επιθυμεί να εγκαταστήσει πακέτα για ένα έργο παγκόσμιου επιπέδου [46].

Το PHPMailer είναι μια δημοφιλής βιβλιοθήκη ανοιχτού κώδικα για την αποστολή email μέσω PHP, καθώς είναι συμβατό με πολλά Frameworks, όπως το Laravel και το Symfony, και χρησιμοποιείται σε μεγάλα έργα όπως το WordPress. Υποστηρίζει πολλούς τρόπους αποστολής email, όπως: Την χρήση του διακομιστή SMTP, την συνάρτηση PHP mail(), καθώς και τις τεχνολογίες Sendmail και gmail. Μερικά βασικά χαρακτηριστικά του PHPMailer είναι η επαλήθευση SMTP, που διασφαλίζει την ασφαλή σύνδεση στους διακομιστές email, ο αυτόματος έλεγχος διευθύνσεων, που επιβεβαιώνει ότι οι διευθύνσεις email είναι έγκυρες, και η προστασία από διάφορα είδη κακόβουλων επιθέσεων. Γενικότερα, το PHPMailer είναι ένα αξιόπιστο εργαλείο για οποιονδήποτε επιθυμεί να στείλει email μέσω PHP, παρέχοντας ασφάλεια και ευελιξία [47].

Παρακάτω, παρουσιάζεται ένα παράδειγμα κώδικα από το αρχείο `composer.json`, όπου δηλώνεται η εξάρτηση από τη βιβλιοθήκη `PHPMailer`, η οποία χρησιμοποιείται στην εφαρμογή «`MultilabelDTree`» για την αποστολή email. Η συγκεκριμένη δήλωση περιλαμβάνει την έκδοση της βιβλιοθήκης που εγκαταστάθηκε μέσω του `Composer`.

```
{
  "require": {
    "phpmailer/phpmailer": "^6.9"
  }
}
```

3.1.6 Scikit-learn και Scikit-Multilearn

Η `scikit-learn` είναι μια από τις πιο δημοφιλής βιβλιοθήκες ανοιχτού κώδικα για μηχανική μάθηση, γραμμένη σε `Python`. Προσφέρει έναν απλό και αποδοτικό τρόπο ενσωμάτωσης χρήσιμων αλγορίθμων και βιβλιοθηκών μηχανικής μάθησης στον κώδικα. Η βιβλιοθήκη αυτή περιλαμβάνει μια μεγάλη γκάμα εργαλείων, όπως διάφορες μεθόδους ταξινόμησης (`classification`), παλινδρόμησης (`regression`), συσταδοποίησης (`clustering`), μείωσης διαστάσεων (`dimensionality reduction`), επιλογής μοντέλου (`model selection`) και προεπεξεργασίας δεδομένων (`data pre-processing`). Η `scikit-learn` υποστηρίζεται και είναι συμβατή από διάφορα λειτουργικά συστήματα, είναι πολύ εύκολη στην εγκατάσταση, και συνεχώς εξελίσσεται στον τομέα της επιστήμης υπολογιστών. Χρησιμοποιείται ευρέως σε επιχειρήσεις, ερευνητικά έργα και επιστημονικές δημοσιεύσεις. Οποιοσδήποτε ενδιαφερόμενος μπορεί να χρησιμοποιήσει και να ανακαλύψει τις δυνατότητες της από την επίσημη ιστοσελίδα: <https://scikit-learn.org> [48], [49].

Η `scikit-multilearn` είναι μια σύγχρονη βιβλιοθήκη ανοιχτού κώδικα, βασισμένη εξίσου στην `Python` και σχεδιασμένη κυρίως για την εκτέλεση ταξινομήσεων πολλαπλών ετικετών. Είναι ενσωματωμένη πλήρως πάνω στο οικοσύστημα της γνωστής βιβλιοθήκης `scikit-learn`, προσφέροντας στους χρήστες έναν αποδοτικό τρόπο χειρισμού προβλημάτων πολλαπλών ετικετών μέσα από χρήσιμες τεχνικές και λειτουργίες. Το κύριο χαρακτηριστικό της είναι ότι υποστηρίζει προσαρμοσμένες υλοποιήσεις αλγορίθμων για `multi-label` ταξινόμηση, προσεγγίσεις διαχωρισμού χώρου ετικετών βασισμένες σε δίκτυα, καθώς και ενσωματωμένους ταξινομητές που αποτελούν χρήσιμα εργαλεία για τους προγραμματιστές. Η βιβλιοθήκη υποστηρίζει τις τρεις βασικές μεθόδους κατηγοριοποίησης: τις μεθόδους μετασχηματισμού προβλήματος (`problem transformation methods`), που μετατρέπουν τα προβλήματα πολλαπλών ετικετών σε προβλήματα μιας ετικέτας, τις μεθόδους προσαρμογής αλγορίθμων (`algorithm adaptation methods`), που προσαρμόζουν αλγορίθμους για τη διαχείριση πολλαπλών ετικετών, και τις μεθόδους συνόλου (`ensemble methods`), που συνδυάζουν πολλούς ταξινομητές για καλύτερα αποτελέσματα [50]. Στην παρούσα εφαρμογή, δόθηκε έμφαση στις μεθόδους μετασχηματισμού προβλήματος, οι οποίες παρουσιάστηκαν αναλυτικά στο Κεφάλαιο 2 και αξιοποιήθηκαν πλήρως με στόχο την κατηγοριοποίηση των δεδομένων πολλαπλών ετικετών. Γενικότερα, η `scikit-multilearn` αποτελεί ένα ευέλικτο και αποδοτικό εργαλείο για ερευνητές και προγραμματιστές, εξασφαλίζοντας αποτελεσματικές λύσεις σε προβλήματα πολλαπλών ετικετών. Μπορείτε να ανακαλύψετε περισσότερα στην επίσημη σελίδα: <http://scikit.ml/index.html>.

3.2 Frontend

3.2.1 HTML

Η HTML (Hypertext Markup Language ή Γλώσσα Σήμανσης Υπερκειμένου) είναι το θεμελιώδες δομικό στοιχείο που χρησιμοποιείται για τη δημιουργία περιεχομένου στο διαδίκτυο. Ορίζει, δηλαδή, τη δομή και το νόημα των στοιχείων μιας ιστοσελίδας. Άλλες τεχνολογίες, όπως η CSS (για την εμφάνιση) και η JavaScript (για τη λειτουργικότητα), τις οποίες θα εξετάσουμε στις επόμενες υπό-ενότητες, συνδυάζονται με την HTML για να δώσουν πλήρη μορφή στις ιστοσελίδες. Πιο συγκεκριμένα, ο όρος "υπερκείμενο" αναφέρεται στους συνδέσμους που ενώνουν τις ιστοσελίδες (είτε εντός ενός ιστότοπου είτε ανάμεσα σε διαφορετικούς ιστότοπους). Πολλοί χρήστες δημιουργούν ή ανεβάζουν περιεχόμενο και το συνδέουν με πολλές άλλες ιστοσελίδες, συμβάλλοντας ενεργά στη δημιουργία αυτού που αποκαλούμε σήμερα "Παγκόσμιος Ιστός" [51].

Κύριο χαρακτηριστικό της HTML είναι οι "ετικέτες", τις οποίες χρησιμοποιεί για να περιγράψει κείμενα, εικόνες ή άλλα στοιχεία που εμφανίζονται σε έναν φυλλομετρητή (browser). Αυτές αποτελούνται από το όνομα του στοιχείου που περιβάλλεται από τα σύμβολα '<' και '>'. Το όνομα του στοιχείου μέσα στην ετικέτα δεν είναι ευαίσθητο σε πεζά ή κεφαλαία γράμματα (case-insensitive). Για παράδειγμα, η ετικέτα <title> μπορεί να γραφτεί είτε ως <Title> είτε ως <TITLE>, χωρίς να αλλάζει η λειτουργία της. Παρ' όλα αυτά, η καθιερωμένη και προτεινόμενη πρακτική είναι να γράφετε τις ετικέτες με πεζά γράμματα για συνέπεια και καλύτερη αναγνωσιμότητα [51].

Η πιο πρόσφατη και κύρια έκδοση της HTML είναι η HTML5, που κυκλοφόρησε το 2008. Η τρέχουσα έκδοση ονομάζεται και ως HTML Living Standard, καθώς εξελίσσεται συνεχώς. Οι μελλοντικές βελτιώσεις της περιλαμβάνουν ενδεικτικά: Καλύτερη προσαρμογή εικόνων σε διαφορετικά μεγέθη οθόνης, προηγμένο έλεγχο για την αναπαραγωγή βίντεο, δυνατότητες για πιο ευέλικτες και επεκτάσιμες γλώσσες και ενσωμάτωση καμερών κινητών συσκευών για καλύτερη υποστήριξη φωτογραφιών και βίντεο. Η HTML συνεχίζει να είναι η καρδιά του Ιστού, συνδυάζοντας απλότητα και ισχυρή λειτουργικότητα [52].

Ακολουθεί ένα παράδειγμα που δείχνει πώς δομείται ο κώδικας της HTML, προσφέροντας μια σαφή εικόνα για τη σύνταξη και τη λειτουργία της γλώσσας.

```
<!DOCTYPE html>
<html>
  <head>
    <title>My web page</title>
  </head>
  <body>
    <h1>Hello, world!</h1>
    <p>This is my first web page.</p>
    <p>It contains a
      <strong>main heading</strong> and <em> paragraph </em>.
    </p>
  </body>
</html>
```

3.2.2 CSS

Μια εξίσου σημαντική τεχνολογία είναι και η CSS (Cascading Style Sheets), η οποία είναι μια γλώσσα που χρησιμοποιείται για να ορίσει την εμφάνιση και τη διάταξη ενός εγγράφου γραμμένου σε HTML ή XML (όπως SVG, MathML ή XHTML). Δηλαδή, μέσω της CSS καθορίζεται ο τρόπος που θα εμφανιστούν τα στοιχεία στην οθόνη μίας ιστοσελίδας. Επιπλέον, η CSS είναι συμβατή με όλα τα σύγχρονα προγράμματα περιήγησης, σύμφωνα με τα πρότυπα του W3C. Παρόλο που στο παρελθόν υπήρχαν διάφορες CSS εκδόσεις, όπως CSS1, CSS2.1 ή CSS3, πλέον δεν χρησιμοποιούνται αριθμημένες εκδόσεις και αναφερόμαστε απλά ως "CSS" με τα επιμέρους "modules" της να έχουν τον αντίστοιχο αριθμό έκδοσης [53].

Αξίζει να σημειωθεί, πως υπάρχουν τρεις κύριοι τύποι CSS. Αρχικά, το Ενσωματωμένο CSS (Inline CSS) χρησιμοποιείται για να εισάγει χαρακτηριστικά στυλ άμεσα στα στοιχεία που βρίσκονται μέσα στις HTML ετικέτες, μέσω της ιδιότητας `style`. Παρά το γεγονός ότι είναι χρήσιμα για γρήγορες αλλαγές σε συγκεκριμένες περιπτώσεις, η διαχείριση των inline css σε μεγαλύτερες ιστοσελίδες μπορεί να είναι δύσκολη. Ένας άλλος τύπος CSS είναι το Εσωτερικό CSS (Internal CSS), το οποίο εφαρμόζεται εντός ενός εγγράφου HTML (αρχείο με κατάληξη `.html`), χρησιμοποιώντας μια ετικέτα `<style>` εντός της ετικέτας `<head>`. Αυτή η μέθοδος είναι αποτελεσματική όταν θέλουμε να δώσουμε χαρακτηριστικά σε μια σελίδα τη φορά, όμως μπορεί να γίνει πολύ χρονοβόρα αν πολλές σελίδες χρειάζονται διαφορετικά χαρακτηριστικά. Από την άλλη, τα Εξωτερικά CSS (External CSS) δημιουργούνται από ένα εξωτερικό φύλλο στυλ (αρχείο με κατάληξη `.css`) και μπορούν να ενσωματωθούν σε πολλές ιστοσελίδες ταυτόχρονα μέσω της ετικέτας `<link>`. Αυτή η προσέγγιση είναι η πιο αποτελεσματική, καθώς διατηρεί το CSS ξεχωριστό και συμβάλλει θετικά στην συντήρηση ολόκληρης της ιστοσελίδας. Ακολουθούν μερικά παραδείγματα των τριών τύπων CSS [54].

Παράδειγμα Inline CSS:

```
<p style="color: orange; font-size: 25px;">
  This is my second paragraph.
</p>
```

Παράδειγμα Internal CSS:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8" />
  <style>
    h1 {
      color: blue;
    }
  </style>
</head>
<body>
  <h1>Good Morning!</h1>
</body>
</html>
```

Παράδειγμα External CSS:

```
<head>
  <link rel="stylesheet" href="nameOfTheSheet.css">
</head>
```

Τέλος, οι Selectors (Επιλογείς) CSS μπορούν και καθορίζουν ποια στοιχεία θα μορφοποιηθούν. Αυτοί χωρίζονται στις εξής τέσσερις κατηγορίες: Element Selector, ID Selector, Class Selector και Universal Selector. Πρώτον, ο Επιλογέας Στοιχείου (Element Selector) στοχεύει συγκεκριμένες ετικέτες HTML για να εφαρμόσει στυλ, όπως για παράδειγμα στην ετικέτα <p> μίας παραγράφου, και χρησιμοποιεί το όνομα της συγκεκριμένης ετικέτας ως αναγνωριστικό. Δεύτερον, ο Επιλογέας Ταυτότητας (ID Selector) στοχεύει σε συγκεκριμένα στοιχεία του εγγράφου που φέρουν ένα μοναδικό ID, το οποίο προηγείται πάντα από το σύμβολο (#). Τρίτον, ο Επιλογέας Κλάσης (Class Selector) επιλέγει στοιχεία με ένα συγκεκριμένο χαρακτηριστικό κλάσης, χρησιμοποιώντας μια τελεία (.) ακολουθούμενη από το όνομα της κλάσης. Τέλος, ο Καθολικός Επιλογέας (Universal Selector) αντιπροσωπεύεται από έναν αστερίσκο (*) και χρησιμοποιείται για να εφαρμόσει στυλ σε όλα τα στοιχεία ενός εγγράφου. Αυτοί οι επιλεγείς βοηθούν τους προγραμματιστές να χτίσουν και να δημιουργήσουν μία όμορφη Γραφική Διεπαφή (User Interface ή UI) μίας εφαρμογής ή ιστοσελίδας. Παρακάτω παρουσιάζονται ενδεικτικά παραδείγματα που δείχνουν τη λειτουργία των διαφορετικών CSS Selectors [54].

Παράδειγμα Element Selector:

```
p {
  background-color: red;
}
```

Παράδειγμα ID Selector:

```
#id {
  background-color: blue;
}
```

Παράδειγμα Class Selector:

```
.intro {
  background-color: green;
}
```

Παράδειγμα Universal Selector:

```
* {
  background-color: black;
}
```

3.2.3 Bootstrap

Το Bootstrap αποτελεί ένα ισχυρό frontend framework που επιτρέπει τους προγραμματιστές και σχεδιαστές να δημιουργούν γρήγορα ομοιόμορφες και responsive ιστοσελίδες. Πρόκειται στην ουσία για μια εκτεταμένη συλλογή από χρήσιμα και επαναχρησιμοποιήσιμα κομμάτια κώδικα, τα οποία είναι γραμμένα σε HTML, CSS και JavaScript. Το Bootstrap έχει ως κύριο στόχο να μειώνει την εκτεταμένη συγγραφή CSS κώδικα διευκολύνοντας σημαντικά τον προγραμματιστή και γλιτώνοντας του χρόνο κατά τον σχεδιασμό ιστοσελίδων [55], [56].

Το Bootstrap έχει αρκετά πλεονεκτήματα για τον frontend προγραμματισμό. Μερικά από αυτά είναι η πλούσια βιβλιοθήκη συστατικών, με την οποία προσφέρει μια μεγάλη γκάμα από προσαρμόσιμα στοιχεία που μπορούν εύκολα να ενσωματωθούν σε μια ιστοσελίδα. Μεταξύ αυτών περιλαμβάνονται: γραμμές πλοήγησης, αναδυόμενα μενού, μπάρες προόδου, μικρογραφίες εικόνων, και πολλά

άλλα. Επιπλέον, διαθέτει ένα προκαθορισμένο σύστημα πλέγματος που επιτρέπει του χρήστες του να οργανώνουν το περιεχόμενο άμεσα. Παράλληλα, περιλαμβάνει ενσωματωμένα εργαλεία που προσαρμόζουν τις εικόνες αυτόματα στο μέγεθος της συσκευής ή της οθόνης. Τέλος, το Bootstrap φημίζεται για την προσαρμοστικότητα και αποδοτικότητα του, καθώς ένα από τα βασικά σημεία του, όπως και για τα άλλα frameworks πάνω στο οποίο εξετάζεται, είναι η επίδραση στο χρόνο φόρτωσης της εφαρμογής. Παρόλο που ενδέχεται να επιβαρύνει την αρχική φόρτωση, η ευελιξία και οι δυνατότητες προσαρμογής του αντισταθμίζουν την οποιαδήποτε καθυστέρηση. Συνεπώς, με το Bootstrap οι χρήστες μπορούν να αναπτύξουν ισχυρές και επαγγελματικές ιστοσελίδες με ελάχιστη προσπάθεια, αξιοποιώντας τις πλούσιες λειτουργίες του [55], [56].

3.2.4 JavaScript

Η JavaScript, γνωστή και ως JS, είναι μια γλώσσα προγραμματισμού που λειτουργεί στην πλευρά του πελάτη (client-side) και είναι σχεδιασμένη για τη δημιουργία εφαρμογών, παιχνιδιών και ιστοσελίδων. Αποτελεί μία από τις βασικές τεχνολογίες του διαδικτύου, μαζί με την HTML και CSS που είδαμε παραπάνω. Η πλειονότητα των προγραμματιστών ιστού χρησιμοποιεί κυρίως JavaScript, ως την κύρια γλώσσα για να προγραμματίσει τις εφαρμογές, την οποία και αξιοποιεί έτσι ώστε να "ζωντανέψει" μια ιστοσελίδα, μετατρέποντάς την από στατική σε δυναμική. Η JS μπορεί να προσφέρει δυνατότητα κίνησης, προσθήκη animations, προβολή εξατομικευμένου περιεχομένου και ενημερώσεις σε πραγματικό χρόνο [57].

Επιπλέον, η γλώσσα αυτή ενισχύει την εμπειρία χρήστη μέσω διαφόρων χαρακτηριστικών όπως: κινούμενα γραφικά και εικόνες, αυτόματη αναπαραγωγή βίντεο, εμφάνιση αναδυόμενων παραθύρων και απόκριση σε συγκεκριμένες ενέργειες του χρήστη. Ένα σημαντικό όφελος της JavaScript για την ανάπτυξη ιστοσελίδων είναι η ευκολία εκμάθησης. Η JavaScript είναι σχετικά απλή και κατανοητή, γεγονός που την καθιστά ιδανική για αρχάριους προγραμματιστές. Επιπλέον, μπορεί και εκτελείται απευθείας στον φυλλομετρητή, παρέχοντας ταχύτητα στην ανάπτυξη και εκτέλεση. Ως μία από τις πιο διαδεδομένες γλώσσες στον ιστό, η JavaScript λαμβάνει συνεχή ενημέρωση και βελτιώσεις, ενώ παράλληλα υπάρχουν πολυάριθμα frameworks και βιβλιοθήκες που καθιστούν την ανάπτυξη της πιο αποτελεσματική και ευέλικτη, όπως για παράδειγμα η React, η Angular και η Vue. Είναι αυτονόητο ότι η JavaScript αποτελεί ένα αναπόσπαστο μέρος των προγραμματιστών και είναι απαραίτητη για τη δημιουργία σύγχρονων, δυναμικών και λειτουργικών ιστοσελίδων [57].

3.2.5 JQuery

Η JQuery είναι μια ελαφριά, γρήγορη και πλούσια σε δυνατότητες βιβλιοθήκη JavaScript. Απλοποιεί σημαντικά τις διαδικασίες προγραμματισμού, όπως την πλοήγηση και επεξεργασία εγγράφων HTML, την διαχείριση γεγονότων, τις κινούμενες εικόνες και τις κλήσεις AJAX. Όλα αυτά τα επιτυγχάνει μέσω μιας εύχρηστης διεπαφής προγραμματισμού εφαρμογών (API) που είναι συμβατή με όλους τους μεγάλους περιηγητές [58].

Επιπρόσθετα, η JQuery ξεχωρίζει για τα πολλά και ποικίλα χαρακτηριστικά της, όπως η ελαφριά δομή, η ταχύτητα, η ευελιξία και η χρησιμότητα. Είναι δια-λειτουργική και ανεξάρτητη από την πλατφόρμα, γεγονός που την καθιστά ιδανική για διαφορετικά περιβάλλοντα ανάπτυξης. Υιοθετεί τη φιλοσοφία "γράψε λιγότερα, κάνε περισσότερα", διευκολύνοντας την εκτέλεση σύνθετων

διαδικασιών, όπως των AJAX κλήσεων και τη διαχείριση του DOM, με απλές και αποδοτικές μεθόδους. Η χρήση της συχνά εγείρει ερωτήματα σχετικά με την ανάγκη επιλογής της αντί της απλής JavaScript, όμως η ευχρηστία και η αποτελεσματικότητά της την καθιστούν μια δημοφιλή και αξιόπιστη επιλογή για την ανάπτυξη σύγχρονων διαδικτυακών εφαρμογών. Επομένως, η JQuery αποτελεί ένα ισχυρό εργαλείο για την ανάπτυξη διαδικτυακών εφαρμογών, επαναπροσδιορίζοντας τον τρόπο με τον οποίο εκατομμύρια προγραμματιστές γράφουν JavaScript [59].

Κεφάλαιο 4

Σχεδίαση και Υλοποίηση του MultilabelDTree

4.1 Λειτουργικές Απαιτήσεις

Οι λειτουργικές απαιτήσεις περιγράφουν λεπτομερώς τις δυνατότητες ενός λογισμικού ή εφαρμογής και τον τρόπο που αυτές ανταποκρίνονται στις ανάγκες των χρηστών. Για την εφαρμογή «MultilabelDTree», οι απαιτήσεις αυτές περιλαμβάνουν:

Εγγραφή Χρηστών: Οι χρήστες θα έχουν τη δυνατότητα να δημιουργούν τον δικό τους λογαριασμό μέσω ειδικής φόρμας εγγραφής (Sign Up). Η σελίδα αυτή θα περιλαμβάνει πεδία με βασικές πληροφορίες όπως email, ονοματεπώνυμο χρήστη και κωδικό πρόσβασης. Για την ολοκλήρωση δημιουργίας του λογαριασμού τους, οι χρήστες θα πρέπει να επιβεβαιώνουν την εγγραφή τους, μέσω σχετικού ηλεκτρονικού μηνύματος επιβεβαίωσης λογαριασμού που θα τους αποσταλεί.

Είσοδος Χρηστών: Οι εγγεγραμμένοι χρήστες θα μπορούν να συνδέονται στην εφαρμογή εισάγοντας τα διαπιστευτήριά τους (email και κωδικό πρόσβασης) μέσω της αντίστοιχης σελίδας με την φόρμα σύνδεσης (Log In), προκειμένου να έχουν πλήρη πρόσβαση στις δυνατότητες και λειτουργίες της εφαρμογής.

Επαναφορά Κωδικού Πρόσβασης: Η εφαρμογή θα προσφέρει λειτουργία επαναφοράς κωδικού πρόσβασης για τους χρήστες που τον έχουν ξεχάσει. Οι χρήστες θα εισάγουν τη διεύθυνση email τους σε μια ειδική φόρμα, και η εφαρμογή θα αποστέλλει ένα σύνδεσμο επαναφοράς. Μέσω αυτού του συνδέσμου, οι χρήστες θα μπορούν να ορίσουν ένα νέο κωδικό πρόσβασης, διασφαλίζοντας την ασφάλεια και την ακεραιότητα του λογαριασμού τους.

Επεξεργασία Στοιχείων Χρήστη: Οι χρήστες θα έχουν τη δυνατότητα να επεξεργάζονται τα προσωπικά τους στοιχεία μέσα από τις ειδικές σελίδες που βρίσκονται στο μενού χρήστη. Συγκεκριμένα, Θα μπορούν να ενημερώνουν πληροφορίες όπως το όνομα, το επώνυμο, τη διεύθυνση email και, εφόσον χρειάζεται, τον κωδικό πρόσβασης. Όλες οι αλλαγές θα αποθηκεύονται με ασφάλεια, ενώ το σύστημα θα παρέχει επιβεβαίωση για την επιτυχή τους ολοκλήρωση.

Διαγραφή Λογαριασμού: Επίσης, η εφαρμογή θα επιτρέπει στους χρήστες να διαγράψουν τον λογαριασμό τους εάν το επιθυμούν, μέσω της σχετικής σελίδας που θα βρουν στο μενού. Πριν την οριστική διαγραφή και εφόσον έχουν εισάγει σωστά τον κωδικό, θα παρέχεται ειδοποίηση για τις συνέπειες, όπως η απώλεια δεδομένων. Η διαδικασία θα ολοκληρώνεται με την επιβεβαίωση οριστικής διαγραφής, εξασφαλίζοντας ότι η ενέργεια πραγματοποιείται μόνο από τον ίδιο τον χρήστη.

Σελίδες Πληροφόρησης Εφαρμογής: Η εφαρμογή θα περιλαμβάνει και ειδικές σελίδες πληροφόρησης, όπως το «Home», το «About» και το «API Docs», τόσο για τους εγγεγραμμένους και μη εγγεγραμμένους χρήστες όσο και για τους προγραμματιστές αντίστοιχα. Οι σελίδες αυτές θα παρέχουν οδηγίες, βήματα και λεπτομέρειες σχετικά με τη λειτουργία και τη χρήση της εφαρμογής. Επιπρόσθετα, θα βοηθούν τους χρήστες να κατανοούν καλύτερα τα εργαλεία και τις δυνατότητες του «MultilabelDTree», εξασφαλίζοντας μια πιο ομαλή και εύκολη εμπειρία χρήσης.

Επεξεργασία Αρχείων - Συνόλων Δεδομένων: Οι χρήστες θα έχουν τη δυνατότητα να επεξεργάζονται τα σύνολα δεδομένων τους με διάφορους τρόπους, όπως το ανέβασμα αρχείων (upload new dataset) με την επιλογή να τα ορίζουν ως δημόσια (public) ή ιδιωτικά (private). Επίσης, τα δεδομένα θα προβάλλονται σε μορφή πίνακα για προεπισκόπηση (dataset preview), θα κατεβαίνουν (download dataset) για τοπική αποθήκευση και θα διαγράφονται (delete dataset) από το σύστημα. Τα σύνολα δεδομένων μπορούν να αξιοποιηθούν, από τους χρήστες, είτε για την εκπαίδευση και δημιουργία ενός νέου μοντέλου είτε για την κατηγοριοποίηση των δεδομένων-στιγμιότυπων τους με βάση ένα ήδη προεκπαιδευμένο μοντέλο. Οι δυνατότητες αυτές θα επιτυγχάνονται στις σελίδες «Build a Model» και «Pretrained Models» αντίστοιχα.

Δημιουργία Multi-label Μοντέλων Κατηγοριοποίησης: Μέσω της πρώτης κύριας σελίδας «Build a Model», οι χρήστες θα μπορούν να δημιουργούν multi-label μοντέλα κατηγοριοποίησης με δυνατότητα επιλογής διάφορων παραμέτρων όπως τα features, τα labels και τον transformation classifier. Θα παρέχεται επίσης η δυνατότητα επιλογής παραμέτρων για τον DTree base classifier και τη μέθοδο cross-validation, όπως το μέγιστο βάθος (max depth), τον ελάχιστο αριθμό δειγμάτων ανά φύλλο (min samples leaf) και τον αριθμό k. Επιπρόσθετα, οι χρήστες θα μπορούν να ενεργοποιούν την λειτουργία Auto Mode, η οποία θα βρίσκει αυτόματα τις βέλτιστες τιμές του transformation classifier, του max depth και του min samples leaf για το μοντέλο, εξασφαλίζοντας την υψηλότερη δυνατή ακρίβεια που επιτυγχάνει.

Αξιολόγηση της Απόδοσης των Μοντέλων: Η αξιολόγηση της απόδοσης των μοντέλων επιτρέπει στους χρήστες να αναλύουν την αποτελεσματικότητα των multi-label μοντέλων κατηγοριοποίησης που έχουν δημιουργήσει. Η εφαρμογή θα παρέχει μετρικές αξιολόγησης, συμπεριλαμβάνοντας τα hamming loss, accuracy, precision, recall και f1-score για την κάθε ετικέτα αλλά και συνολικά για το μοντέλο. Οι μετρικές αυτές θα παρουσιάζονται σε μορφή πινάκων, ώστε να διευκολύνουν τους χρήστες να κατανοήσουν την απόδοση του μοντέλου τους.

Αποθήκευση Μοντέλων: Θα πρέπει να υπάρχει λειτουργία αποθήκευσης μοντέλων, που θα δίνει στους χρήστες τη δυνατότητα να διατηρούν τα multi-label μοντέλα κατηγοριοποίησης που έχουν δημιουργήσει για μελλοντική χρήση. Τα αποθηκευμένα μοντέλα θα μπορούν να αξιοποιηθούν ανά πάσα στιγμή για αναλυτική επισκόπηση ή για την κατηγοριοποίηση νέων multi-label δεδομένων, ειδικά σε μη ταξινομημένα σύνολα (unclassified datasets).

Επεξεργασία Προεκπαιδευμένων Μοντέλων: Η δεύτερη κύρια σελίδα, «Pretrained Models», θα δίνει στους χρήστες τη δυνατότητα να χειρίζονται τα μοντέλα που έχουν ήδη δημιουργήσει. Μέσα από αυτή τη σελίδα, οι χρήστες θα μπορούν αρχικά να βλέπουν πληροφορίες για τις παραμέτρους ενός συγκεκριμένου επιλεγμένου μοντέλου. Επιπλέον, θα μπορούν να κατεβάζουν τοπικά τα μοντέλα τους, σε .pkl μορφή, για περαιτέρω χρήση εκτός της εφαρμογής ή να διαγράψουν όσα δεν τους χρειάζονται, διατηρώντας τη λίστα τους οργανωμένη και προσαρμοσμένη στις τρέχουσες ανάγκες.

Οπτικοποίηση Δέντρων Απόφασης: Η εφαρμογή θα προσφέρει στους χρήστες την επιπλέον δυνατότητα να οπτικοποιούν τα Δέντρα Απόφασης (Decision Trees) των μοντέλων τους, παρέχοντας μια πιο κατανοητή αναπαράσταση της δομής τους. Μέσω αυτής της λειτουργίας, οι χρήστες θα μπορούν να εξετάζουν τη διαδοχή των κόμβων, τα κριτήρια διαχωρισμού και τη σύνδεση των χαρακτηριστικών (features) με τις τελικές αποφάσεις. Η οπτικοποίηση θα γίνεται μέσω των κλασικών γραφημάτων Δέντρων Απόφασης, προσφέροντας τη δυνατότητα λεπτομερούς εξερεύνησης, καθώς και της επιλογής λήψης συγκεκριμένων δέντρων σε .png μορφή για περαιτέρω χρήση.

Κατηγοριοποίηση Νέων Στιγμιότυπων: Τέλος, η εφαρμογή θα παρέχει στους χρήστες τη δυνατότητα κατηγοριοποίησης νέων στιγμιότυπων (instances), χρησιμοποιώντας τα μοντέλα που έχουν ήδη δημιουργήσει. Συγκεκριμένα, οι χρήστες θα μπορούν να ανεβάζουν (upload) μη ταξινομημένα σύνολα δεδομένων προς κατηγοριοποίηση, ενώ τα αποτελέσματα της τελικής κατηγοριοποίησης θα παρουσιάζονται σε μορφή πίνακα, διευκολύνοντας έτσι την επισκόπηση και κατανόηση των προβλέψεων της κάθε ετικέτας (label). Επιπλέον, οι χρήστες θα μπορούν να κατεβάζουν (download) το τελικό ταξινομημένο σύνολο δεδομένων σε .csv μορφή αλλά και να βλέπουν την αξιολόγησή του μέσω ενός popup modal που θα προσφέρει μια ολοκληρωμένη επισκόπηση της απόδοσης.

4.2 Αρχιτεκτονική της Εφαρμογής

Όπως προαναφέρθηκε, ο στόχος της εφαρμογής «MultilabelDTree» είναι να αξιοποιήσει προηγμένες τεχνικές αυτοματοποιημένης μηχανικής μάθησης (AutoML), δίνοντας τη δυνατότητα στους χρήστες να δημιουργούν δικά τους μοντέλα και να κατηγοριοποιούν δεδομένα πολλαπλών ετικετών (multi-label), με τρόπο εύκολο και ευέλικτο. Αρχικά, οι εγγεγραμμένοι χρήστες μπορούν να ανεβάζουν τα δικά τους σύνολα δεδομένων προς εκπαίδευση. Σε αυτό το στάδιο, η εφαρμογή επιτρέπει στους χρήστες να προσαρμόζουν τις παραμέτρους τόσο των χαρακτηριστικών (features) και ετικετών (labels) όσο και των δέντρων απόφασης, όπως το μέγιστο βάθος (max depth) και τα ελάχιστα δείγματα ανά φύλλο (min samples leaf). Παράλληλα, οι χρήστες επιλέγουν την τεχνική μετασχηματισμού προβλήματος που επιθυμούν, όπως BinaryRelevance, LabelPowerset ή ClassifierChain, προκειμένου να δημιουργήσουν ένα μοντέλο σύμφωνα με τις ανάγκες τους. Το «MultilabelDTree» παρέχει στους χρήστες την επιπλέον δυνατότητα ενεργοποίησης της αυτόματης επιλογής (Auto Mode) της καλύτερης μεθόδου μετασχηματισμού προβλήματος και των καλύτερων τιμών παραμέτρων, ο προσδιορισμός των οποίων γίνεται βάσει τις υψηλότερη ακρίβειας που επιτυγχάνουν.

Στη συνέχεια, εμφανίζεται η αξιολόγηση του μοντέλου, η οποία υποστηρίζεται από τη μέθοδο *k-fold cross-validation*. Οι χρήστες εδώ, έχουν τη δυνατότητα να δουν λεπτομερείς πληροφορίες απόδοσης για κάθε ετικέτα (label), ενώ ταυτόχρονα παρέχεται και μια συνολική εικόνα της αποτελεσματικότητας του μοντέλου. Αυτό επιτρέπει στους χρήστες να εντοπίζουν τα δυνατά σημεία και τις αδυναμίες των μοντέλων τους, ώστε να αποφασίσουν στο τέλος εάν θέλουν να το αποθηκεύσουν ή όχι για μελλοντική χρήση. Εφόσον οι χρήστες ολοκλήρωσαν τα παραπάνω στάδια, μπορούν να μεταβούν στην σελίδα με τα προεκπαιδευμένα μοντέλα. Εκεί τους δίνεται η δυνατότητα να κατηγοριοποιήσουν τα δεδομένα ενός νέου μη ταξινομημένου συνόλου, με βάση ένα προεκπαιδευμένο μοντέλο που θα επιλέξουν. Τέλος, μπορούν να κατεβάσουν σε .csv μορφή το τελικό ταξινομημένο σύνολο με τις προβλέψεις, αλλά και να δουν τα αποτελέσματα των μετρικών αξιολόγησης.

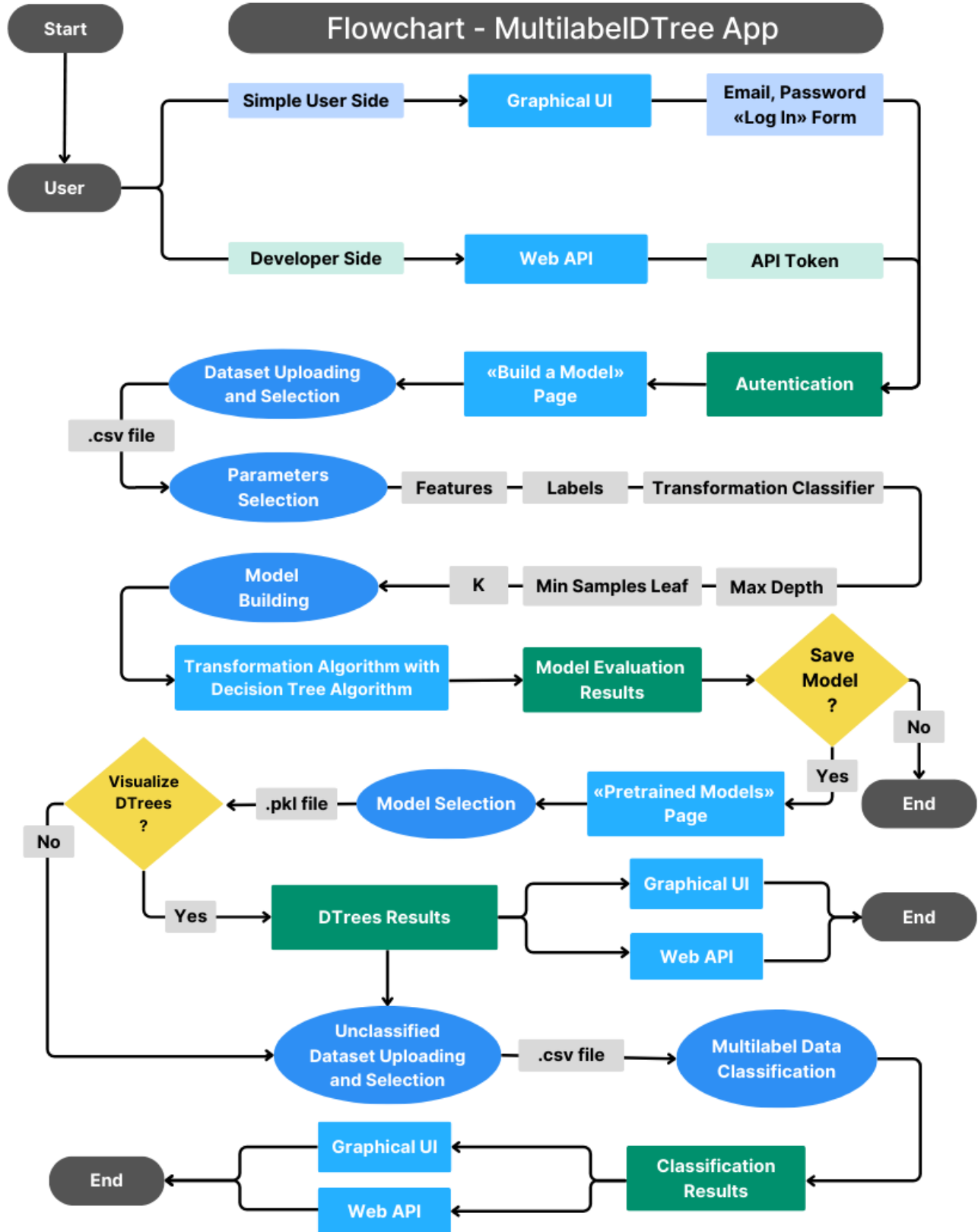
Όλες οι παραπάνω λειτουργίες είναι διαθέσιμες μέσω μιας φιλικής προς τον χρήστη γραφικής διεπαφής όσο και μέσω ενός ελεύθερου διαδικτυακού API. Ειδικότερα, για την πρόσβαση μέσω της γραφικής διεπαφής, οι χρήστες θα πρέπει να εισάγουν τη διεύθυνση email και τον κωδικό πρόσβασης που όρισαν κατά την εγγραφή τους. Από την άλλη, για την ταυτοποίηση μέσω του Web API, θα χρησιμοποιούν το προσωπικό τους API Token που παράγεται αυτόματα κατά την εγγραφή τους. Όσον αφορά την εμφάνιση των αποτελεσμάτων-δεδομένων, αυτά θα παρουσιάζονται σε μορφή JSON όταν χρησιμοποιείται το Web API. Το Σχήμα 4.1, παρουσιάζει αναλυτικά, μέσα από ένα διάγραμμα ροής, τη σειρά των βημάτων και λειτουργιών που ακολουθεί ο χρήστης στην εφαρμογή.

Η αρχιτεκτονική του «MultilabelDTree» περιλαμβάνει πέντε κύρια μέρη. Τον πυρήνα της εφαρμογής αποτελεί το Web API, το οποίο έχει αναπτυχθεί με τη γλώσσα προγραμματισμού PHP. Το Web API είναι μία γέφυρα επικοινωνίας μεταξύ του Backend και του Frontend, ενώ διαχειρίζεται και την εκτέλεση των περισσότερων λειτουργιών.

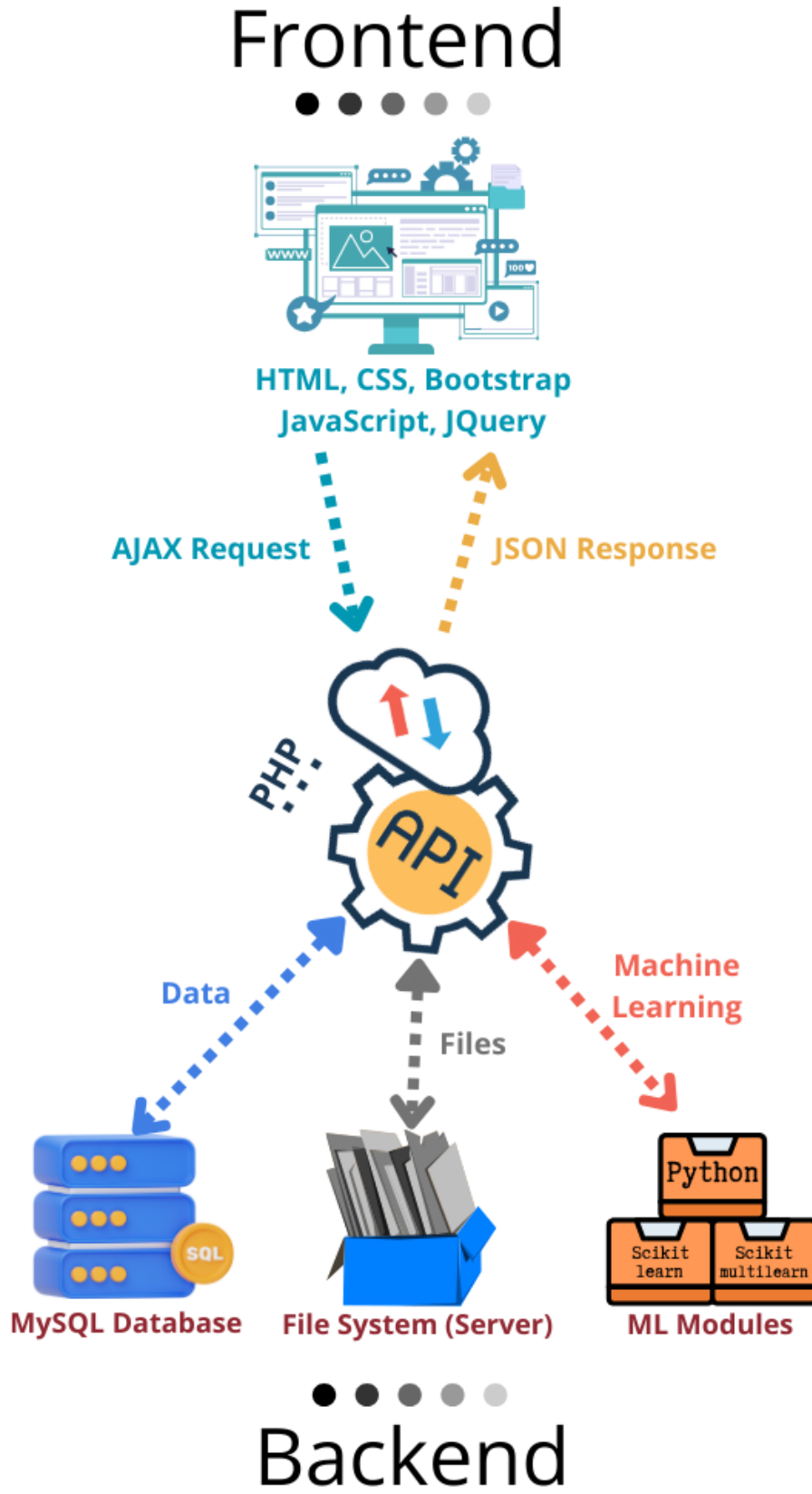
Στο Backend χρησιμοποιείται η MySQL ως η κύρια βάση δεδομένων, η οποία διαχειρίζεται τις ενέργειες που σχετίζονται με την εγγραφή των χρηστών, την αυθεντικοποίησή τους, καθώς και την αποθήκευση των προεκπαιδευμένων μοντέλων με τις ετικέτες τους (labels). Συνεπώς, το Web API αλληλεπιδρά και με το σύστημα αρχείων (file system) του server, για την αποθήκευση των αρχείων του κάθε χρήστη, όπως τα datasets και τα μοντέλα.

Επιπρόσθετα, χρησιμοποιούνται και ορισμένα modules, τα οποία έχουν αναπτυχθεί σε Python, χρησιμοποιώντας τις βασικές βιβλιοθήκες Scikit-learn και Scikit-multilearn. Αυτά τα modules χρησιμοποιούνται για την δημιουργία και εκτέλεση των AutoML αλγορίθμων, καθώς και των κύριων εργασιών Μηχανικής Μάθησης που διαθέτει η εφαρμογή μας. Έτσι, το Web API καλεί όλα αυτά τα modules για να επικοινωνεί με τη γραφική διεπαφή, επιτρέποντας στους χρήστες να έχουν πρόσβαση στις λειτουργίες της εφαρμογής μέσα από ένα εύχρηστο περιβάλλον.

Τέλος, στο Frontend, η γραφική διεπαφή και λειτουργικότητα αναπτύχθηκε με HTML, CSS, Bootstrap, JavaScript και JQuery, όπως αναφέρθηκε και στο προηγούμενο κεφάλαιο με τις τεχνολογίες. Συνολικά, η αρχιτεκτονική του «MultilabelDTree» παρουσιάζεται στο Σχήμα 4.2.



Σχήμα 4.1: Διάγραμμα Ροής του MultilabelDTree.



Σχήμα 4.2: Αρχιτεκτονική του MultilabelDTree.

4.3 Τύποι Συνόλων Δεδομένων

Η εφαρμογή «MultilabelDTree» περιλαμβάνει δύο διαφορετικούς τύπους συνόλων δεδομένων, ανάλογα με το ποιοι χρήστες έχουν πρόσβαση σε αυτά. Η πρώτη κατηγορία περιλαμβάνει τα ιδιωτικά σύνολα δεδομένων (private datasets), τα οποία κάθε εγγεγραμμένος χρήστης μπορεί να ανεβάσει στον προσωπικό του φάκελο. Τα προσωπικά σύνολα ενός χρήστη είναι διαθέσιμα αποκλειστικά για τον ίδιο και δεν υπάρχει δυνατότητα να τα μοιραστεί με άλλους χρήστες της εφαρμογής. Η δεύτερη κατηγορία αφορά τα δημόσια σύνολα δεδομένων (public datasets). Σε αυτή την περίπτωση, ένας χρήστης μπορεί να ανεβάσει αρχεία στον δημόσιο φάκελο που διαθέτει η εφαρμογή, καθιστώντας τα διαθέσιμα σε όλους για προβολή και χρήση.

Η εν λόγω λειτουργία, που αφορά τον διαχωρισμό των συνόλων δεδομένων σε δύο τύπους, αποσκοπεί κυρίως στη διευκόλυνση των χρηστών, επιτρέποντάς τους να δημιουργούν αποδοτικά μοντέλα σύμφωνα με τις ανάγκες τους. Αξίζει να σημειωθεί ότι η χρήση δημόσιων συνόλων δεδομένων είναι διαθέσιμη μόνο κατά τη λειτουργία εκπαίδευσης και δημιουργίας νέων μοντέλων που βρίσκονται στη σελίδα «Build a Model». Αντίθετα, στην λειτουργία κατηγοριοποίησης νέων στιγμιοτύπων, της σελίδας «Pretrained Models», η επιλογή του κατάλληλου αταξινομητου συνόλου γίνεται αποκλειστικά απ'τον ίδιο τον χρήστη, χωρίς να υπάρχει πρόσβαση σε αυτό από τους άλλους.

Το δικαίωμα για μεταφόρτωση και διαγραφή δημόσιων συνόλων δεδομένων παραχωρείται αποκλειστικά στους εγγεγραμμένους χρήστες από τον διαχειριστή της εφαρμογής, εφόσον έχουν υποβάλει οι ίδιοι σχετικό αίτημα και έχουν λάβει την απαραίτητη έγκριση.

4.4 Χρήστες της Εφαρμογής

Οι χρήστες της εφαρμογής κατηγοριοποιούνται σε τέσσερις βασικές ομάδες, ανάλογα με τα δικαιώματα που διαθέτουν:

Διαχειριστής: Ο χρήστης αυτής της κατηγορίας είναι υπεύθυνος για τη συνολική διαχείριση του συστήματος και διαθέτει πλήρη δικαιώματα στον χειρισμό των δημόσιων συνόλων δεδομένων. Επίσης, έχει τη δυνατότητα να ορίζει, μέσω του RDBMS, ποιοι άλλοι χρήστες μπορούν να αποκτήσουν δικαίωμα επεξεργασίας στα δημόσια σύνολα δεδομένων.

Εγγεγραμμένοι Χρήστες με Δημόσια Δικαιώματα: Αυτοί οι χρήστες έχουν τη δυνατότητα να χειρίζονται τα δημόσια σύνολα δεδομένων, και άρα να ανεβάζουν ή να διαγράφουν δημόσια αρχεία. Τις ενέργειες αυτές τις επιτυγχάνουν βάσει δικαιωμάτων που τους έχει δοθεί από τον διαχειριστή.

Εγγεγραμμένοι Χρήστες χωρίς Δημόσια Δικαιώματα: Από την άλλη, αυτή η ομάδα χρηστών δεν έχει το δικαίωμα να εκτελέσει οποιαδήποτε ενέργεια σχετίζεται με τα δημόσια σύνολα δεδομένων, διότι δεν τους έχει παραχωρηθεί η σχετική άδεια από τον διαχειριστή της εφαρμογής.

Μη Εγγεγραμμένοι Χρήστες: Τέλος, οι συγκεκριμένοι χρήστες δεν έχουν πραγματοποιήσει εγγραφή στην εφαρμογή «MultilabelDTree», και επομένως δεν μπορούν να έχουν πρόσβαση στις δυνατότητες της εφαρμογής με τα αντίστοιχα δικαιώματα.

4.5 Υλοποίηση του Backend

4.5.1 Βάση Δεδομένων

Όπως προαναφέρθηκε, η βάση δεδομένων της εφαρμογής είναι η MySQL, στόχος της οποίας είναι η διαχείριση των ενεργειών που σχετίζονται με την εγγραφή των χρηστών, την αυθεντικοποίησή τους, καθώς και την αποθήκευση των προεκπαιδευμένων μοντέλων με τις ετικέτες τους (labels). Έτσι, δημιουργήθηκε μία βάση στο λογισμικό HeidiSQL, με όνομα 'multilabeldtree_db', για την κάλυψη των αναγκών της εφαρμογής.

Το HeidiSQL είναι ένα δωρεάν λογισμικό ανοιχτού κώδικα, σχεδιασμένο για χρήστες που εργάζονται με συστήματα διαχείρισης βάσεων δεδομένων. Το εργαλείο αυτό προσφέρει μια εύκολη και φιλική εμπειρία χρήσης, επιτρέποντας τη σύνδεση με διάφορα είδη βάσεων δεδομένων, όπως MariaDB, MySQL, Microsoft SQL, PostgreSQL, SQLite, Interbase και Firebird, με στόχο την διαχείριση και επεξεργασία των δεδομένων μίας εφαρμογής [60].

Για την επίτευξη των παραπάνω ενεργειών, δημιουργήθηκαν τέσσερις πίνακες στην βάση, με ονόματα 'users', 'verify_account', 'models' και 'labels' αντίστοιχα. Συγκεκριμένα, ο πίνακας 'users' δημιουργήθηκε για να αποθηκεύει τα προσωπικά δεδομένα των εγγεγραμμένων χρηστών της εφαρμογής, ενώ ο πίνακας 'verify_account' δημιουργήθηκε για να αποθηκεύει προσωρινές πληροφορίες επαλήθευσης των λογαριασμών των χρηστών. Επιπλέον, ο πίνακας 'models' χρησιμεύει για την αποθήκευση των μοντέλων πολλαπλών ετικετών με τις αντίστοιχες πληροφορίες τους. Τέλος, ο πίνακας 'labels' δημιουργήθηκε για να αποθηκεύει τις ετικέτες των αντίστοιχων multi-label μοντέλων που περιέχει ο πίνακας 'models'.

Ο Πίνακας users (4.1) περιλαμβάνει οκτώ βασικά πεδία.

- Το πεδίο 'id' είναι το κύριο κλειδί του πίνακα και αποτελεί το μοναδικό αναγνωριστικό του κάθε χρήστη. Το πεδίο αυτό χρησιμοποιεί την εντολή 'auto_increment', που σημαίνει ότι το id δημιουργείται και εισάγεται με αυτόματο τρόπο στον πίνακα.
- Το πεδίο 'fname', αντιπροσωπεύει το όνομα του χρήστη.
- Το πεδίο 'lname', αντιπροσωπεύει το επώνυμο του χρήστη.
- Το πεδίο 'email', περιλαμβάνει τη διεύθυνση email του χρήστη.
- Το πεδίο 'pass', περιλαμβάνει τον κωδικό λογαριασμού του χρήστη για την είσοδό του στην εφαρμογή. Για λόγους ασφαλείας και εμπιστευτικότητας, ο κωδικός αυτός μετατρέπεται αρχικά σε μορφή hash, μέσω του αλγορίθμου 'CRYPT_BLOWFISH' της php συνάρτησης password_hash(), και έπειτα καταχωρείται στην Βάση Δεδομένων της εφαρμογής.
- Το πεδίο 'token', περιέχει έναν μοναδικό κωδικό που δίνει τη δυνατότητα στους χρήστες να αποκτούν πρόσβαση στο Web API και να υποβάλλουν αιτήματα σε αυτό. Ο κωδικός δημιουργείται και εισάγεται αυτόματα, ακολουθώντας την εξής διαδικασία: Αρχικά, συνδυάζεται το όνομα του χρήστη με την έξοδο της συνάρτησης 'NOW()' της MySQL, η οποία επιστρέφει την τρέχουσα ημερομηνία και ώρα. Έπειτα, με τη βοήθεια του αλγορίθμου MD5, το

συνδυασμένο αποτέλεσμα μετατρέπεται σε hash, το οποίο αποδίδει έναν δεκαεξαδικό αριθμό 32 χαρακτήρων.

- Το πεδίο 'email_verification', δείχνει εάν ο χρήστης έχει επιβεβαιώσει την διεύθυνση email του. Το πεδίο αυτό έχει την προεπιλεγμένη τιμή '0', η οποία μπορεί αν γίνει '1' μόνο εάν ο χρήστης επιβεβαιώσει επιτυχώς την διεύθυνση email.
- Το πεδίο 'public_permission', δείχνει εάν ο συγκεκριμένος χρήστης έχει δικαίωμα επεξεργασίας στα δημόσια σύνολα δεδομένων. Ομοίως, το πεδίο έχει ως προεπιλεγμένη τιμή το '0', και εφόσον ο διαχειριστής της εφαρμογής παραχωρήσει τη σχετική άδεια στον χρήστη, θα λάβει την τιμή '1'.

Πίνακας 4.1: Πίνακας users της Βάσης Δεδομένων

Name	Datatype	Length/Set
id	INT	11
fname	VARCHAR	50
lname	VARCHAR	50
email	VARCHAR	50
pass	VARCHAR	100
token	VARCHAR	100
email_verification	TINYINT	1
public_permission	TINYINT	1

Ο Πίνακας verify_account (4.2) περιλαμβάνει τα παρακάτω τέσσερα πεδία.

- Το πεδίο 'id', είναι το κύριο κλειδί του πίνακα και αποτελεί το μοναδικό αναγνωριστικό του κάθε λογαριασμού επιβεβαίωσης.
- Το πεδίο 'user_id', αποτελεί ξένο κλειδί του πεδίου 'id' στον πίνακα 'users', αντιπροσωπεύοντας τον χρήστη με τον λογαριασμό επιβεβαίωσης.
- Το πεδίο 'verif_key', περιέχει έναν κωδικό που απαιτείται για την ολοκλήρωση της επιβεβαίωσης μιας ενέργειας σε έναν λογαριασμό. Ο κωδικός αυτός αποτελείται από μια τυχαία ακολουθία χαρακτήρων, η οποία παράγεται με τη χρήση του αλγορίθμου MD5 και της συνάρτησης random_bytes() της PHP.
- Το πεδίο 'creation_time', καταγράφει την ημερομηνία και την ώρα που δημιουργήθηκε ο κωδικός επιβεβαίωσης. Αυτό το πεδίο είναι χρήσιμο για τον έλεγχο της ισχύος του κωδικού, δηλαδή εάν έχει λήξει ή όχι.

Πίνακας 4.2: Πίνακας verify_account της Βάσης Δεδομένων

Name	Datatype	Length/Set
id	INT	11
user_id	INT	11
verif_key	VARCHAR	100
creation_time	TIMESTAMP	-

Ο Πίνακας models (4.3) περιλαμβάνει τέσσερα πεδία.

- Το πεδίο 'id', είναι το κύριο κλειδί του πίνακα και αποτελεί το μοναδικό αναγνωριστικό του κάθε μοντέλου.
- Το πεδίο 'user_id', αποτελεί ξένο κλειδί του πεδίου 'id' στον πίνακα 'users', αντιπροσωπεύοντας τον χρήστη που δημιούργησε το μοντέλο.
- Το πεδίο 'transformation_approach', περιέχει το όνομα της μεθόδου/προσέγγισης μετασχηματισμού προβλήματος που χρησιμοποιήθηκε στο αντίστοιχο μοντέλο.
- Το πεδίο 'model_name', περιέχει το όνομα του μοντέλου πολλαπλών ετικετών που δημιουργήθηκε από τον χρήστη.

Πίνακας 4.3: Πίνακας models της Βάσης Δεδομένων

Name	Datatype	Length/Set
id	INT	11
user_id	INT	11
transformation_approach	VARCHAR	50
model_name	VARCHAR	50

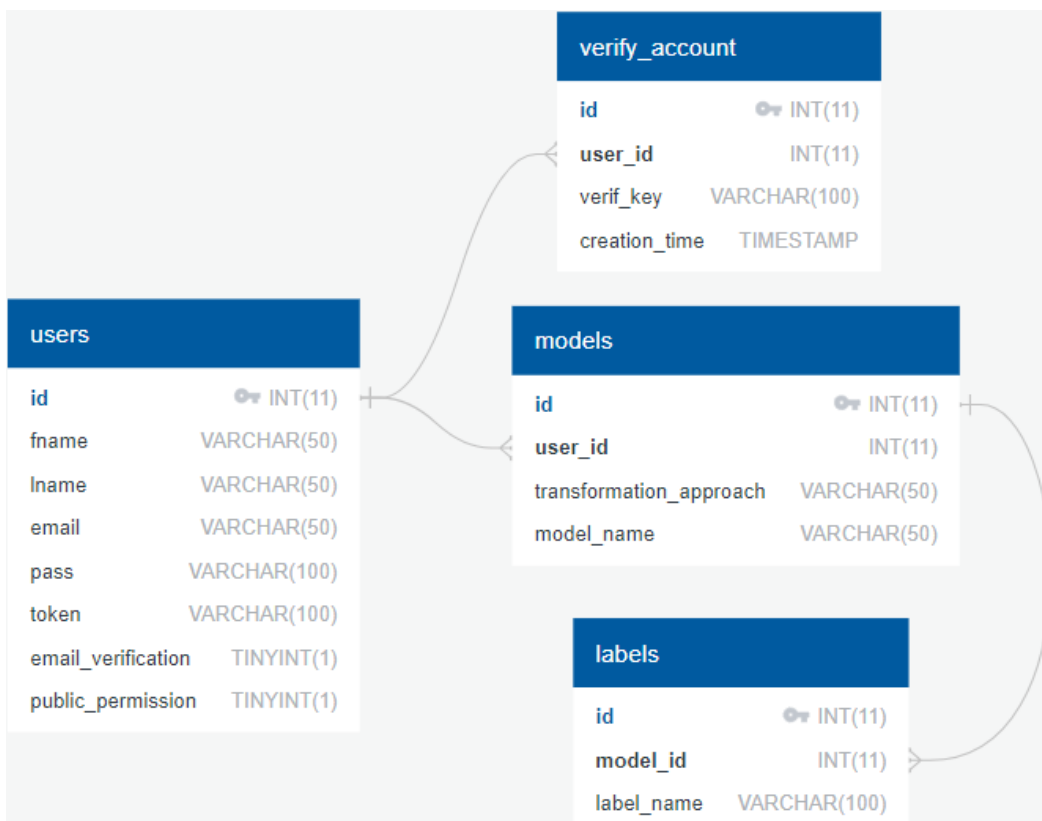
Ο Πίνακας labels (4.4) περιλαμβάνει τα εξής τρία πεδία.

- Το πεδίο 'id', είναι το κύριο κλειδί του πίνακα και αποτελεί το μοναδικό αναγνωριστικό της κάθε ετικέτας.
- Το πεδίο 'model_id', αποτελεί ξένο κλειδί του πεδίου 'id' στον πίνακα 'models', αντιπροσωπεύοντας το μοντέλο στο οποίο ανήκει η ετικέτα.
- Το πεδίο 'label_name', περιέχει το όνομα της συγκεκριμένης ετικέτας.

Πίνακας 4.4: Πίνακας labels της Βάσης Δεδομένων

Name	Datatype	Length/Set
id	INT	11
model_id	INT	11
label_name	VARCHAR	100

Στο Σχήμα 4.3 παρουσιάζεται το διάγραμμα οντοτήτων-συσχετίσεων (ER diagram) της Βάσης Δεδομένων της εφαρμογής, το οποίο δημιουργήθηκε με τη χρήση του λογισμικού QuickDBD.



Σχήμα 4.3: ER Diagram της Βάσης Δεδομένων.

4.5.2 Web API

Το Web API αποτελεί τον πυρήνα λειτουργίας της εφαρμογής, διότι μέσω αυτού πραγματοποιείται η εκτέλεση των περισσότερων λειτουργιών, διευκολύνοντας σημαντικά την επικοινωνία μεταξύ Frontend και Backend. Για να αποκτήσει κάποιος πρόσβαση στο Web API του «MultilabelDTree», απαιτείται η κλήση του κατάλληλου endpoint έτσι ώστε να πραγματοποιηθεί η συγκεκριμένη λειτουργία στην εφαρμογή. Το endpoint, μετά την κατάλληλη κλήση, προχωρά στην εκτέλεση του σχετικού PHP αρχείου και επιστρέφει τα αποτελέσματα σε μορφή JSON μετά την ολοκλήρωση της διαδικασίας. Επίσης, κάθε endpoint απαιτεί την παροχή των κατάλληλων παραμέτρων, οι οποίες διαμορφώνονται ανάλογα με την εκάστοτε περίπτωση, δηλαδή ανάλογα με την μέθοδο HTTP που χρησιμοποιείται. Ειδικότερα, οι μέθοδοι POST, PUT και DELETE εισάγουν τις παραμέτρους στο σώμα (body) του αιτήματος ενώ, από την άλλη, η μέθοδος GET τοποθετεί τις παραμέτρους στο URL του αιτήματος, μετά τον ειδικό χαρακτήρα ' ? ', ο οποίος υποδεικνύει ότι ακολουθούν συγκεκριμένες παράμετροι. Αξίζει να σημειωθεί ότι κάθε παράμετρος διαχωρίζεται από τις υπόλοιπες με το σύμβολο ' & ', ώστε να διακρίνονται και να είναι κατανοητές στο πλαίσιο του αιτήματος.

Το σύνολο των δημόσιων API Endpoints της εφαρμογής φαίνεται αναλυτικά στον Πίνακα 4.5. Κάθε ένα από τα endpoints-αρχεία php είναι αποθηκευμένο σε έναν ειδικό κατάλογο της εφαρμογής, που βρίσκεται στον σύνδεσμο <https://kclusterhub.iee.ihu.gr/multilabeldtree/server/php/api/>. Για παράδειγμα εάν ένας χρήστης επιθυμεί να διαγράψει ένα σύνολο δεδομένων (dataset) από τον προσωπικό του κατάλογο, θα πρέπει να στείλει αίτημα με την μέθοδο DELETE στο endpoint: <https://kclusterhub.iee.ihu.gr/multilabeldtree/server/php/api/destroyDataset.php>, έχοντας τοποθετήσει στο σώμα (body), του αιτήματος, τις κατάλληλες παραμέτρους: { "token": "yourToken", "folder": "public|private", "file": "selectedfile.csv" }. Αντίστοιχα, εάν ο χρήστης επιθυμεί να εμφανίσει την λίστα των ανεβασμένων συνόλων δεδομένων, τότε θα πρέπει να καλέσει με την μέθοδο GET το endpoint που σχετίζεται με αυτή την λειτουργία, δηλαδή το <https://kclusterhub.iee.ihu.gr/multilabeldtree/server/php/api/receiveDatasets.php?token=yourToken>, προκειμένου να λάβει τα απαραίτητα δεδομένα από την απόκριση (response).

Παρακάτω παρατίθενται μερικά σημαντικά παραδείγματα PHP κώδικα του Web API της εφαρμογής, προκειμένου να διευκρινιστεί και να γίνει κατανοητή η χρησιμότητά του. Μέσα από αυτά τα παραδείγματα, αναλύονται βασικές λειτουργίες και διαδικασίες που σχετίζονται με την αλληλεπίδραση του χρήστη στην εφαρμογή. Ειδικότερα, δίνονται τα παρακάτω έξι παραδείγματα:

- Κώδικας Ελέγχου Ορθότητας Στοιχείων κατά την Εγγραφή Χρήστη (Sign Up Fields Validation).
- Κώδικας Ταυτοποίησης Χρήστη μέσω Token (User Identification).
- Κώδικας Αιτήματος προς την Βάση Δεδομένων (Database Request).
- Κώδικας Ανεβάσματος ενός Δημόσιου/Ιδιωτικού Multi-label Συνόλου Δεδομένων (Multi-label Dataset Uploading).
- Κώδικας Επιστροφής Περιεχομένου ενός Multi-label Συνόλου Δεδομένων (Multi-label Dataset Content).
- Κώδικας Αιτήματος προς ένα Python Αρχείο (Python Request).

Πίνακας 4.5: Public API Endpoints της Εφαρμογής

A/A	Μέθοδος HTTP	Endpoint	Λειτουργία
1	POST	signup.php	Εγγραφή Χρήστη
2	POST	login.php	Είσοδος Χρήστη
3	POST	editAccount.php	Ενημέρωση Πληροφοριών Χρήστη
4	DELETE	deleteAccount.php	Διαγραφή Λογαριασμού Χρήστη
5	GET	receiveDatasets.php? token={token}	Επιστροφή Συνόλων Δεδομένων Εκπαίδευσης
6	POST	uploadDataset.php	Ανέβασμα ενός Συνόλου Δεδομένων Εκπαίδευσης
7	GET	multilabelDatasetContent.php? token={token}& file={selectedFile}& folder={public private}	Επιστροφή Περιεχομένου ενός Multi-label Συνόλου Δεδομένων Εκπαίδευσης
8	DELETE	destroyDataset.php	Διαγραφή ενός Συνόλου Δεδομένων Εκπαίδευσης
9	GET	receiveUnclassifiedDatasets.php? token={token}	Επιστροφή Μη Ταξινομημένων Συνόλων Δεδομένων
10	POST	uploadUnclassifiedDataset.php	Ανέβασμα ενός Μη Ταξινομημένου Συνόλου Δεδομένων
11	GET	multilabelUnclassified DatasetContent.php? token={token}& file={selectedFile}	Επιστροφή Περιεχομένου ενός Multi-label Μη Ταξινομημένου Συνόλου Δεδομένων
12	DELETE	destroyUnclassifiedDataset.php	Διαγραφή ενός Μη Ταξινομημένου Συνόλου Δεδομένων
13	GET	receiveModels.php? token={token}	Επιστροφή Προεκπαιδευμένων Μοντέλων
14	GET	multilabelModelContent.php? token={token}& file={selectedFile}	Επιστροφή Περιεχομένου ενός Multi-label Προεκπαιδευμένου Μοντέλου
15	DELETE	destroyModel.php	Διαγραφή ενός Προεκπαιδευμένου Μοντέλου
16	GET	visualizeDTrees.php? token={token}& file={selectedFile}	Οπτικοποίηση Δέντρων Απόφασης ενός Προεκπαιδευμένου Μοντέλου
17	POST	multilabelCrossValidation.php	Δημιουργία και Αξιολόγηση του Multi-label Μοντέλου. Επιστροφή Αποτελεσμάτων των Μετρικών Αξιολόγησης
18	POST	saveMultilabelModel.php	Αποθήκευση του Multi-label Μοντέλου που Δημιούργησε ο Χρήστης
19	POST	classifyMultilabelData.php	Κατηγοριοποίηση ενός Αταξινομητου Συνόλου Δεδομένων με βάση ένα Προεκπαιδευμένο Μοντέλο. Επιστροφή Αποτελεσμάτων των Μετρικών Αξιολόγησης

Ο κώδικας που ακολουθεί, υλοποιεί τον έλεγχο ορθότητας των στοιχείων που εισάγει ο χρήστης στην φόρμα κατά την εγγραφή του. Πρώτα, ελέγχει τη μέθοδο HTTP του αιτήματος και εάν δεν είναι POST, επιστρέφει μήνυμα σφάλματος "Method not allowed" με κατάσταση HTTP 405. Στη συνέχεια, πραγματοποιεί έλεγχο για τα υποχρεωτικά πεδία της φόρμας (`fname`, `lname`, `email`, `pass`). Εάν κάποιο από αυτά τα πεδία λείπει τότε επιστρέφει το σχετικό μήνυμα σφάλματος, όπως για παράδειγμα "First Name is Required", με κατάσταση HTTP 400, εάν δεν έχει εισαχθεί το όνομα του χρήστη. Η εκτέλεση του κώδικα σταματά σε κάθε περίπτωση σφάλματος, εξασφαλίζοντας ότι τα δεδομένα που θα αποσταλούν είναι πλήρη και έγκυρα.

Κώδικας Ελέγχου Ορθότητας Στοιχείων κατά την Εγγραφή Χρήστη:

```
$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(['errormsg'=>"Method not allowed."]);
    exit;
}

// Form's Required Fields Validation.
if(!isset($input['fname'])) {
    header("HTTP/1.1 400 Bad Request");
    print json_encode(['errormsg'=>"First Name is Required."]);
    exit;
}

if(!isset($input['lname'])) {
    header("HTTP/1.1 400 Bad Request");
    print json_encode(['errormsg'=>"Last Name is Required."]);
    exit;
}

if(!isset($input['email'])) {
    header("HTTP/1.1 400 Bad Request");
    print json_encode(['errormsg'=>"Email is Required."]);
    exit;
}

if(!isset($input['pass'])) {
    header("HTTP/1.1 400 Bad Request");
    print json_encode(['errormsg'=>"Password is Required."]);
    exit;
}
```

```

if(!isset($_input['pass_confirm'])) {
    header("HTTP/1.1 400 Bad Request");
    print json_encode(['errormsg'=>"Please Confirm Password."]);
    exit;
}

```

Ο παρακάτω κώδικας υλοποιεί τη διαδικασία ταυτοποίησης ενός χρήστη μέσω token. Περιλαμβάνει μια συνάρτηση 'token_exists' που ελέγχει αν ένα token υπάρχει στη βάση δεδομένων, μετρώντας τις εγγραφές στον πίνακα users που αντιστοιχούν σε αυτό. Αν το token βρεθεί, επιστρέφεται true, διαφορετικά false. Στη συνέχεια, ο κώδικας ελέγχει αν το token έχει αποσταλεί στο αίτημα της μεθόδου POST. Εάν το token λείπει, επιστρέφεται μήνυμα σφάλματος "Token is not set" με κατάσταση HTTP 400. Εάν το token δεν υπάρχει στη βάση δεδομένων, επιστρέφεται μήνυμα σφάλματος "Token doesn't exist", ομοίως με κατάσταση HTTP 400. Αυτός ο έλεγχος διασφαλίζει την εγκυρότητα του token πριν επιτραπεί οποιαδήποτε περαιτέρω ενέργεια στην εφαρμογή.

Κώδικας Ταυτοποίησης Χρήστη μέσω Token:

```

// Function to check token's expiration.
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;
    }
}

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

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

```

Ο παρακάτω PHP κώδικας υλοποιεί τον έλεγχο εισόδου ενός χρήστη στην εφαρμογή. Αρχικά, λαμβάνει το email και τον κωδικό πρόσβασης από την φόρμα εισόδου του χρήστη. Στη συνέχεια, πραγματοποιεί αναζήτηση στη βάση δεδομένων για να βρει αν υπάρχει λογαριασμός με το συγκεκριμένο email. Εάν ο λογαριασμός δεν υπάρχει, επιστρέφει μήνυμα σφάλματος "Account does not exist". Ελέγχει επίσης αν το email του χρήστη έχει επαληθευτεί και επιστρέφει μήνυμα σφάλματος εάν δεν έχει γίνει. Τέλος, επαληθεύει τον κωδικό πρόσβασης που δόθηκε από τον χρήστη και το συγκρίνει με τον αποθηκευμένο στη βάση δεδομένων, χρησιμοποιώντας τη php μέθοδο 'password_verify'. Εάν ο κωδικός πρόσβασης είναι λανθασμένος, επιστρέφεται μήνυμα σφάλματος "Wrong password". Σε κάθε περίπτωση σφάλματος, διακόπτεται η εκτέλεση του κώδικα και επιστρέφεται HTTP 400 status.

Κώδικας Αιτήματος προς την Βάση Δεδομένων:

```
$email = $input['email'];
$pass = $input['pass'];

// Checking User's Account Information.
$query = 'select * from users where email=?';
$stmt = $mysqli->prepare($query);
$stmt->bind_param('s', $email);
$stmt->execute();
$res = $stmt->get_result();
$res = $res->fetch_assoc();

if(empty($res)) {
    header("HTTP/1.1 400 Bad Request");
    print json_encode(['errormsg'=>"Account does not exist."]);
    exit;
}

if(!$res['email_verification']) {
    header("HTTP/1.1 400 Bad Request");
    print json_encode(['errormsg'=>"Email did not verified."]);
    exit;
}

if(!password_verify($pass, $res['pass'])) {
    header("HTTP/1.1 400 Bad Request");
    print json_encode(['errormsg'=>"Wrong password."]);
    exit;
}
```

Ο κώδικας που ακολουθεί, ελέγχει και διαχειρίζεται τη διαδικασία ανεβάσματος ενός αρχείου-συνόλου δεδομένων, διαφοροποιώντας τις ενέργειες ανάλογα με τον φάκελο προορισμού (public ή private). Εάν ο φάκελος είναι public, αρχικά ελέγχεται εάν ο χρήστης έχει το δικαίωμα ανεβάσματος δημόσιων αρχείων, μέσω του token που συνδέεται με το πεδίο 'public_permission'

της βάσης δεδομένων. Αν ο χρήστης δεν έχει δικαιώματα, επιστρέφεται μήνυμα σφάλματος "You aren't allowed to upload public datasets" με κατάσταση HTTP 403. Στη συνέχεια, ελέγχεται αν το αρχείο υπάρχει ήδη στη διαδρομή ../../py/public/datasets/. Εάν όντως υπάρχει, επιστρέφεται μήνυμα σφάλματος "Dataset already exists" με κατάσταση HTTP 400. Αν περάσουν όλοι οι έλεγχοι, το αρχείο μετακινείται στη συγκεκριμένη διαδρομή με τη χρήση της συνάρτησης 'move_uploaded_file', δηλαδή αποθηκεύεται το σύνολο δεδομένων στο file system του server. Αντίθετα, εάν ο φάκελος είναι τύπου private, ο κώδικας επιστρέφει το email του συγκεκριμένου χρήστη, το μετατρέπει σε hash μορφή μέσω του αλγορίθμου MD5, και έπειτα αποθηκεύει το σύνολο δεδομένων στο προσωπικό directory του χρήστη εντός του file system.

Κώδικας Ανεβάσματος ενός Δημόσιου/Ιδιωτικού Multi-label Συνόλου Δεδομένων:

```
$upload = false;

if($folder == "public") {

    $query = 'select public_permission from users where token=?';
    $st = $mysqli->prepare($query);
    $st->bind_param('s', $_POST['token']);
    $st->execute();
    $res = $st->get_result();
    $public_permission = $res->fetch_assoc()['public_permission'];

    if($public_permission == 0) {
        header("HTTP/1.1 403 Forbidden");
        print json_encode(['errormsg'=>"You aren't allowed to
        upload public datasets."]);
        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(['errormsg'=>"Dataset already exists."]);
        exit;
    }

    $upload = move_uploaded_file($_FILES["file"]["tmp_name"],
    $file_path);
}
else {
```

```

$email = user_mail($_POST['token']);
$hash_user = md5($email);

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

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

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

if(!$upload) {
    header("HTTP/1.1 400 Bad Request");
    print json_encode(['errormsg'=>"Unable to upload dataset."]);
    exit;
}

print json_encode(['message'=>"Dataset successfully uploaded."]);

```

Ο παρακάτω κώδικας διαβάζει το περιεχόμενο ενός .csv αρχείου που περιέχει δεδομένα ενός multi-label συνόλου. Αρχικά, ανοίγει το αρχείο από την καθορισμένη διαδρομή '\$file_path', μέσω της συνάρτησης 'fopen', και έπειτα επεξεργάζεται την κάθε γραμμή χρησιμοποιώντας τη συνάρτηση 'fgetcsv'. Στην συνέχεια, ελέγχει αν η κάθε γραμμή περιέχει δεδομένα, αποκλείοντας τις κενές γραμμές από το σύνολο. Για κάθε μη κενή γραμμή, αποθηκεύει τα δεδομένα σε έναν πολυδιάστατο πίνακα '\$csv_array'. Μετά την επεξεργασία όλων των γραμμών, το αρχείο κλείνει με τη συνάρτηση 'fclose'. Αυτός ο κώδικας διασφαλίζει ότι το περιεχόμενο ενός multi-label συνόλου επιστρέφεται καθαρό, χωρίς να υπάρχουν κενές εγγραφές.

Κώδικας Επιστροφής Περιεχομένου ενός Multi-label Συνόλου Δεδομένων:

```

$row = 0;
if(($open_file = fopen($file_path, "r")) !== FALSE) {
    while (($row_data = fgetcsv($open_file, 2048, ",", "")) !== FALSE) {
        // Empty rows validation. If such rows exist,
        // exclude them from the dataset.
        if (array_filter($row_data)) {
            $countFields = count($row_data);
            for($i = 0; $i < $countFields; $i++) {
                $csv_array[$row][$i] = $row_data[$i];
            }
        }
    }
}

```

```

        $row++;
    }
}
fclose($open_file);

```

Τέλος, ο κώδικας που ακολουθεί, εκτελεί μια python εντολή για τη διαδικασία Cross-Validation σε δεδομένα ενός multi-label συνόλου. Για την επίτευξη αυτού χρησιμοποιείται το python script `multilabel_dt_crossvalidation.py` που βρίσκεται υλοποιημένο στο `py` directory. Τα απαραίτητα δεδομένα, όπως η διαδρομή του αρχείου '`$file_path`', τα χαρακτηριστικά του συνόλου '`$featuresImplode`', οι ετικέτες του '`$labelsImplode`', και άλλες παράμετροι (π.χ. μέγιστο βάθος, ελάχιστα δείγματα ανά φύλλο, αριθμός `k`, και τύπος ταξινομητή), περνιούνται ως ορίσματα στη εντολή. Ο κώδικας χρησιμοποιεί την μέθοδο '`shell_exec`' για να εκτελέσει την python εντολή, ενώ παράλληλα μέσω της `try-catch` πιάνει τυχόν εξαιρέσεις που μπορεί να προκύψουν. Αν προκύψει κάποιο σφάλμα κατά την εκτέλεση ή αν η python εντολή δεν επιστρέψει αποτέλεσμα, τότε εμφανίζεται μήνυμα σφάλματος HTTP 400 με την σχετική περιγραφή. Εάν η εντολή εκτελεστεί επιτυχώς, το αποτέλεσμα εκτυπώνεται στην έξοδο.

Κώδικας Αιτήματος προς ένα Python Αρχείο:

```

$results;
try {
    $results = shell_exec("python3
        ../../py/multilabel_dt_crossvalidation.py
        $file_path $featuresImplode $labelsImplode
        $max_depth $min_samples_leaf $kFoldsInt $classifier");
} catch(Exception $e) {
    header("HTTP/1.1 400 Bad Request");
    print json_encode(['errormsg'=>"An error has occurred
        while trying to run the Python module for Cross-Validation.
        <br><br> Please check the possibility of missing values
        existence in given columns and try again."]);
    exit;
}

if(!$results || $results == null) {
    header("HTTP/1.1 400 Bad Request");
    print json_encode(['errormsg'=>"An error has occurred
        while trying to run the Python module for Cross-Validation.
        <br><br> Please check the possibility of missing values
        existence in given columns and try again."]);
    exit;
}

print($results);

```

4.5.3 Χειρισμός Multi-label Μοντέλων

Ο χειρισμός των multi-label μοντέλων πραγματοποιείται στην σελίδα «Build a Model» της εφαρμογής και περιλαμβάνει την δημιουργία, την αξιολόγηση και την αποθήκευση μοντέλων. Αρχικά, για την δημιουργία ενός multi-label μοντέλου από τον χρήστη, απαιτείτε αναγκαία η επιλογή και εισαγωγή των κατάλληλων παραμέτρων. Αυτές περιλαμβάνουν τα χαρακτηριστικά (features) και τις ετικέτες (labels) του συνόλου δεδομένων, τον ταξινομητή μετασχηματισμού προβλήματος (transformation classifier), καθώς και τις τιμές για το μέγιστο βάθος (max depth), τον ελάχιστο αριθμό δειγμάτων ανά φύλλο (min samples leaf) και το k. Η αυτόματη λειτουργία (Auto Mode) μπορεί να ενεργοποιηθεί, από τον εγγεγραμμένο χρήστη, για τις παραμέτρους 'transformation classifier', 'max depth' και 'min samples leaf', καθώς αυτές θεωρούνται οι πιο σημαντικές για την εφαρμογή.

Εφόσον ο χρήστης ολοκληρώσει την επιλογή των απαιτούμενων παραμέτρων, μπορεί να πατήσει το κουμπί 'Build Model' για να ξεκινήσει η διαδικασία δημιουργίας του μοντέλου. Μετά την ολοκλήρωσή της, εμφανίζονται τα αποτελέσματα των μετρικών αξιολόγησης του μοντέλου. Η διαδικασία αυτή υλοποιείται μέσω του multilabel_dt_crossvalidation.py αρχείου, το οποίο εκτελείται μέσω της php. Παρακάτω παρουσιάζονται μερικά σημαντικά αποσπάσματα κώδικα από το συγκεκριμένο αρχείο, περιλαμβάνοντας την αρχική δήλωση των multi-label ταξινομητών μετασχηματισμού προβλήματος, την τελική εκτέλεση της μεθόδου k-fold cross-validation με τον υπολογισμό των μετρικών αξιολόγησης, καθώς και τη λειτουργικότητα AutoML για τον αυτόματο υπολογισμό των καλύτερων τιμών παραμέτρων με βάση το μικρότερο 'hamming loss'.

Ο παρακάτω κώδικας παρουσιάζει τη διαδικασία δήλωσης των multi-label ταξινομητών μετασχηματισμού προβλήματος, μαζί με τις σχετικές παραμέτρους τους. Όπως έχει αναφερθεί προηγουμένως, η εφαρμογή μας υποστηρίζει τη χρήση των 3 δημοφιλών αλγορίθμων: Binary Relevance, Label Powerset και Classifier Chain. Έτσι, αρχικοποιούνται σε ένα dictionary με όνομα classifiers.

Δήλωση Multi-label Ταξινομητών Μετασχηματισμού Προβλήματος:

```
# Defining the multilabel classifiers.
classifiers = {
    'BinaryRelevance': BinaryRelevance(
        classifier=DecisionTreeClassifier(), # Initializing params.
        require_dense=[False, True] # X: dense, y: sparse.
    ),
    'LabelPowerset': LabelPowerset(
        classifier=DecisionTreeClassifier(), # Initializing params.
        require_dense=[False, True] # X: dense, y: sparse.
    ),
    'ClassifierChain': ClassifierChain(
        classifier=DecisionTreeClassifier(), # Initializing params.
        require_dense=[False, True] # X: dense, y: sparse.
    )
}
```

Ο επόμενος κώδικας περιγράφει αναλυτικά τη διαδικασία του k-fold cross-validation, ενώ παράλληλα υπολογίζει τις απαιτούμενες μετρικές αξιολόγησης για το μοντέλο. Όπως είναι αντιληπτό, υπολογίζονται οι μετρικές απόδοσης 'hamming loss', 'accuracy', 'precision', 'recall' και 'f-score'. Αυτές περιλαμβάνουν τόσο την απόδοση για κάθε μεμονωμένη ετικέτα όσο και τη συνολική απόδοση του μοντέλου. Εξαιρέση αποτελούν το 'hamming loss' και 'accuracy' που, εξαιτίας της φύσης τους, υπολογίζονται μόνο για το σύνολο του μοντέλου. Οι μετρικές που αντιστοιχούν σε κάθε ετικέτα αποθηκεύονται σε δομές δεδομένων τύπου λίστες (lists) από λίστες (nested lists), όπου ο αριθμός τους αντιστοιχεί στον αριθμό των ετικετών. Αντίθετα, οι μετρικές που αφορούν τη συνολική απόδοση του μοντέλου αποθηκεύονται σε απλές λίστες, καθώς δεν εξαρτώνται από τον αριθμό των ετικετών. Ο υπολογισμός των μετρικών πραγματοποιείται εντός της μεθόδου cross-validation, όπως φαίνεται και στον παρακάτω κώδικα.

Εκτέλεση k-fold cross-validation και Υπολογισμός Μετρικών Αξιολόγησης:

```
# Initialize list of lists to store metrics for each class label.
arr_pre = [[] for _ in range(len(selectedLabels))]
arr_rec = [[] for _ in range(len(selectedLabels))]
arr_fsc = [[] for _ in range(len(selectedLabels))]

pre_per_label = [[] for _ in range(len(selectedLabels))]
rec_per_label = [[] for _ in range(len(selectedLabels))]
fsc_per_label = [[] for _ in range(len(selectedLabels))]

# Lists to store metrics for each fold.
hamming_losses = []
accuracy_scores = []
precision_scores = []
recall_scores = []
f_scores = []

# Performing k-fold cross-validation.
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 = classLabels.iloc[train_index],
                    classLabels.iloc[test_index]

    classifier.fit(X_train, y_train)
    predictions = classifier.predict(X_test)

    # Converting predictions to dense format.
    pred = classifier.predict(X_test).toarray()

    # Calculating precision, recall, and f-score for each label.
    for i, label in enumerate(selectedLabels):
```

```

unique_values = y_test[label].unique()
pre, rec, fsc, _ = metrics.precision_recall_fscore_support (
    y_test[label], pred[:, i], average=None,
    zero_division=0.0, labels = unique_values)
arr_pre[i].append(pre)
arr_rec[i].append(rec)
arr_fsc[i].append(fsc)

# Calculating avg evaluation metrics.
hamming_loss = metrics.hamming_loss(y_test, predictions)
accuracy = metrics.accuracy_score(y_test, predictions)
precision, recall, fscore, _ = metrics.
    precision_recall_fscore_support(y_test, predictions,
    average='macro', zero_division=0.0)

hamming_losses.append(hamming_loss)
accuracy_scores.append(accuracy)
precision_scores.append(precision)
recall_scores.append(recall)
f_scores.append(fscore)

```

Ο παρακάτω κώδικας χειρίζεται τις παραμέτρους που όρισε ο χρήστης για το μοντέλο του, ο οποίος είτε επέλεξε συγκεκριμένες τιμές είτε ενεργοποίησε τη λειτουργία Auto για ορισμένες από αυτές. Αναλύοντας τη λογική του κώδικα, παρατηρούμε ότι λαμβάνονται υπόψη όλες οι πιθανές περιπτώσεις παραμέτρων που όρισε ο χρήστης κατά την διαδικασία δημιουργίας του μοντέλου. Συγκεκριμένα, οι περιπτώσεις μπορεί να είναι οι εξής:

1. Όλες οι παράμετροι έχουν οριστεί σε λειτουργία Auto.
2. Όλες οι παράμετροι έχουν συγκεκριμένες τιμές.
3. Μερικές παράμετροι έχουν οριστεί σε λειτουργία Auto, ενώ άλλες έχουν συγκεκριμένες τιμές.

Τα αποτελέσματα των παραμέτρων υπολογίζονται σειριακά για την αποφυγή πιθανών σφαλμάτων και προκύπτουν μέσω της μεθόδου cross-validation. Αφού ολοκληρωθεί ο υπολογισμός, δημιουργείται ο τελικός ταξινομητής Decision Tree (DTree), στον οποίο εισάγονται οι καλύτερες τιμές των παραμέτρων 'max depth' και 'min samples leaf' που υπολογίστηκαν από την αυτοματοποιημένη διαδικασία. Ο νέος ταξινομητής DTree ενσωματώνεται ως ταξινομητής βάσης (base classifier) στον ταξινομητή μετασχηματισμού προβλήματος (transformation classifier), παράγοντας έτσι το τελικό μοντέλο προς χρήση.

Υπολογισμός Παραμέτρων με AutoML:

```

# Auto classifier selection.
if sys.argv[7] == 'Auto':
    best_classifier = None

```

```

best_hamming_loss = float('inf')

for name, clf in classifiers.items():
    hamming_losses = []
    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 = classLabels.iloc[train_index],
            classLabels.iloc[test_index]

        clf.fit(X_train, y_train)
        predictions = clf.predict(X_test)

        hamming_loss = metrics.hamming_loss(y_test,
            predictions)
        hamming_losses.append(hamming_loss)

    avg_hamming_loss = sum(hamming_losses) / k

    if avg_hamming_loss < best_hamming_loss:
        best_hamming_loss = avg_hamming_loss
        best_classifier = clf
        best_classifier_name = name

classifier = best_classifier # The best classifier.

# Auto min_samples_leaf selection.
if min_samples_leaf == 'Auto':
    min_samples_leaf, best_hamming_loss = auto_select_param(
        'min_samples_leaf', range(1, 50, 3), attr,
        classLabels, kf, classifier, k)
else: # Specific min_samples_leaf selection.
    min_samples_leaf = int(min_samples_leaf)

# Auto max_depth selection.
if max_depth == 'Auto':
    max_depth, best_hamming_loss = auto_select_param(
        'max_depth', range(1, 50, 3), attr, classLabels,
        kf, classifier, k)
elif max_depth == 'None': # None max_depth selection.
    max_depth = None
else: # Specific max_depth selection.
    max_depth = int(max_depth)

classifier.classifier = DecisionTreeClassifier(

```

```

        max_depth=max_depth,
        min_samples_leaf=min_samples_leaf) # Final DTree parameters.
else: # Specific classifier selection.
    classifier = classifiers.get(sys.argv[7])
    best_classifier_name = sys.argv[7]

# Unkown classifier message interrupt.
if classifier is None:
    print(f"\nError: Unknown classifier type: {sys.argv[7]}")
    sys.exit(1)

# Auto min_samples_leaf selection.
if min_samples_leaf == 'Auto':
    min_samples_leaf, best_hamming_loss = auto_select_param(
        'min_samples_leaf', range(1, 50, 3), attr,
        classLabels, kf, classifier, k)
else: # Specific min_samples_leaf selection.
    min_samples_leaf = int(min_samples_leaf)

# Auto max_depth selection.
if max_depth == 'Auto':
    max_depth, best_hamming_loss = auto_select_param(
        'max_depth', range(1, 50, 3), attr,
        classLabels, kf, classifier, k)
elif max_depth == 'None': # None max_depth selection.
    max_depth = None
else: # Specific max_depth selection.
    max_depth = int(max_depth)

classifier.classifier = DecisionTreeClassifier(
    max_depth=max_depth,
    min_samples_leaf=min_samples_leaf) # Final DTree parameters.

```

Μετά την επιτυχή δημιουργία και αξιολόγηση του multi-label μοντέλου, παρέχεται η δυνατότητα στον χρήστη να αποθηκεύσει το μοντέλο για μελλοντική χρήση. Για την αποθήκευση, ο χρήστης καλείται να ορίσει ένα όνομα για το μοντέλο του και έπειτα να πατήσει το κουμπί 'Save Model' για να ολοκληρωθεί η διαδικασία. Η λειτουργία αποθήκευσης πραγματοποιείται στο αρχείο 'save_multilabel_model.py', το οποίο εκτελείται μέσω της php. Παρακάτω παρατίθεται ένα απόσπασμα κώδικα από το συγκεκριμένο αρχείο, που δείχνει την αποθήκευση του μοντέλου με χρήση της βιβλιοθήκης 'joblib' της python. Μέσω αυτής, το μοντέλο και ο μετασχηματισμός του μετατρέπονται σε αρχεία με κατάληξη .pkl και αποθηκεύονται στον προσωπικό φάκελο του χρήστη.

Αποθήκευση Multi-label Μοντέλου:

```
# Defining the multilabel classifiers.
classifiers = {
    'BinaryRelevance': BinaryRelevance(
        classifier=DecisionTreeClassifier(max_depth=max_depth,
            min_samples_leaf=min_samples_leaf),
        require_dense=[False, True] # X: dense, y: sparse.
    ),
    'LabelPowerset': LabelPowerset(
        classifier=DecisionTreeClassifier(max_depth=max_depth,
            min_samples_leaf=min_samples_leaf),
        require_dense=[False, True] # X: dense, y: sparse.
    ),
    'ClassifierChain': ClassifierChain(
        classifier=DecisionTreeClassifier(max_depth=max_depth,
            min_samples_leaf=min_samples_leaf),
        require_dense=[False, True] # X: dense, y: sparse.
    )
}

classifier = classifiers.get(sys.argv[8]) # selected classifier.

classifier.fit(attr, classLabels)
joblib.dump(classifier, model_trf)
classifier.classifier.fit(attr, classLabels)
joblib.dump(classifier, model_path)

print(json.dumps({"message": "Model successfully saved."}))
```

4.5.4 Χειρισμός Προεκπαιδευμένων Μοντέλων

Ο χειρισμός των προεκπαιδευμένων μοντέλων πραγματοποιείται στην σελίδα «Pretrained Models» της εφαρμογής και περιλαμβάνει την επιστροφή του περιεχομένου ενός μοντέλου, την οπτικοποίηση των δέντρων απόφασής του, καθώς και την κατηγοριοποίηση νέων στιγμιτύπων ενός αταξινομήτου συνόλου δεδομένων. Όσον αφορά την επισκόπηση του περιεχομένου ενός μοντέλου, όταν ο χρήστης επιλέξει το επιθυμητό προεκπαιδευμένο μοντέλο, εμφανίζονται συγκεκριμένες παράμετροι για προβολή. Το περιεχόμενο που επιστρέφεται περιλαμβάνει τα χαρακτηριστικά 'features', τον τύπο του ταξινομητή μετασχηματισμού προβλήματος 'classifier type', καθώς και τις τιμές για το 'max depth' και το 'min samples leaf' του δέντρου απόφασης. Με αυτόν τον τρόπο, ο χρήστης μπορεί να έχει πλήρη εικόνα των παραμέτρων του προεκπαιδευμένου μοντέλου που επέλεξε.

Ο κώδικας που ακολουθεί, περιλαμβάνει την λειτουργία επιστροφής του περιεχομένου ενός προεκπαιδευμένου μοντέλου, η οποία υλοποιείται στο 'get_multilabelModel_content.py'

αρχείο. Συγκεκριμένα, στην αρχή διαβάζεται το path όπου είναι αποθηκευμένο το '.pkl' αρχείο (model_file), και στη συνέχεια γίνεται χρήση της μεταβλητής 'joblib_model' όπου μέσω αυτής μετατρέπεται το αρχείο ξανά σε ένα multi-label μοντέλο, με τον μετασχηματισμό προβλήματος που ο χρήστης είχε επιλέξει. Έπειτα, κάνοντας χρήση των σχετικών δυνατοτήτων που προσφέρει η βιβλιοθήκη αυτή, επιστρέφονται τα 'features' (columns), ο τύπος του ταξινομητή μετασχηματισμού προβλήματος (classifier_type), το μέγιστο βάθος (max_depth), καθώς και ο ελάχιστος αριθμός δειγμάτων ανά φύλο (min_samples_leaf) του μοντέλου.

Επιστροφή Περιεχομένου ενός Προεκπαιδευμένου Μοντέλου:

```
model_file = sys.argv[1]
joblib_model = joblib.load(model_file)

# Returning feature names (columns).
columns = joblib_model.classifier.feature_names_in_.tolist()

# Returning classifier type.
classifier_type = joblib_model.__class__.__name__

# Returning max_depth.
max_depth = joblib_model.classifier.max_depth

# Returning min_samples_leaf.
min_samples_leaf = joblib_model.classifier.min_samples_leaf

print(json.dumps({
    "columns": columns,
    "classifier_type": classifier_type,
    "max_depth": max_depth,
    "min_samples_leaf": min_samples_leaf
}))
```

Ο παρακάτω κώδικας, περιγράφει τη διαδικασία οπτικοποίησης των δέντρων απόφασης ενός προεκπαιδευμένου μοντέλου που έχει επιλέξει ο χρήστης. Η λειτουργία αυτή υλοποιείται στο αρχείο 'visualize_ml_dtrees.py'. Ειδικότερα, δημιουργήθηκε μια match-case συνθήκη, όπου με βάση τον τύπο του ταξινομητή (classifier_type) εκτελείται η αντίστοιχη λειτουργία των δέντρων απόφασης. Ο τύπος του ταξινομητή μπορεί να έχει τρεις string τιμές: 'BinaryRelevance', 'LabelPowerset' ή 'ClassifierChain', ανάλογα τον μετασχηματισμό προβλήματος του μοντέλου. Επιπλέον, για τη δημιουργία και εξαγωγή των δέντρων απόφασης, γίνεται χρήση της βιβλιοθήκης 'tree' και της μεθόδου 'export_graphviz()' από την 'sklearn'. Τέλος, σημαντικό ρόλο παίζουν και οι παράμετροι 'features' και 'labels' των δέντρων, οι οποίες συμβάλλουν στον καθορισμό των αποτελεσμάτων τους. Για την περίπτωση του 'LabelPowerset' κατασκευάζεται ένα ενιαίο δέντρο απόφασης, το οποίο λαμβάνει υπόψη όλους τους πιθανούς συνδυασμούς ετικετών (labels) ως ξεχωριστές κλάσεις. Αντίθετα, στο 'BinaryRelevance' και 'ClassifierChain', δημιουργείται ένα ξεχωριστό δυαδικό δέντρο απόφασης για κάθε ετι-

κέτα, όπου το καθένα περιλαμβάνει τα χαρακτηριστικά και τις τιμές που σχετίζονται με την παρουσία (1) ή την απουσία (0) της συγκεκριμένης ετικέτας σε κάθε παρατήρηση. Στην περίπτωση του 'ClassifierChain', η διαδικασία ενισχύεται περαιτέρω, καθώς κάθε δέντρο λαμβάνει υπόψη του και τις προβλέψεις των προηγούμενων δέντρων στην αλυσίδα.

Οπτικοποίηση Δέντρων Απόφασης:

```
# Creating the DTrees based on classifier type selection.
match classifier_type:
    case 'BinaryRelevance':
        # Converting label values from int to strings
        # for DTree plot representation.
        labels_strings = [[str(item) for item in sublist]
                           for sublist in labels]

        for i, n_tree in enumerate(joblib_model_trf.classifiers_):
            tree_path2 = f"{tree_path}_{i + 1}.dot"
            dot_data = tree.export_graphviz(n_tree,
                                             out_file = tree_path2, feature_names = features,
                                             class_names = labels_strings[i], filled = True,
                                             node_ids = True, rounded = True,
                                             precision = 2, max_depth = 10)

    case 'LabelPowerset':
        # Converting label's unique combinations to strings
        # for DTree plot representation.
        label_combinations = [str(combination) for combination in
                               joblib_model_trf.unique_combinations_]

        # Label Powerset creates only one DTree.
        n_tree = joblib_model_trf.classifier

        tree_path2 = tree_path + ".dot"
        dot_data = tree.export_graphviz(n_tree,
                                         out_file = tree_path2, feature_names = features,
                                         class_names = label_combinations, filled = True,
                                         node_ids = True, rounded = True, precision = 2,
                                         max_depth = 10)

    case 'ClassifierChain':
        # Converting label values from int to strings
        # for DTree plot representation.
        labels_strings = [[str(item) for item in sublist]
                           for sublist in labels]
```

```

for i, n_tree in enumerate(joblib_model_trf.classifiers_):
    # Start with the original feature names.
    feature_names = list(features)

    # Extend feature names to include previous
    # DTree results.
    # Using `i > 0` to skip adding anything
    # for the first DTree.
    if i > 0:
        feature_names += [features[j] for j in range(i)]

    tree_path2 = f"{tree_path}_{i + 1}.dot"
    dot_data = tree.export_graphviz(n_tree,
        out_file = tree_path2, feature_names = feature_names,
        class_names = labels_strings[i], filled = True,
        node_ids = True, rounded = True, precision = 2,
        max_depth = 10)

    print(json.dumps({
        "message": "Dot data for DTrees created successfully."
    }))

```

Τέλος, όσον αφορά την κατηγοριοποίηση των νέων στιγμιοτύπων ενός αταξινόμητου συνόλου δεδομένων (unclassified dataset), ο χρήστης θα πρέπει αρχικά να έχει επιλέξει το επιθυμητό προεκπαιδευμένο μοντέλο, με βάση το οποίο θα πραγματοποιηθεί η κατηγοριοποίηση. Για να καταστεί δυνατή η εκτέλεση αυτής της διαδικασίας, το αταξινόμητο dataset θα πρέπει να είναι όμοιο με αυτό του μοντέλου, δηλαδή τα 'features' του μοντέλου να υπάρχουν και στο dataset. Όπως φαίνεται και στον ακόλουθο κώδικα του αρχείου 'classifyMultilabelData.py', χρησιμοποιείται η μέθοδος 'predict()' στο μοντέλο, μέσω της οποίας πραγματοποιούνται οι προβλέψεις. Στη συνέχεια, εφόσον οι προβλέψεις έχουν περάσει από την μέθοδο cross-validation, προστίθενται στο DataFrame ως νέες στήλες, με ονόματα 'predictions_L1', 'predictions_L2' κ.λπ., ανάλογα με τον αριθμό των ετικετών που προβλέφθηκαν. Έπειτα, δημιουργείται μια νέα λίστα στηλών που περιλαμβάνει τόσο τα αρχικά χαρακτηριστικά όσο και τις νέες στήλες προβλέψεων, δημιουργώντας έτσι το τελικό κατηγοριοποιημένο σύνολο δεδομένων. Αυτό το σύνολο αποθηκεύεται σε ένα αρχείο .csv για μελλοντική χρήση, μέσω της μεθόδου 'to_csv()'.

Κατηγοριοποίηση Νέων Στιγμιοτύπων:

```

predictions = model.predict(attr)

... (cross-validation method) ...

# Iterating through each column and add it
# to the DataFrame.
for i in range(pred.shape[1]):

```

```

column_name = f'predictions_L{i + 1}'
dataset[column_name] = pred[:, i]

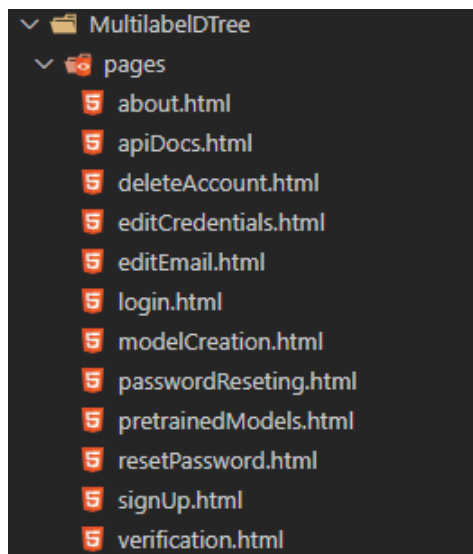
# Updating the selectedFeatures list
# to include the new columns.
cols = selectedFeatures + [f'predictions_L{i + 1}' for
    i in range(pred.shape[1])]
dataset[cols].to_csv(save_path, index = False,
    encoding = 'utf-8')

```

4.6 Υλοποίηση του Frontend

Ένα εξίσου σημαντικό μέρος του «MultilabelDTree» είναι το Frontend, το οποίο επιτρέπει τη χρήση όλων των λειτουργιών που παρέχει η εφαρμογή μέσω μίας εύχρηστης γραφικής διεπαφής. Το Frontend αξιοποιεί την επικοινωνία του με το Web API, χρησιμοποιώντας γλώσσες και τεχνολογίες που του δίνουν αυτή την δυνατότητα. Όπως είναι γνωστό, η εφαρμογή μας αξιοποιεί βασικές τεχνολογίες του web, συνδυάζοντας τις δυνατότητες της HTML, της CSS αλλά και του framework Bootstrap για την δομή, το σχεδιασμό και τη εμφάνιση. Επιπλέον, γίνεται χρήση της γλώσσας προγραμματισμού JavaScript σε συνεργασία με τη βιβλιοθήκη JQuery, για να ενισχυθεί η διαδραστικότητα και η δυναμικότητα της εφαρμογής.

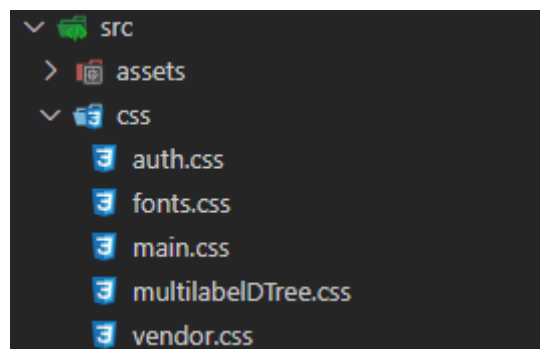
Συγκεκριμένα, με τη χρήση της γλώσσας HTML δημιουργείτε η δομή όλων των σελίδων της εφαρμογής, καθώς και η οργάνωση του περιεχομένου τους. Στο Σχήμα 4.4, απεικονίζονται τα HTML αρχεία (συνολικά 12) που αντιπροσωπεύουν τις σελίδες (pages) της εφαρμογής. Τα αρχεία αυτά, βρίσκονται στον φάκελο «pages», ενώ το αρχείο 'index.html' βρίσκεται στον κεντρικό κατάλογο, καθώς αντιπροσωπεύει την αρχική σελίδα της εφαρμογής, η οποία φορτώνεται αυτόματα όταν ο χρήστης επισκέπτεται τον σχετικό σύνδεσμο.



Σχήμα 4.4: Αρχεία HTML Κώδικα για τις Σελίδες της Εφαρμογής

Αναλυτικότερα, το αρχείο `'about.html'` σχεδιάστηκε για να παρουσιάζει την δομή της σελίδας πληροφόρησης, η οποία παρέχει τον σκοπό δημιουργίας της εφαρμογής και βασικές πληροφορίες σχετικά με αυτήν. Η σελίδα αυτή είναι προσβάσιμη από όλους τους χρήστες, είτε είναι εγγεγραμμένοι είτε όχι. Το αρχείο `'apiDocs.html'` περιλαμβάνει λεπτομερείς πληροφορίες για το Web API της εφαρμογής, καθώς και για τη βάση δεδομένων που δημιουργήθηκε. Συνεπώς, η πρόσβαση σε αυτή τη σελίδα είναι αποκλειστικά για τους εγγεγραμμένους χρήστες και προγραμματιστές. Επιπλέον, τα αρχεία `'signUp.html'`, `'login.html'` και `'verification.html'` δημιουργήθηκαν για τη διαχείριση της εγγραφής χρηστών, την επιβεβαίωση των λογαριασμών τους μετά την εγγραφή, καθώς και την είσοδό τους στην εφαρμογή μέσα από τις σχετικές φόρμες που προσφέρουν. Επιπρόσθετα, οι σελίδες `'editCredentials.html'`, `'editEmail.html'` και `'deleteAccount.html'`, που είναι διαθέσιμες μέσω του μενού χρήστη, δίνουν τη δυνατότητα στους εγγεγραμμένους χρήστες να επεξεργάζονται τα βασικά στοιχεία του λογαριασμού τους ή να τον διαγράψουν, εφόσον το επιθυμούν. Οι σελίδες `'passwordResetting.html'` και `'resetPassword.html'` εξυπηρετούν περιπτώσεις όπου ένας χρήστης έχει ξεχάσει τον κωδικό του. Η σελίδα `'resetPassword.html'` παρέχει τη φόρμα για την εισαγωγή του email του χρήστη, στο οποίο θα του αποσταλεί ο σύνδεσμος επαναφοράς κωδικού, ενώ η σελίδα με το αρχείο `'passwordResetting.html'` παρέχει τη φόρμα για την εισαγωγή του νέου κωδικού του χρήστη, η οποία είναι μόνο προσβάσιμη μέσω του συνδέσμου που του έχει αποσταλεί στο email. Τέλος, οι βασικές σελίδες `'modelCreation.html'` και `'pretrainedModels.html'`, δημιουργήθηκαν για την υποστήριξη των κύριων λειτουργιών της εφαρμογής, όπου η πρώτη αφορά τη δημιουργία και αποθήκευση των multi-label μοντέλων, ενώ η δεύτερη επιτρέπει την προβολή των προεκπαιδευμένων μοντέλων και την κατηγοριοποίηση νέων στιγμιοτύπων από μη ταξινομημένα σύνολα δεδομένων.

Με τη χρήση της γλώσσας CSS, καθορίστηκε η εμφάνιση του περιεχομένου στις σελίδες, μέσω της προσθήκης των κατάλληλων στυλ κανόνων στα στοιχεία της HTML. Όπως φαίνεται και στο Σχήμα 4.5, που ακολουθεί, τα αρχεία CSS είναι αποθηκευμένα στον κατάλογο `«src/css»` της εφαρμογής. Συνολικά, δημιουργήθηκαν πέντε βασικά αρχεία `.css`, η δομή των οποίων επαναχρησιμοποιείται σε περισσότερες από μία σελίδες της εφαρμογής.

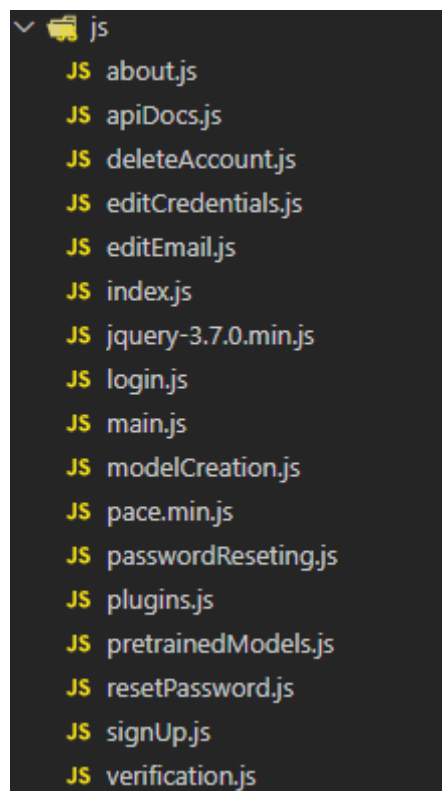


Σχήμα 4.5: Αρχεία CSS Κώδικα για τις Σελίδες της Εφαρμογής

Το αρχείο `'auth.css'` χρησιμοποιείται σε όλες τις σελίδες που αφορούν την αυθεντικοποίηση χρηστών, όπως για παράδειγμα στις σελίδες εγγραφής, εισόδου στην εφαρμογή κ.ά. Επιπλέον, το αρχείο `'fonts.css'`, περιλαμβάνει ορισμένα εξωτερικά φόντα γραμματοσειρών που χρησιμο-

ποιήθηκαν στην εφαρμογή και συνέβαλαν θετικά στην αισθητική της. Το αρχείο `'vendor.css'` περιέχει έτοιμα κομμάτια CSS κώδικα για την εμφάνιση ορισμένων λειτουργιών της εφαρμογής, όπως του «preloader», του «Smooth Scroll Down» και του «Back to Top». Το αρχείο `'main.css'` καθορίζει την εμφάνιση των κύριων σελίδων της εφαρμογής, όπως της αρχική σελίδα και της σελίδας «About». Τέλος, το αρχείο `'multilabelDTree.css'` διαχειρίζεται την εμφάνιση των βασικών σελίδων της εφαρμογής, δηλαδή των σελίδων «Build a Model» και «Pretrained Models».

Με τη χρήση της γλώσσας προγραμματισμού JavaScript και της βιβλιοθήκης JQuery, υλοποιήθηκε η λειτουργικότητα της εφαρμογής από την πλευρά του πελάτη (client-side). Η JQuery απλοποίησε τη διαδραστικότητα του χρήστη με την εφαρμογή, ενώ μέσω της τεχνικής AJAX (Asynchronous JavaScript and XML) έγινε δυνατή η ζωντανή αλλαγή του περιεχομένου και της δομής της εφαρμογής, χωρίς την ανάγκη συνεχούς επαναφόρτωσης των σελίδων. Οι κλήσεις προς τα endpoints του Web API της εφαρμογής πραγματοποιούνται με τη χρήση της μεθόδου `'$.ajax()'` της JQuery, καθώς προσφέρει έναν εύκολο τρόπο για την υλοποίηση αιτήσεων AJAX. Στο Σχήμα 4.6, που ακολουθεί, παρουσιάζονται τα 17 αρχεία JavaScript που αναπτύχθηκαν για τη λειτουργικότητα της εφαρμογής. Ακολουθεί, επίσης, ένα παράδειγμα κώδικα με τη χρήση της μεθόδου `'$.ajax()'` για την καλύτερη κατανόηση της όλης διαδικασίας.



Σχήμα 4.6: Αρχεία JavaScript Κώδικα της Εφαρμογής

Ειδικότερα, η ανάπτυξη της εφαρμογής βασίστηκε σε ένα οργανωμένο σύστημα JavaScript αρχείων, όπου κάθε σελίδα διαθέτει το δικό της script για την υποστήριξη της αντίστοιχης λειτουργικότητας. Έτσι, για τις κύριες σελίδες της εφαρμογής, δηλαδή την αρχική και το «About», δημιουργήθηκαν τα αρχεία `'index.js'` και `'about.js'`. Η σελίδα «apiDocs», που περιγράφει το Web

API της εφαρμογής, υποστηρίζεται από το αρχείο 'apiDocs.js'. Για τη σελίδα εγγραφής του χρήστη και τη σελίδα εισόδου του στην εφαρμογή, δημιουργήθηκαν τα αρχεία 'signUp.js' και 'login.js' αντίστοιχα. Επίσης, για τις σελίδες διαχείρισης των πληροφοριών και του λογαριασμού του χρήστη, όπως η επεξεργασία στοιχείων, η αλλαγή email, η διαγραφή λογαριασμού και η επαναφορά κωδικού, αναπτύχθηκαν τα αρχεία 'editCredentials.js', 'editEmail.js', 'deleteAccount.js', 'resetPassword.js' και 'passwordReseting.js'. Οι σελίδες επιβεβαίωσης υποστηρίζονται από το σχετικό αρχείο, με όνομα 'verification.js'. Σημαντικό αποτελεί το αρχείο 'main.js', το οποίο κρατάει τον πυρήνα μερικών βασικών λειτουργιών του frontend, που σχετίζονται κυρίως με την αρχική σελίδα. Επιπλέον, στα script αρχεία 'modelCreation.js' και 'pretrainedModels.js', αναπτύχθηκε η λογική και λειτουργικότητα των βασικών σελίδων «Build a Model» και «Pretrained Models» της εφαρμογής. Παράλληλα, το αρχείο 'pace.min.js', από τη βιβλιοθήκη PACE JS, χρησιμοποιείται για τη λειτουργικότητα του «preloader» που χρησιμοποιήθηκε στην εφαρμογή μας, ενώ το 'plugins.js' παρέχει επιπρόσθετα έτοιμα features που χρησιμοποιούνται σε διάφορες σελίδες της. Τέλος, για την αξιοποίηση της βιβλιοθήκης JQuery, απαιτείται η εισαγωγή του 'jquery-3.7.0.min.js' αρχείου, το οποίο περιέχει τη σχετική λειτουργικότητα με την κατάλληλη έκδοση.

Κώδικας JavaScript - JQuery - AJAX για την κλήση Endpoint του Web API:

```
$.ajax({
  url: '../server/php/api/signUp.php',
  method: 'POST',
  data: JSON.stringify({
    fname: fname, // User's First Name.
    lname: lname, // User's Last Name.
    email: email, // User's Email.
    pass: pass, // User's Password.
    pass_confirm: pass_confirm // User's Confirmed Password.
  }),
  dataType: "json",
  contentType: 'application/json',
  success: function () {
    alert_success("You successfully signed up.");
    alert_warning("Check your inbox to verify account.");
    $('#loadingbtn').hide();
    $("#email_resend").show();
  },
  error: function (xhr, status, error) {
    var response = JSON.parse(xhr.responseText);
    var errormes = response.errormesg;
    alert_danger(errormes);
    $('#loadingbtn').hide();
    if (errormes == "Mailer Error. Message could not be sent.") {
      $("#email_resend").show();
    } else {
```

```
        $('#regbtn').show();  
    }  
}  
});
```

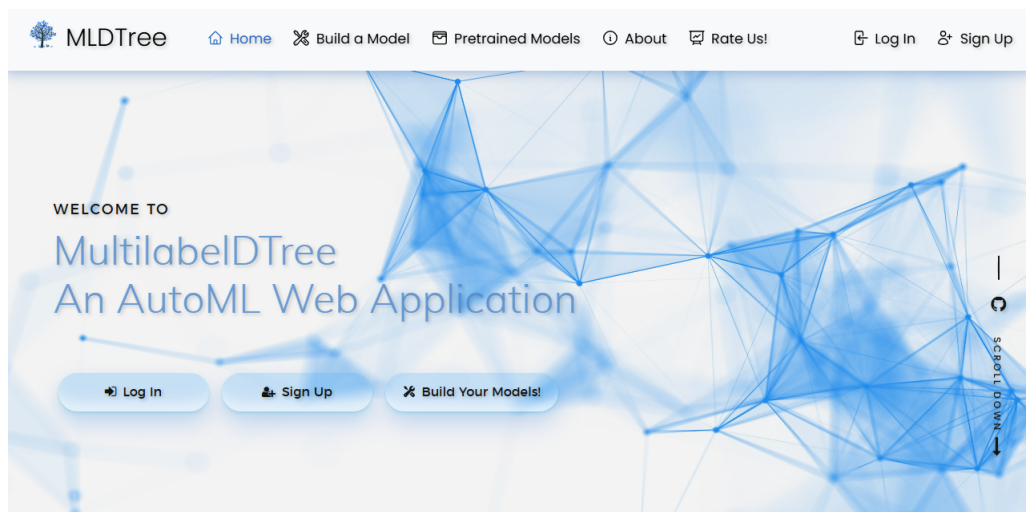
Ο παραπάνω κώδικας, χρησιμοποιεί τη μέθοδο `$.ajax()` της JQuery για να στείλει τα δεδομένα εγγραφής χρήστη (όνομα, επώνυμο, email και κωδικό πρόσβασης) στο `signUp.php` αρχείο του server. Τα δεδομένα αποστέλλονται σε μορφή JSON, χρησιμοποιώντας την μέθοδο POST. Αν η εγγραφή ολοκληρωθεί επιτυχώς, εμφανίζονται μηνύματα επιτυχίας και προειδοποίησης για την επιβεβαίωση λογαριασμού του χρήστη, μέσω email. Σε περίπτωση σφάλματος, εμφανίζεται το αντίστοιχο μήνυμα και ανάλογα με το είδος του σφάλματος, είτε επιτρέπεται στον χρήστη να επαναστείλει το email επιβεβαίωσης λογαριασμού είτε εμφανίζεται το κουμπί για νέα προσπάθεια εγγραφής. Μετά, την επιτυχή ολοκλήρωση αυτών των διαδικασιών, τα δεδομένα λογαριασμού του χρήστη αποθηκεύονται στην βάση δεδομένων MySQL της εφαρμογής.

Κεφάλαιο 5

Παρουσίαση του MultilabelDTree

5.1 Αρχική Σελίδα της Εφαρμογής

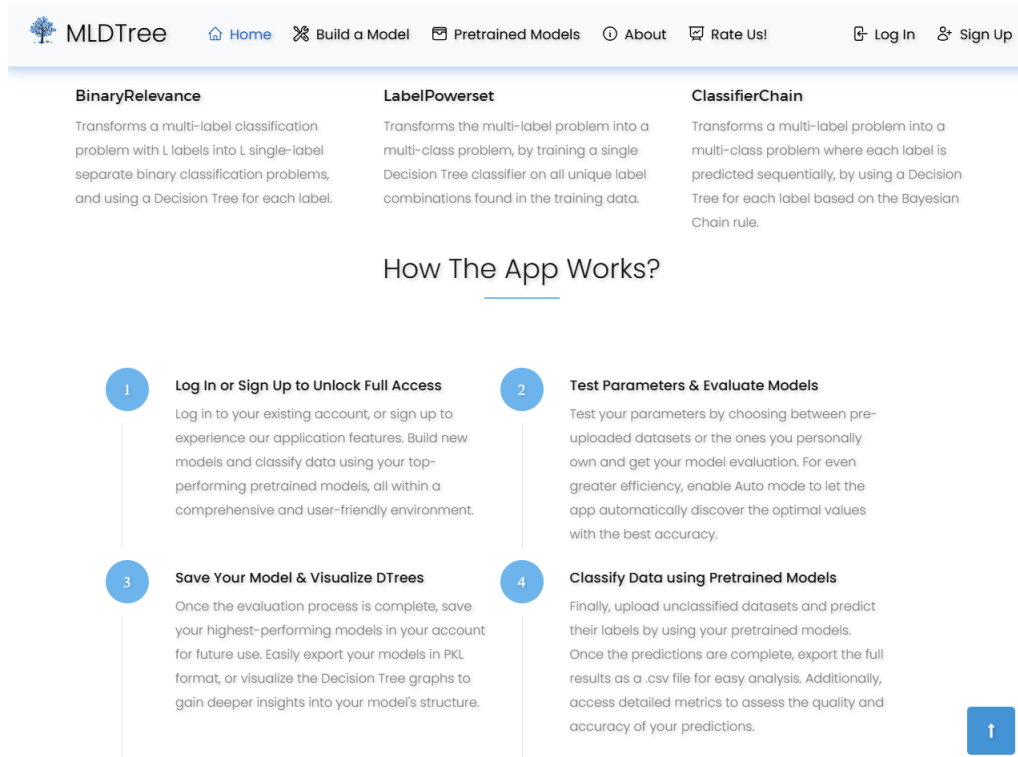
Ανοίγοντας την εφαρμογή, ο χρήστης μεταφέρεται στην αρχική σελίδα. Όπως φαίνεται στο Σχήμα 5.1, αρχικά εμφανίζεται ένα μήνυμα καλωσορίσματος προς τον χρήστη. Στο επάνω μέρος της σελίδας βρίσκεται το μενού πλοήγησης, μέσω του οποίου ο χρήστης μπορεί να εξερευνήσει τις διάφορες σελίδες της εφαρμογής αλλά και να πραγματοποιήσει είτε εγγραφή είτε είσοδο με τον προσωπικό του λογαριασμό, ώστε να αποκτήσει πρόσβαση στις δυνατότητές της. Στο κάτω μέρος της αρχικής σελίδας παρέχεται μια σύντομη περιγραφή σχετικά με την εφαρμογή, ενώ στη συνέχεια παρουσιάζονται οι βασικές λειτουργίες της, μέσω αναλυτικών βημάτων (Σχήμα 5.2).



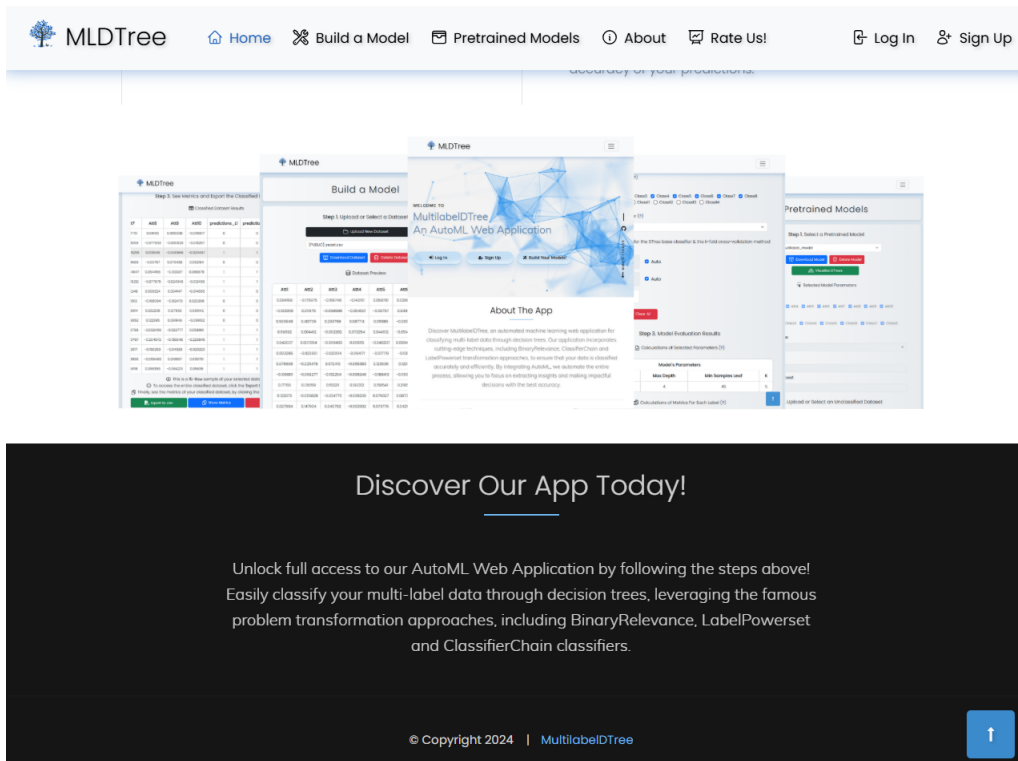
About The App

Discover MultilabelDTree, an automated machine learning web application for classifying multi-label data through decision trees. Our application incorporates cutting-edge techniques, including BinaryRelevance, LabelPowerset and ClassifierChain transformation approaches, to ensure that your data is classified accurately and efficiently. By integrating AutoML, we automate the entire process, allowing you to focus on extracting insights and making impactful decisions with the best accuracy.

Σχήμα 5.1: Αρχική Σελίδα του «MultilabelDTree»



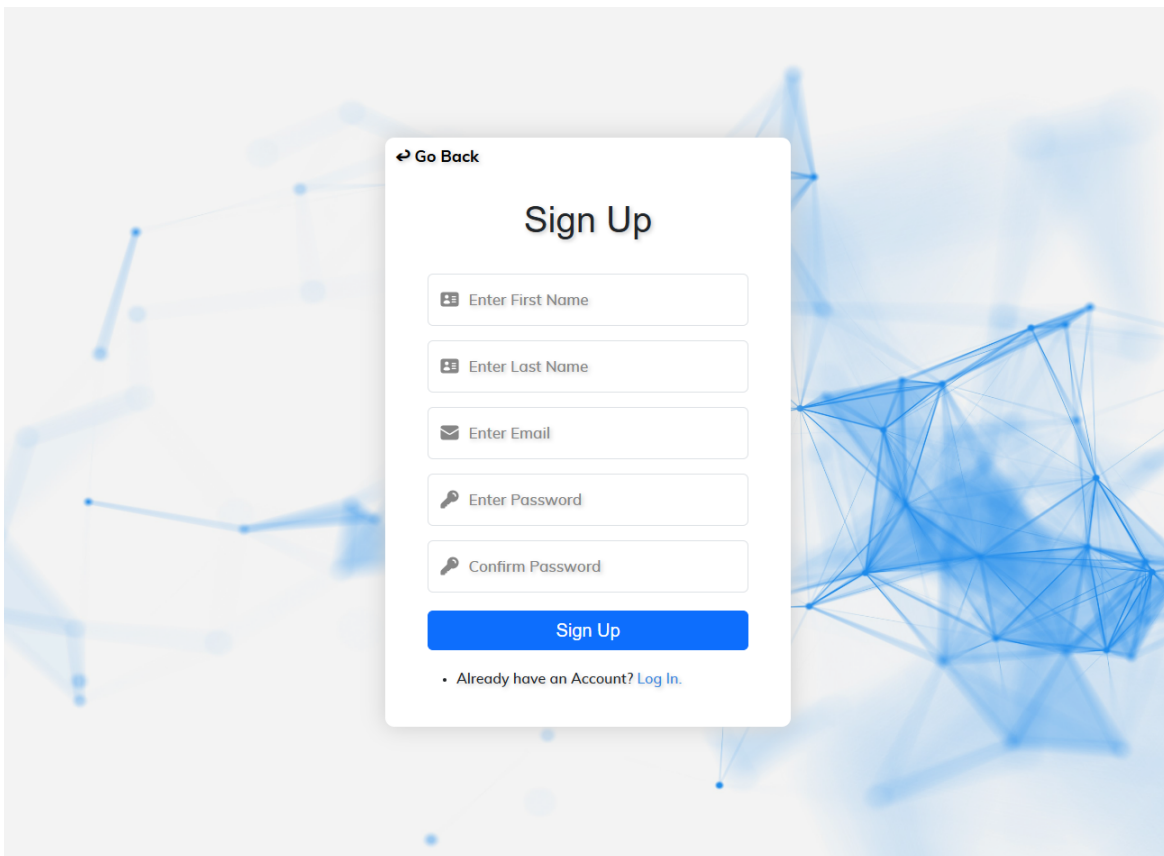
Σχήμα 5.2: Δυνατότητες και Λειτουργίες του «MultilabelDTree»



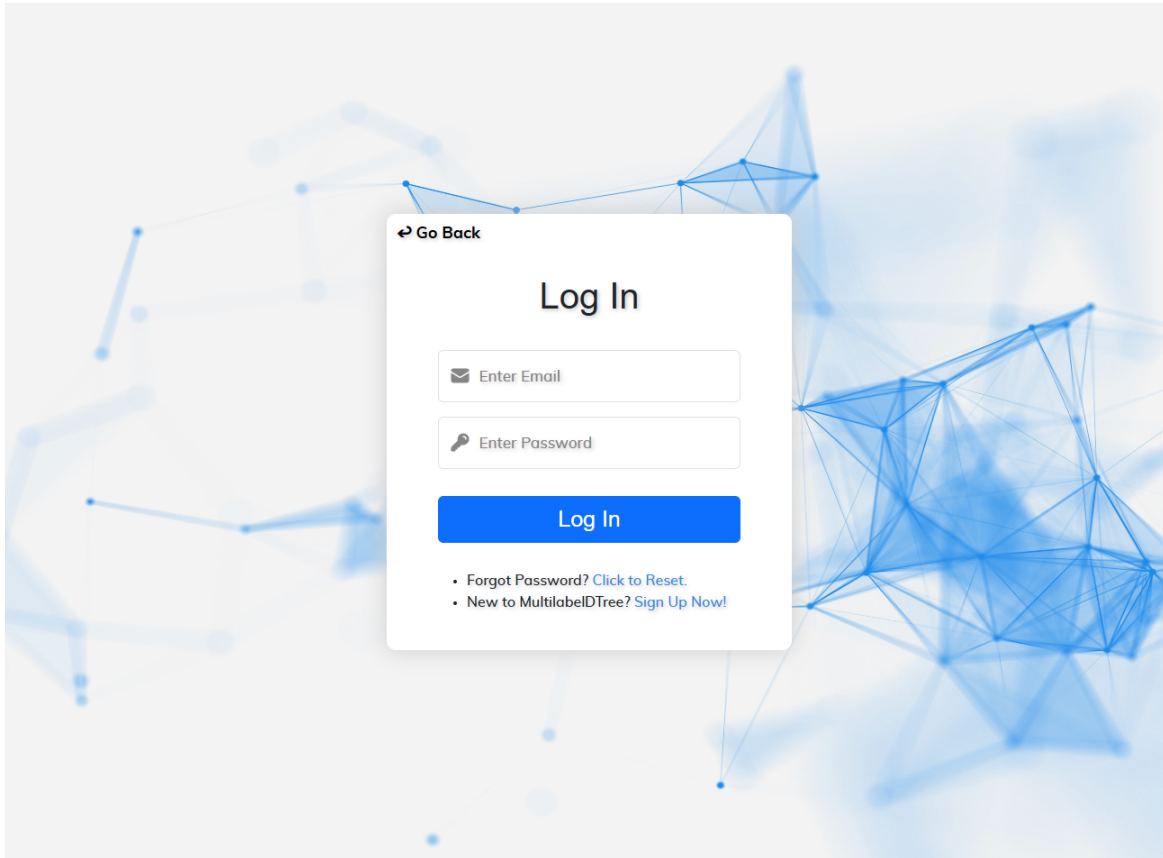
Σχήμα 5.3: Φωτογραφίες και Υποσέλιδο του «MultilabelDTree»

5.2 Sign Up και Log In του Χρήστη

Για την Εγγραφή (Sign Up) ή την Είσοδο (Log In) στην εφαρμογή, ο χρήστης μπορεί να επιλέξει τις αντίστοιχες επιλογές που βρίσκονται στο επάνω μέρος της αρχικής σελίδας, πατώντας τα σχετικά κουμπιά. Στην επιλογή της εγγραφής, απαιτείται η συμπλήρωση των βασικών στοιχείων του χρήστη μέσω της ειδικής φόρμας. Τα στοιχεία αυτά είναι το όνομα, το επώνυμο, η διεύθυνση email και ένας ασφαλής κωδικός πρόσβασης. Μετά την ολοκλήρωση της εγγραφής, ο χρήστης αποκτά πρόσβαση σε όλες τις λειτουργίες της εφαρμογής. Αντίστοιχα, μέσω της επιλογής εισόδου, οι χρήστες που έχουν ήδη λογαριασμό πρέπει να εισάγουν τα βασικά τους στοιχεία, δηλαδή το email και τον κωδικό πρόσβασης, έτσι ώστε να συνεχίσουν την περιήγησή τους, απολαμβάνοντας πλήρη πρόσβαση στις δυνατότητες του «MultilabelDTree». Παρακάτω, στο Σχήμα 5.4 και Σχήμα 5.5, παρουσιάζονται οι σελίδες εγγραφής και εισόδου αντίστοιχα.

The image shows a 'Sign Up' form centered on a light blue background with a network-like pattern of dots and lines. The form is white with a rounded border. At the top left of the form is a 'Go Back' button with a left-pointing arrow. The title 'Sign Up' is centered below the button. There are five input fields, each with a small icon on the left: 'Enter First Name' (person icon), 'Enter Last Name' (person icon), 'Enter Email' (envelope icon), 'Enter Password' (key icon), and 'Confirm Password' (key icon). Below the input fields is a prominent blue 'Sign Up' button. At the bottom of the form, there is a link: '• Already have an Account? [Log In.](#)'

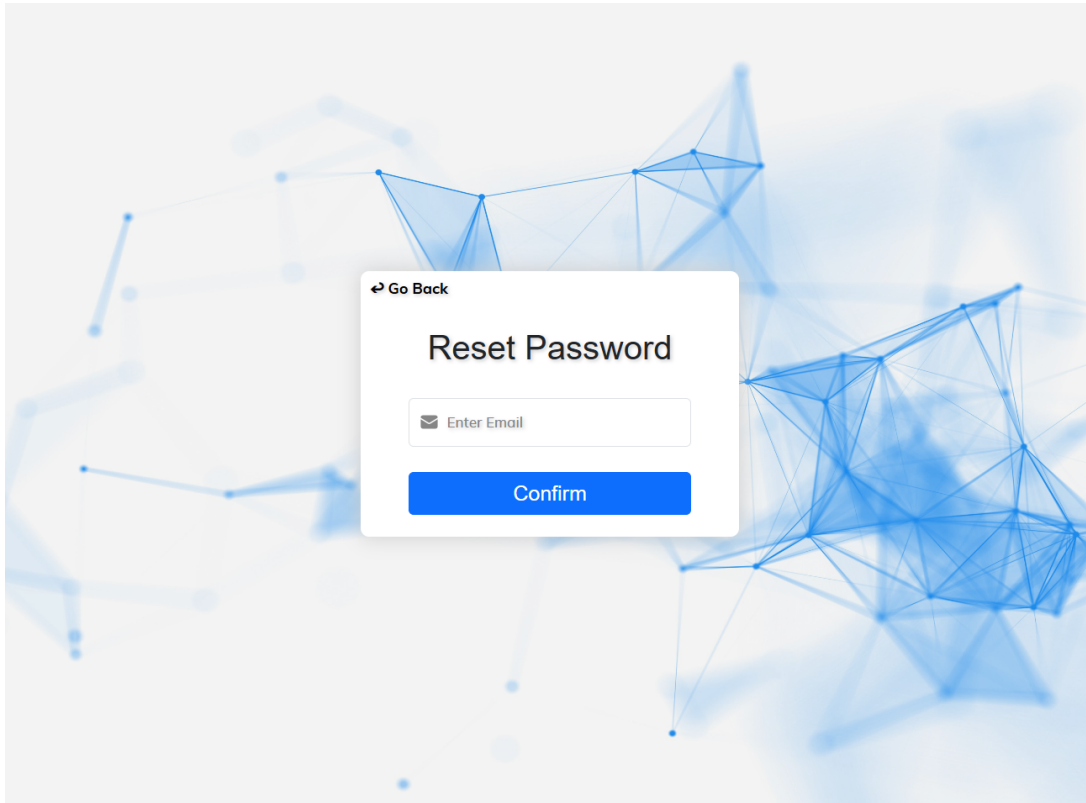
Σχήμα 5.4: Σελίδα Εγγραφής του «MultilabelDTree»



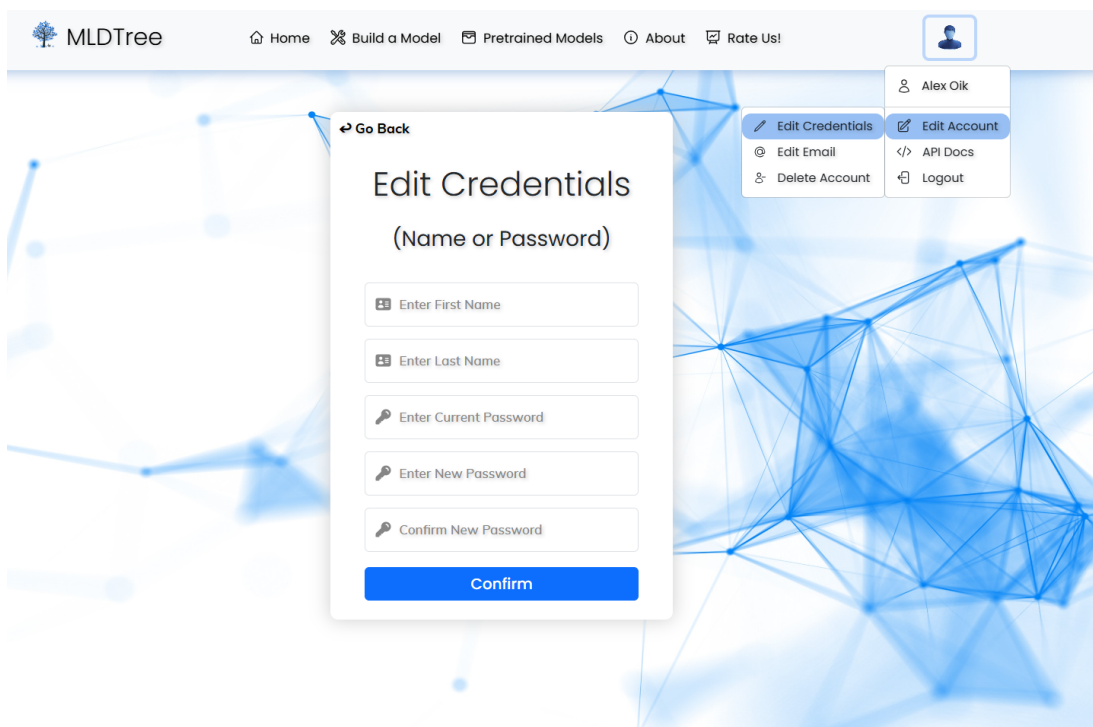
Σχήμα 5.5: Σελίδα Εισόδου του «MultilabelDTree»

5.3 Ανάκτηση Κωδικού και Επεξεργασία Λογαριασμού

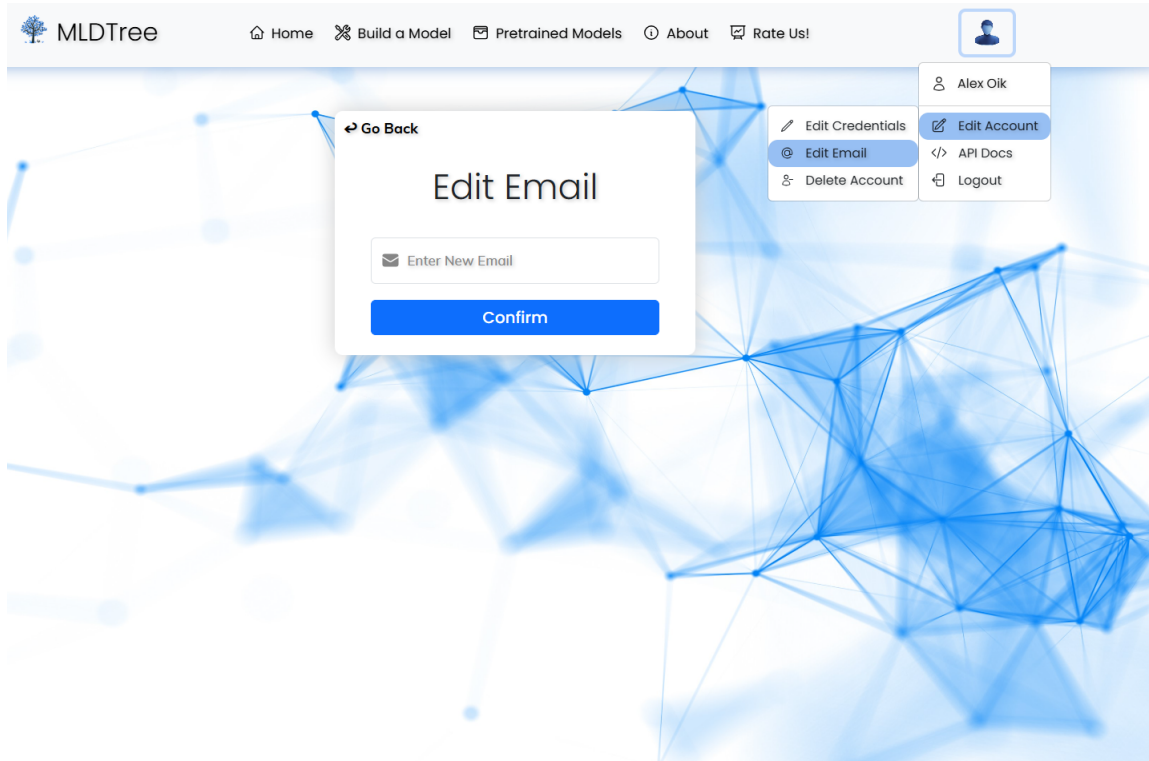
Σε περίπτωση που ο χρήστης ξεχάσει τον κωδικό πρόσβασης, μπορεί να μεταβεί στην σελίδα ανάκτησης κωδικού «Reset Password» (Σχήμα 5.6), η οποία είναι διαθέσιμη από τον σχετικό σύνδεσμο που βρίσκεται κάτω από την φόρμα εισόδου. Ακολουθώντας τα βήματα που υποδεικνύονται, δηλαδή την εισαγωγή του email που χρησιμοποιήθηκε κατά την εγγραφή, ο χρήστης λαμβάνει ένα μήνυμα επαναφοράς κωδικού, το οποίο περιέχει τον σχετικό σύνδεσμο για την δημιουργία ενός νέου κωδικού πρόσβασης για τον λογαριασμό του. Επιπλέον, στο μενού χρήστη, παρέχεται η δυνατότητα τροποποίησης των βασικών στοιχείων του λογαριασμού. Συγκεκριμένα, υπάρχει η σελίδα «Edit Credentials» για την επεξεργασία του ονοματεπώνυμου ή του κωδικού πρόσβασης, η σελίδα «Edit Email» για την αλλαγή του email, και η σελίδα «Delete Account» για τη διαγραφή του λογαριασμού, εφόσον το επιθυμεί ο χρήστης. Οι σχετικές σελίδες απεικονίζονται στα επόμενα σχήματα.



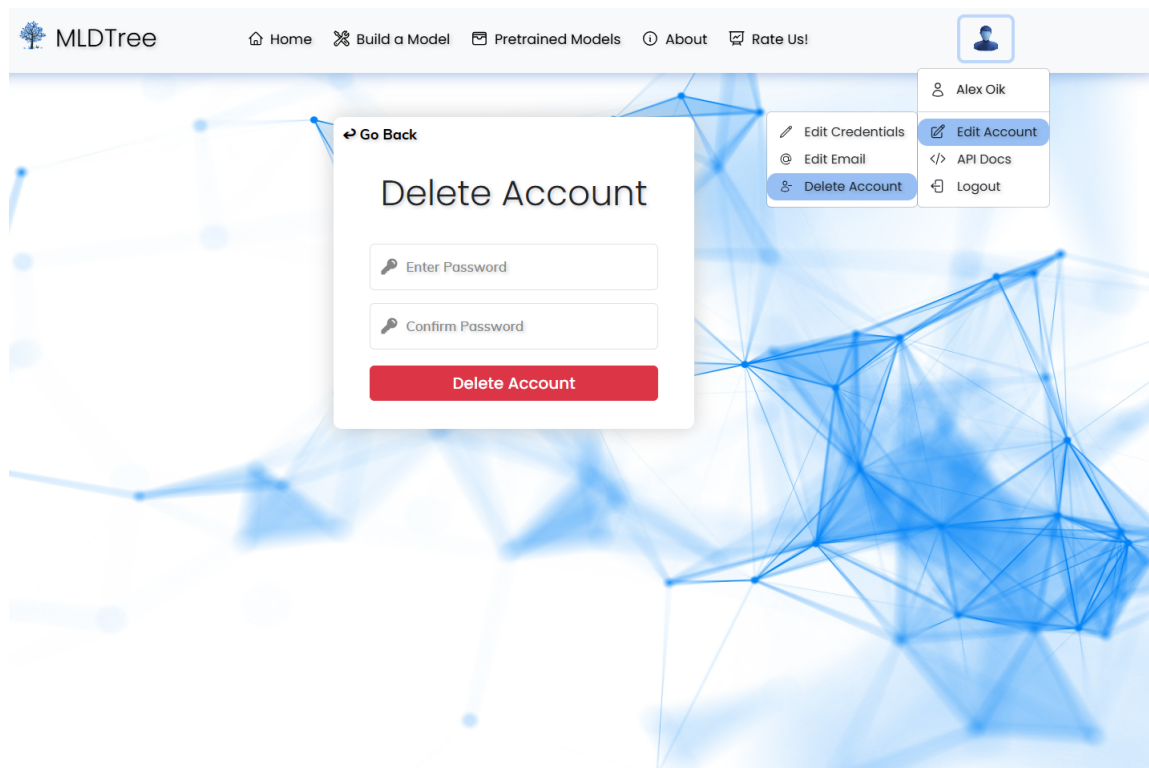
Σχήμα 5.6: Σελίδα Ανάκτησης Κωδικού Λογαριασμού του «MultilabelDTree»



Σχήμα 5.7: Σελίδα Επεξεργασίας Στοιχείων Λογαριασμού του «MultilabelDTree»



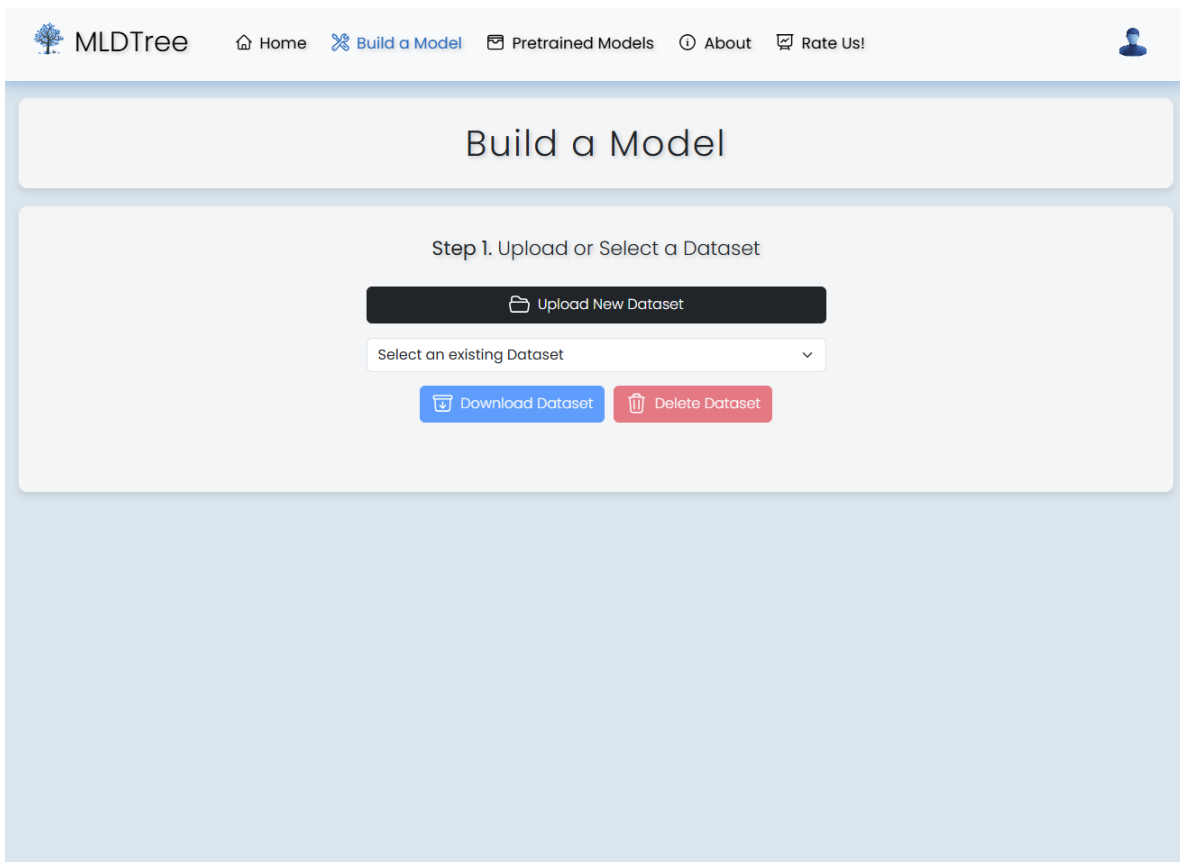
Σχήμα 5.8: Σελίδα Επεξεργασίας Email Λογαριασμού του «MultilabelDTree»



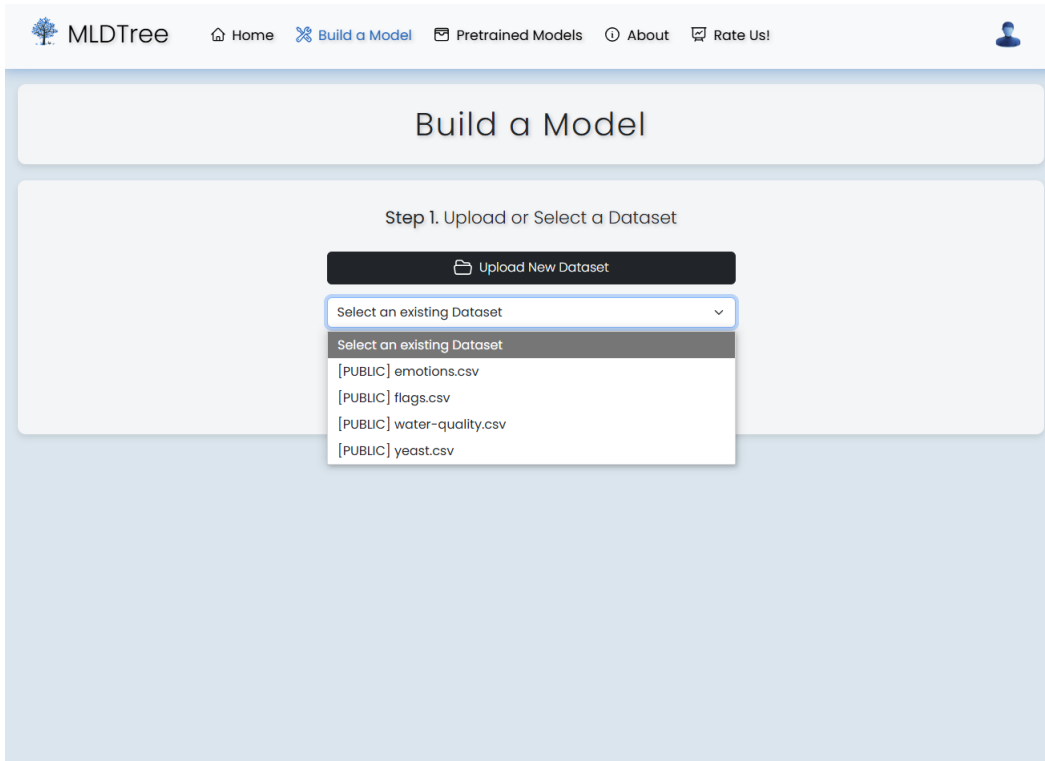
Σχήμα 5.9: Σελίδα Διαγραφής Λογαριασμού του «MultilabelDTree»

5.4 Δημιουργία Multi-label Μοντέλων

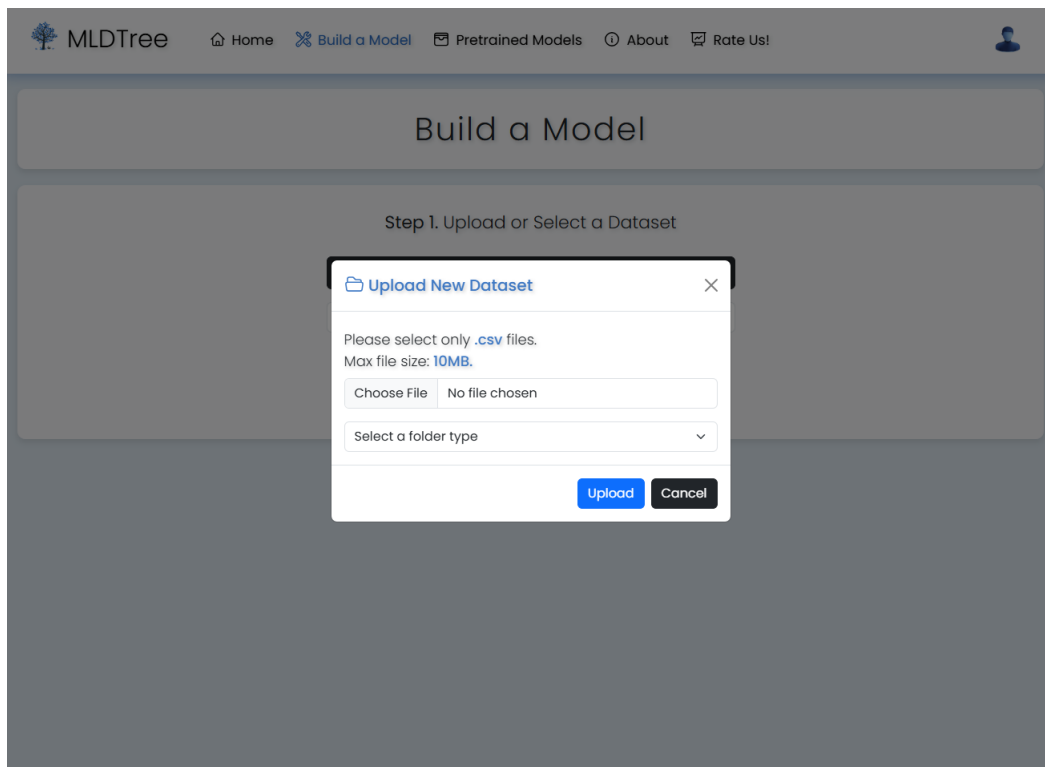
Κατά την είσοδο του χρήστη στην πρώτη κύρια σελίδα της εφαρμογής (Σχήμα 5.10), με τίτλο «Build a Model», του δίνεται η δυνατότητα να δημιουργήσει ένα multi-label μοντέλο εκπαίδευσης. Αρχικά, ο χρήστης καλείται να επιλέξει ένα από τα διαθέσιμα datasets που εμφανίζονται στη σχετική λίστα (Σχήμα 5.11), ή να ανεβάσει ο ίδιος ένα δικό του νέο multi-label σύνολο δεδομένων (Σχήμα 5.12) και να το επιλέξει. Όσον αφορά το ανέβασμα του συνόλου δεδομένων από τον χρήστη, το αρχείο θα πρέπει να έχει κατάληξη .csv και μέγεθος έως 10MB. Επιπλέον, ο χρήστης καλείται να επιλέξει τον κατάλληλο τύπο φακέλου, δηλαδή είτε public είτε private, ανάλογα με το πώς επιθυμεί να ανεβάσει το αρχείο. Έπειτα, εμφανίζεται μία προεπισκόπηση (Preview) 15 γραμμών από το επιλεγμένο σύνολο δεδομένων, συνοδευόμενη από τον συνολικό αριθμό εγγραφών που περιλαμβάνει (Σχήμα 5.13). Στη συνέχεια, ακολουθεί η επιλογή των κατάλληλων παραμέτρων του μοντέλου, η αξιολόγησή του, αλλά και η αποθήκευσή του από τον χρήστη, εφόσον καλύπτει τις ανάγκες του. Οι λειτουργίες αυτές περιγράφονται παρακάτω.



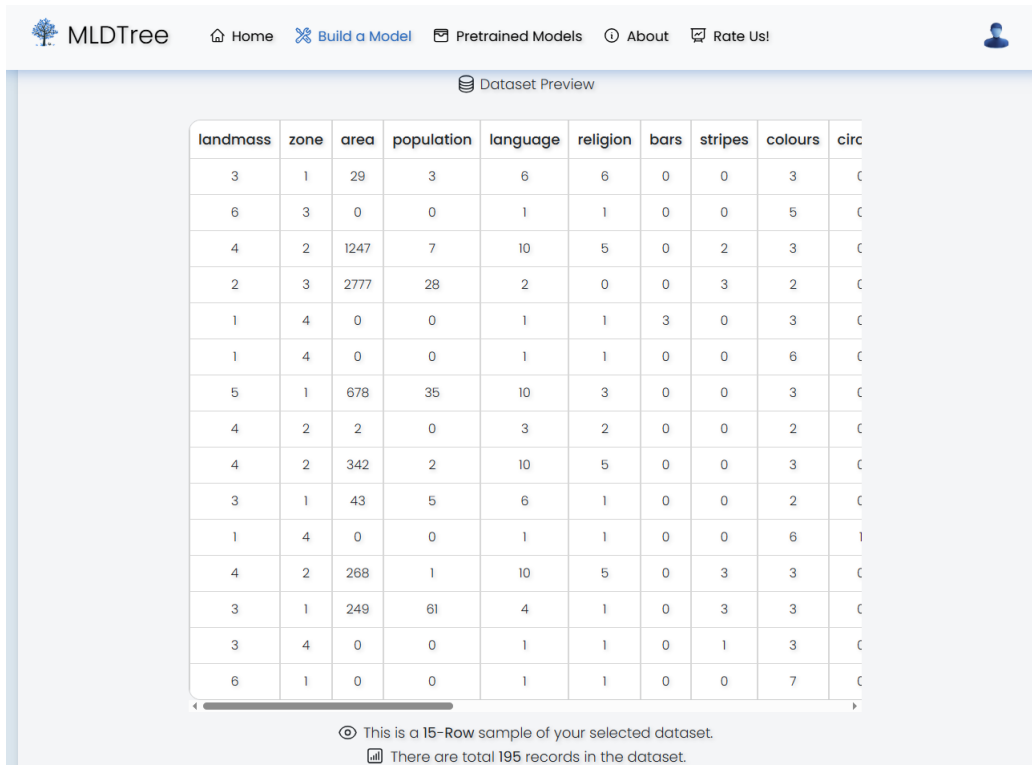
Σχήμα 5.10: Σελίδα Δημιουργίας Multi-label Μοντέλων του «MultilabelDTree»



Σχήμα 5.11: Λίστα για την Επιλογή ενός Multi-label Συνόλου Δεδομένων



Σχήμα 5.12: Modal για το Ανέβασμα ενός Multi-label Συνόλου Δεδομένων

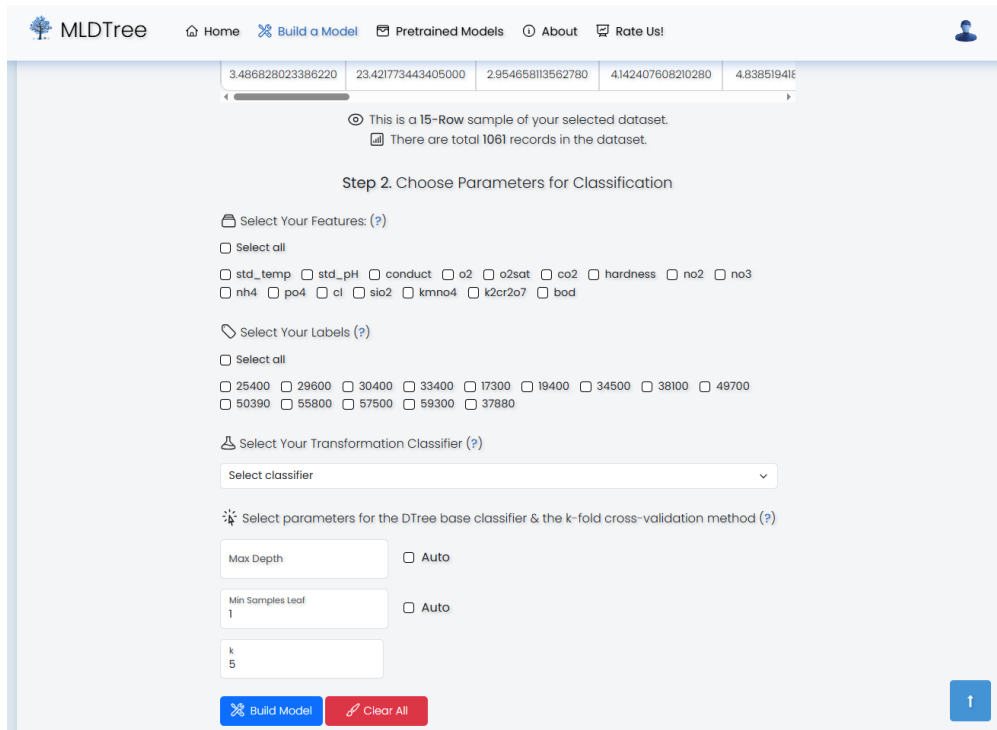


landmass	zone	area	population	language	religion	bars	stripes	colours	circ
3	1	29	3	6	6	0	0	3	0
6	3	0	0	1	1	0	0	5	0
4	2	1247	7	10	5	0	2	3	0
2	3	2777	28	2	0	0	3	2	0
1	4	0	0	1	1	3	0	3	0
1	4	0	0	1	1	0	0	6	0
5	1	678	35	10	3	0	0	3	0
4	2	2	0	3	2	0	0	2	0
4	2	342	2	10	5	0	0	3	0
3	1	43	5	6	1	0	0	2	0
1	4	0	0	1	1	0	0	6	1
4	2	268	1	10	5	0	3	3	0
3	1	249	61	4	1	0	3	3	0
3	4	0	0	1	1	0	1	3	0
6	1	0	0	1	1	0	0	7	0

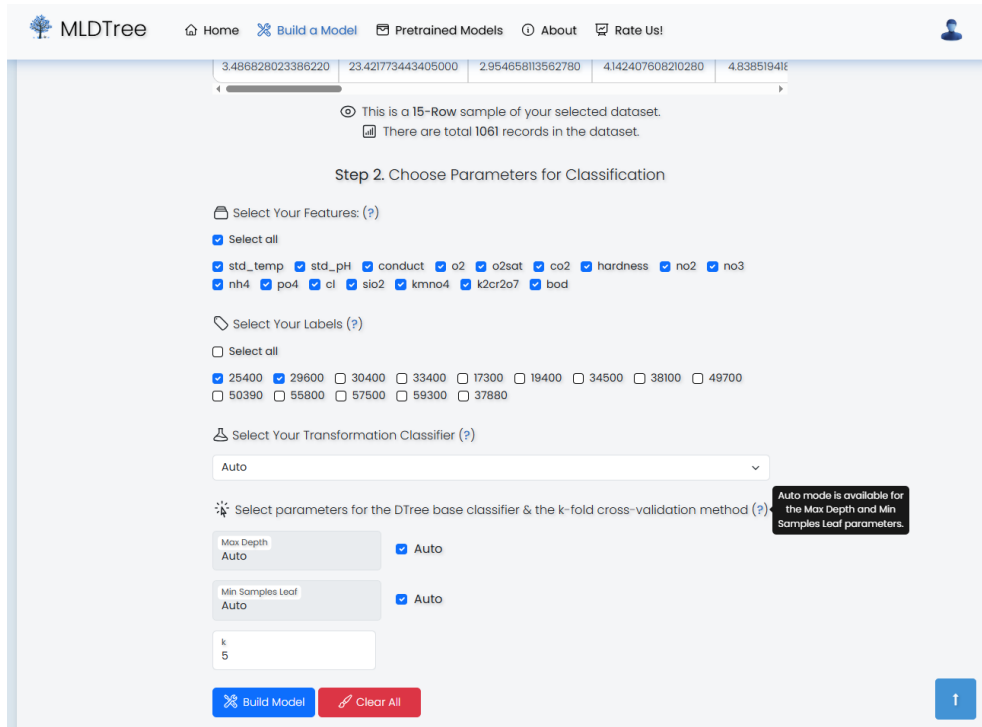
ⓘ This is a 15-Row sample of your selected dataset.
 📄 There are total 195 records in the dataset.

Σχήμα 5.13: Πίνακας για το Preview ενός Multi-label Συνόλου Δεδομένων

Σε αυτό το στάδιο, ο χρήστης επιλέγει τις παραμέτρους του μοντέλου του. Σε πρώτη φάση, επιλέγει τα χαρακτηριστικά (features) και τις ετικέτες (labels) από το σύνολο δεδομένων που επέλεξε. Αμέσως μετά, επιλέγει τον ταξινομητή μετασχηματισμού προβλήματος (transformation classifier) που επιθυμεί, από την διαθέσιμη λίστα. Στο τέλος, καλείται να εισάγει την τιμή για το μέγιστο βάθος (max depth) και τον ελάχιστο αριθμό δειγμάτων ανά φύλλο (min samples leaf) του DTree ταξινομητή βάσης, ενώ απαραίτητη προϋπόθεση είναι να έχει εισάγει και μία τιμή k για την μέθοδο k -fold cross-validation (Σχήμα 5.14). Η αυτόματη λειτουργία (Auto Mode) είναι διαθέσιμη αποκλειστικά για τις παραμέτρους του ταξινομητή μετασχηματισμού προβλήματος, του max depth, αλλά και του min samples leaf αντίστοιχα. Για κάθε παράμετρο, εμφανίζεται ένα βοηθητικό μήνυμα (tooltip) για την καθοδήγηση και την καλύτερη κατανόηση του χρήστη (Σχήμα 5.15).



Σχήμα 5.14: Επιλογή Παραμέτρων για τη Δημιουργία του Multi-label Μοντέλου



Σχήμα 5.15: Παράδειγμα Βοηθητικού Μηνύματος (tooltip) για τον Χρήστη

Στη συνέχεια, παρουσιάζονται τα αποτελέσματα αξιολόγησης του μοντέλου, τα οποία εμφανίζονται σε μορφή πινάκων για λόγους απλότητας (Σχήμα 5.16). Αρχικά, καταγράφονται οι τελικές τιμές των παραμέτρων του μοντέλου, ώστε ο χρήστης να γνωρίζει ποιες χρησιμοποιήθηκαν. Αυτό είναι ιδιαίτερα χρήσιμο όταν έχει ενεργοποιηθεί η αυτόματη λειτουργία (Auto Mode), καθώς σε αυτήν την περίπτωση ο χρήστης δεν καθορίζει εξαρχής τις τιμές των παραμέτρων. Αμέσως μετά, παρουσιάζονται τα αποτελέσματα των μετρικών αξιολόγησης για κάθε ετικέτα ξεχωριστά. Συνεπώς, ο αριθμός των πινάκων ισοδυναμεί με τον αριθμό των ετικετών. Ο τελευταίος πίνακας συνοψίζει τις μέσες τιμές των μετρικών αξιολόγησης για το σύνολο του μοντέλου. Όπως έχει ήδη αναφερθεί, οι μετρικές που χρησιμοποιούνται για την αξιολόγηση του μοντέλου περιλαμβάνουν τα εξής: Hamming Loss, Accuracy, Precision, Recall και F-score.

Τέλος, μετά την παρουσίαση των αποτελεσμάτων αξιολόγησης του μοντέλου, ο χρήστης καλείται να αποφασίσει εάν επιθυμεί να το αποθηκεύσει για μελλοντική χρήση. Εάν ναι, θα πρέπει να εισάγει ένα όνομα για το μοντέλο του και έπειτα να πατήσει το κουμπί «Save Model» (Σχήμα 5.17). Μετά το πάτημα του κουμπιού, εμφανίζεται, μέσω ενός modal, ένα μήνυμα που ενημερώνει τον χρήστη για την επιτυχή αποθήκευση του μοντέλου και τον προτρέπει να μεταβεί στη σελίδα «Pretrained Models» της εφαρμογής, όπου μπορεί να το δει (Σχήμα 5.18).

MLDTree Home Build a Model Pretrained Models About Rate Us!

Step 3. Model Evaluation Results

Calculations of Selected Parameters (?)

Model's Parameters			
Classifier	Max Depth	Min Samples Leaf	K
LabelPowerset	1	34	5

Calculations of Metrics For Each Label (?)

Metrics for Label 1 - 25400			
Label	Precision	Recall	F-score
1	0.52	0.57	0.54
0	0.59	0.62	0.59

Metrics for Label 2 - 29600			
Label	Precision	Recall	F-score
0	0.29	0.4	0.34
1	0.44	0.6	0.51

Calculations of Avg Metrics (?)

Average Metrics				
Hamming Loss	Accuracy	Precision	Recall	F-Score
0.32	0.46	0.24	0.19	0.21

Σχήμα 5.16: Αποτελέσματα Αξιολόγησης του Multi-label Μοντέλου

Metrics for Label 1 - 25400

Label	Precision	Recall	F-score
1	0.52	0.57	0.54
0	0.59	0.62	0.59

Metrics for Label 2 - 29600

Label	Precision	Recall	F-score
0	0.29	0.4	0.34
1	0.44	0.6	0.51

Calculations of Avg Metrics (?)

Average Metrics

Hamming Loss	Accuracy	Precision	Recall	F-Score
0.32	0.46	0.24	0.19	0.21

Step 4. Save Your Created Model

Name your Model. First character should be a letter. Spaces aren't allowed.

Model Name: my_multilabel_model

Buttons: Save Model, Cancel

Σχήμα 5.17: Αποθήκευση του Multi-label Μοντέλου

Metrics for Label 1 - saltires

Label	Precision	Recall	F-score
0	0.97	0.87	0.89
1	0.98	0.73	0.81

Metrics for Label 2 - crescent

Label	Precision	Recall	F-score
0	0.96	0.96	0.96
1	0	0	0

Average Metrics

Hamming Loss	Accuracy	Precision	Recall	F-Score
0.06	0.88	0.5	0.3	0.36

Step 4. Save Your Created Model

Model Name: alex_model

Buttons: Save Model, Cancel

Notification

Model successfully saved.

You can now check out your model by going in the [Pretrained Models](#) page!

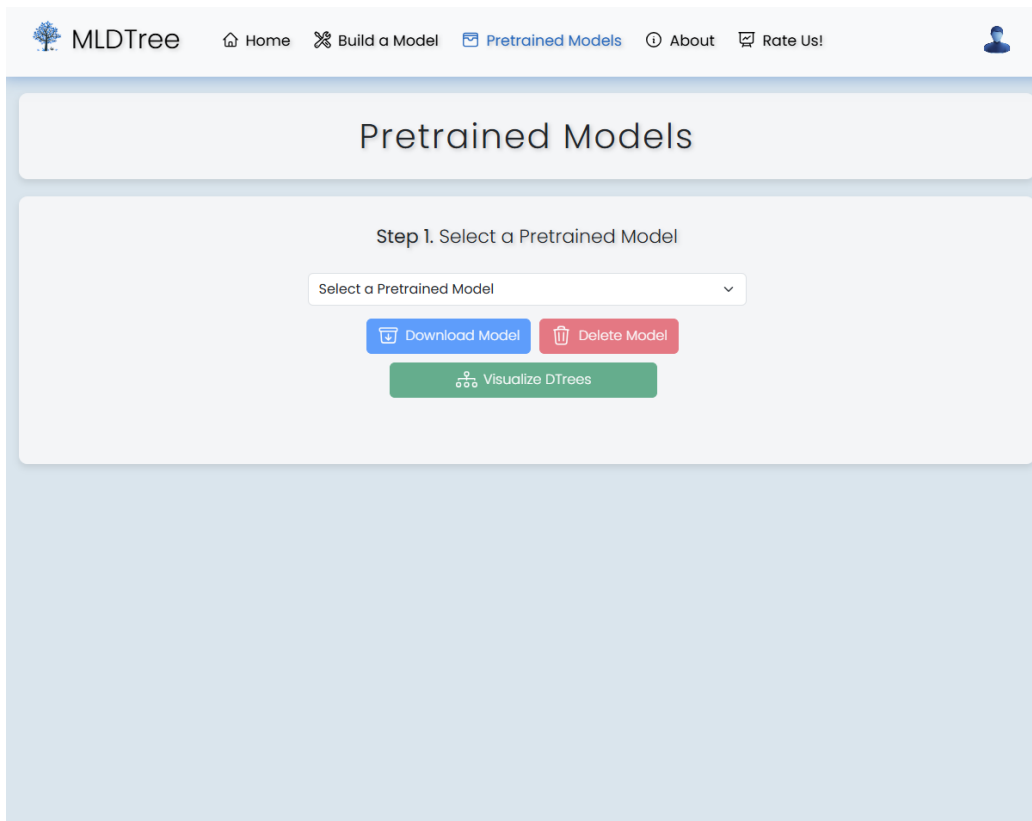
OK

Σχήμα 5.18: Μήνυμα Προτροπής για τον Χρήστη

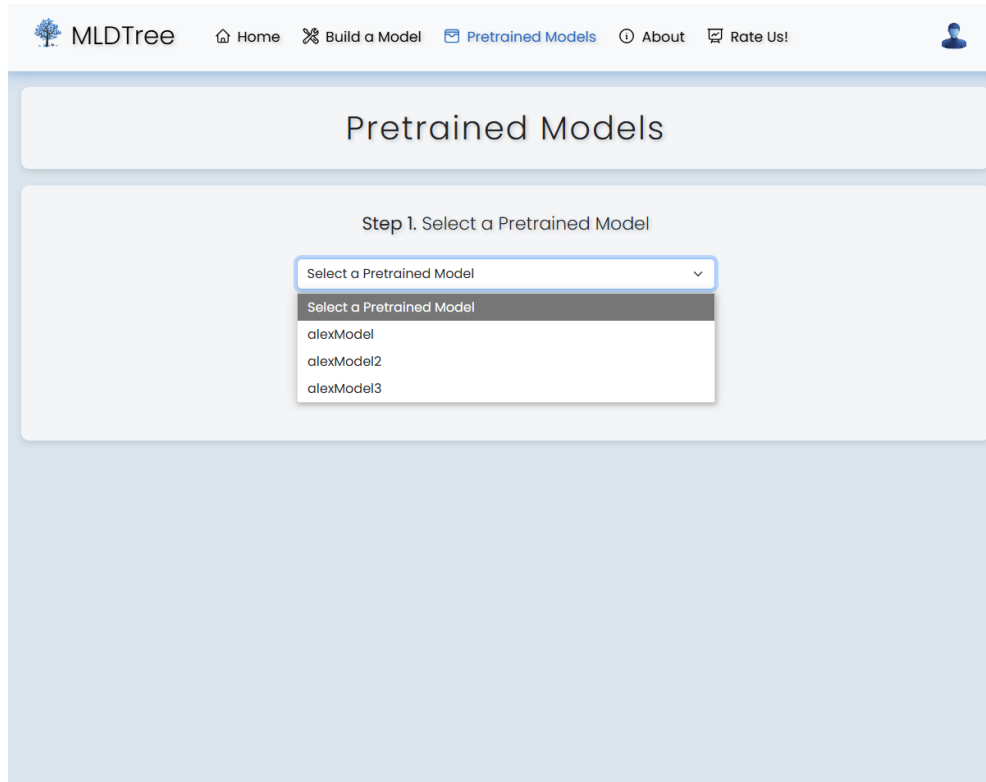
5.5 Χρήση Προεκπαιδευμένων Μοντέλων

Από την άλλη, όταν ο χρήστης μεταβεί στη δεύτερη κύρια σελίδα της εφαρμογής, δηλαδή τη σελίδα «Pretrained Models» (Σχήμα 5.19), έχει τη δυνατότητα να δει τα προεκπαιδευμένα μοντέλα που έχει δημιουργήσει με τις σχετικές παραμέτρους τους. Αρχικά, ο χρήστης θα πρέπει να επιλέξει το επιθυμητό μοντέλο από τη λίστα (Σχήμα 5.20). Μετά την επιλογή, εμφανίζονται οι παράμετροι του συγκεκριμένου μοντέλου, οι οποίες χρησιμοποιήθηκαν κατά τη δημιουργία του. Συγκεκριμένα, παρουσιάζονται τα χαρακτηριστικά (features), οι ετικέτες (labels), ο τύπος του ταξινομητή (classifier type), το μέγιστο βάθος (max depth) και το ελάχιστο πλήθος δειγμάτων ανά φύλλο (min samples leaf), όπως φαίνεται και στο Σχήμα 5.21.

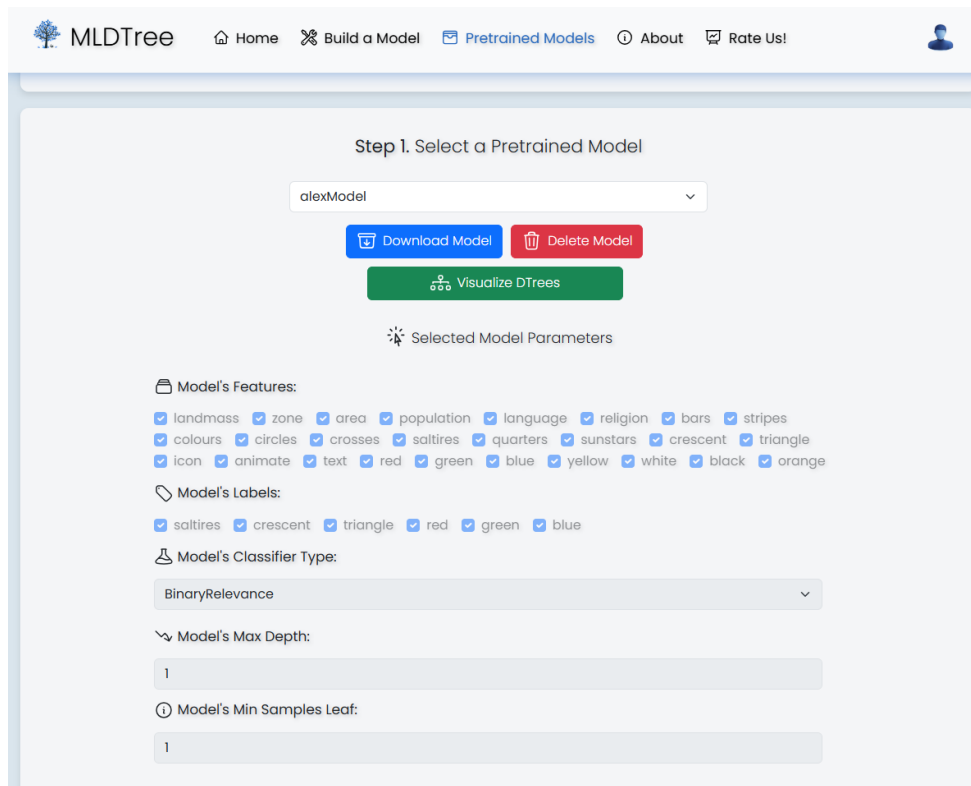
Επιπλέον, ο χρήστης έχει τη δυνατότητα να κατεβάσει ή να διαγράψει το επιλεγμένο μοντέλο του, χρησιμοποιώντας τα κουμπιά «Download Model» και «Delete Model», αντίστοιχα. Μια εξίσου σημαντική λειτουργία είναι η οπτικοποίηση των Δέντρων Απόφασης του μοντέλου, η οποία ενεργοποιείται όταν ο χρήστης πατήσει το κουμπί «Visualize DTrees». Έτσι, ανάλογα τον τύπο του μετασχηματισμού προβλήματος που έχει το μοντέλο, δημιουργούνται τα κατάλληλα Δέντρα Απόφασης, τα οποία είναι διαθέσιμα για προβολή μέσω ενός Modal. Ο χρήστης, έπειτα, μπορεί να κατεβάσει τα δέντρα σε .png μορφή, πατώντας τα αντίστοιχα κουμπιά. Ένα παράδειγμα παρουσιάζεται στο Σχήμα 5.22, όπου εμφανίζονται τα δέντρα ενός 'Binary Relevance Classification'.



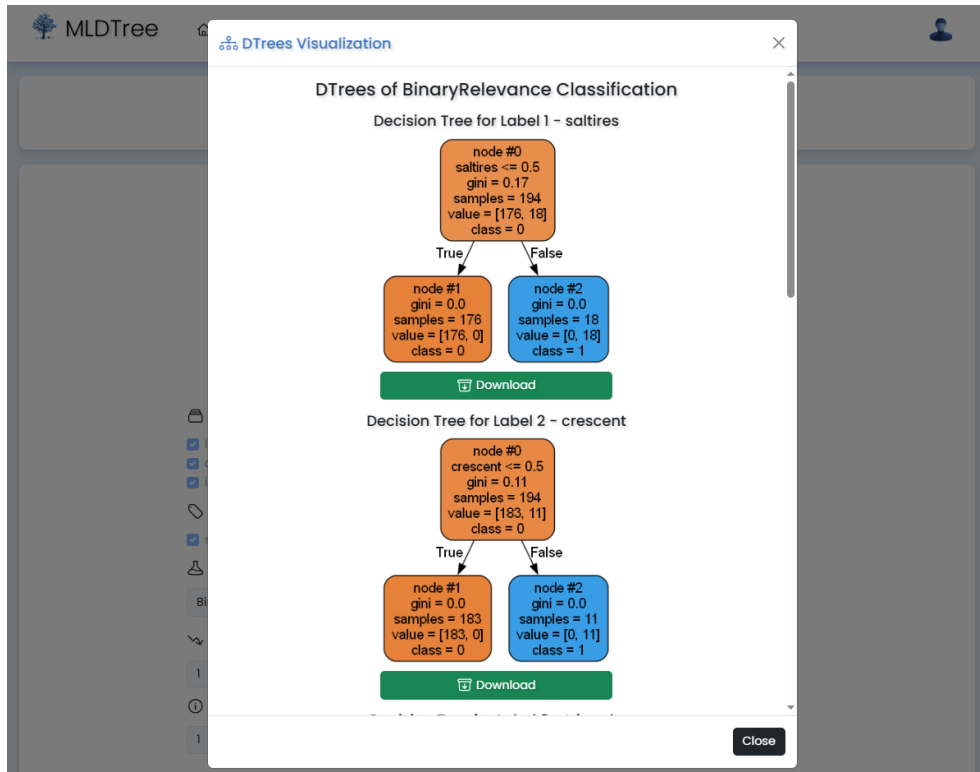
Σχήμα 5.19: Σελίδα Προεκπαιδευμένων Μοντέλων του MultilabelDTree



Σχήμα 5.20: Λίστα για την Επιλογή ενός Προεκπαιδευμένου Μοντέλου

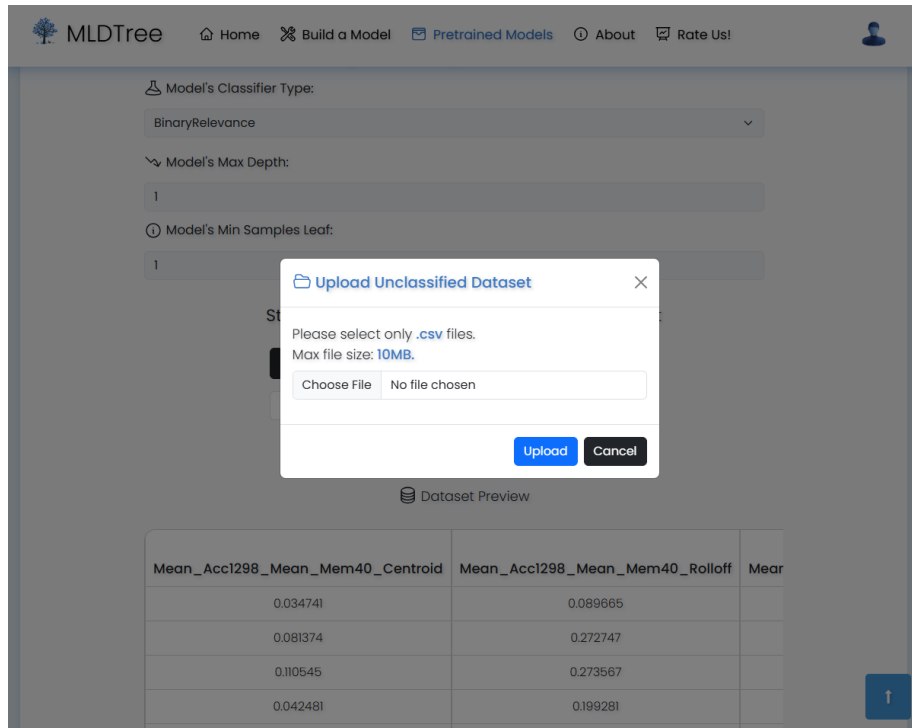


Σχήμα 5.21: Προβολή Επιλεγμένων Παραμέτρων του Μοντέλου

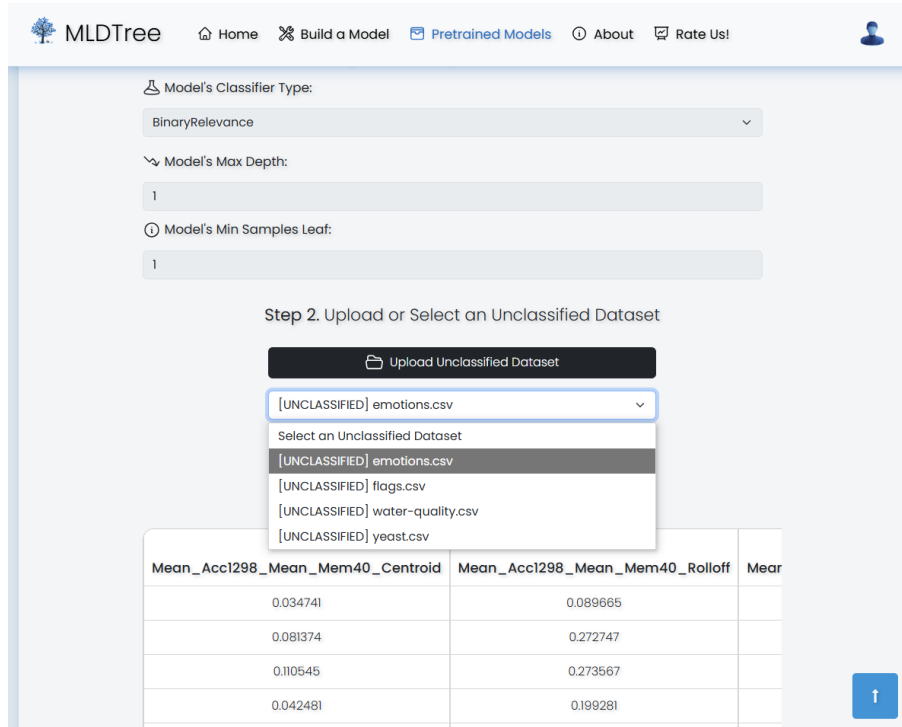


Σχήμα 5.22: Οπτικοποίηση Δέντρων Απόφασης μέσω Modal

Παρακάτω, ο χρήστης μπορεί να ξεκινήσει τη διαδικασία κατηγοριοποίησης των νέων στιγμιότυπων ενός αταξινόμητου συνόλου δεδομένων. Αρχικά, θα πρέπει να ανεβάσει ένα νέο αταξινόμητο σύνολο δεδομένων με πολλαπλές ετικέτες (multi-label) και να το επιλέξει (Σχήμα 5.23). Το νέο σύνολο δεδομένων πρέπει να πληροί συγκεκριμένες προδιαγραφές, δηλαδή να είναι αρχείο τύπου .csv και να μην υπερβαίνει τα 10MB σε μέγεθος. Εναλλακτικά, μπορεί να επιλέξει ένα από τα ήδη ανεβασμένα σύνολα δεδομένων που εμφανίζονται στη διαθέσιμη λίστα (Σχήμα 5.24). Ομοίως, ο χρήστης έχει τη δυνατότητα να διαγράψει ένα επιλεγμένο αταξινόμητο σύνολο δεδομένων από τη λίστα ή να το κατεβάσει, χρησιμοποιώντας τα κουμπιά «Delete Dataset» και «Download Dataset» αντίστοιχα. Μετά την επιλογή του συνόλου δεδομένων, εμφανίζεται μια σχετική προεπισκόπηση 10 γραμμών (Dataset Preview) σε μορφή πίνακα, συνοδευόμενη από το σύνολο των εγγραφών του (Σχήμα 5.25).



Σχήμα 5.23: Modal για το Ανέβασμα ενός Αταξινόμητου Συνόλου Δεδομένων



Σχήμα 5.24: Λίστα για την Επιλογή ενός Αταξινόμητου Συνόλου Δεδομένων

landmass	zone	area	population	language	religion	bars	stripes	colours	circ
3	1	29	3	6	6	0	0	3	0
6	3	0	0	1	1	0	0	5	0
4	2	1247	7	10	5	0	2	3	0
2	3	2777	28	2	0	0	3	2	0
1	4	0	0	1	1	3	0	3	0
1	4	0	0	1	1	0	0	6	0
5	1	678	35	10	3	0	0	3	0
4	2	2	0	3	2	0	0	2	0
4	2	342	2	10	5	0	0	3	0
3	1	43	5	6	1	0	0	2	0
1	4	0	0	1	1	0	0	6	0
4	2	268	1	10	5	0	3	3	0
3	1	249	61	4	1	0	3	3	0
3	4	0	0	1	1	0	1	3	0
6	1	0	0	1	1	0	0	7	0

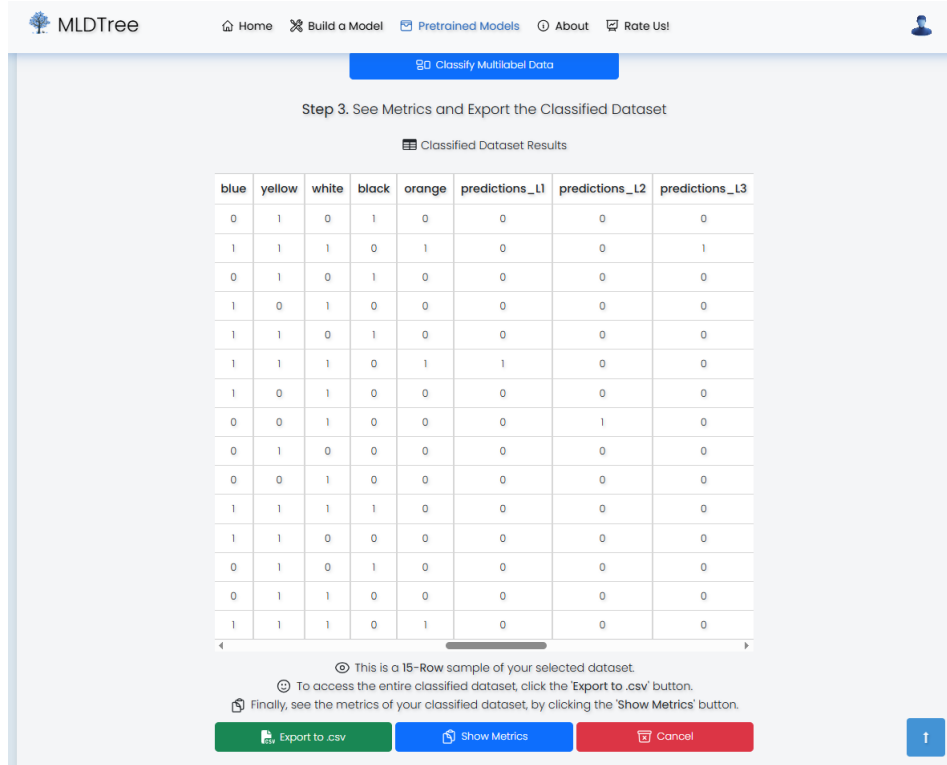
© This is a 15-Row sample of your selected dataset.
 There are total 195 records in the dataset.

Classify Multilabel Data

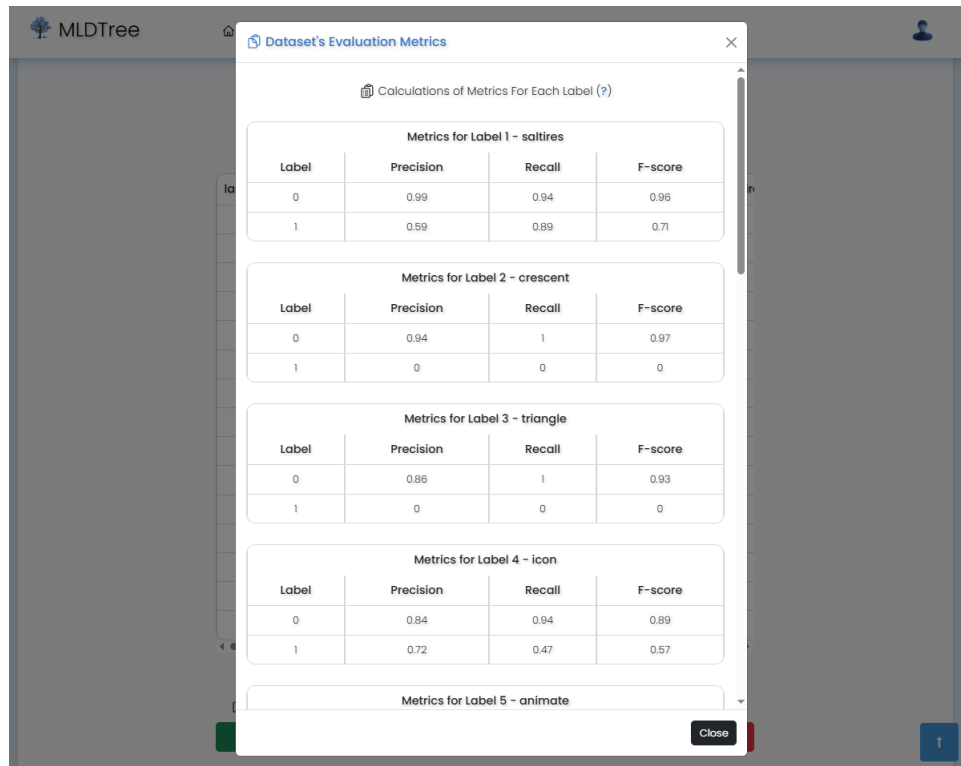
Σχήμα 5.25: Πίνακας για το Preview ενός Αταξινομητου Συνόλου Δεδομένων

Όπως φαίνεται στο παραπάνω σχήμα, ο χρήστης μπορεί να ξεκινήσει τη διαδικασία κατηγοριοποίησης των νέων στιγμιότυπων του αταξινομητου συνόλου, πατώντας το κουμπί «Classify Multilabel Dataset». Να σημειωθεί, ότι τα χαρακτηριστικά (features) του μοντέλου θα πρέπει να ταιριάζουν με τις στήλες του μη ταξινομημένου συνόλου δεδομένων, ώστε η διαδικασία να πραγματοποιηθεί και να εκτελεστεί σωστά. Ο συγκεκριμένος έλεγχος πραγματοποιείται μέσω της σχετικής php λειτουργίας. Εάν διαπιστωθεί ασυμβατότητα, εμφανίζεται το κατάλληλο μήνυμα, το οποίο προτρέπει τον χρήστη να επιλέξει ένα διαφορετικό σύνολο δεδομένων που πληροί τις απαιτήσεις.

Εφόσον ολοκληρωθεί επιτυχώς η κατηγοριοποίηση του αταξινομητου συνόλου, εμφανίζονται τα ταξινομημένα αποτελέσματα σε μορφή πίνακα. Ο πίνακας αυτός περιλαμβάνει τα στιγμιότυπα από τις στήλες features, μαζί με τις προβλέψεις τους από τις στήλες predictions_i (όπου i = ο αριθμός της ετικέτας). Οι προβλέψεις μπορούν να έχουν είτε τιμή 0, που δηλώνει την απουσία, είτε τιμή 1, που δηλώνει την παρουσία μιας ετικέτας σε ένα στιγμιότυπο. Επιπρόσθετα, ο χρήστης έχει τη δυνατότητα να εξάγει τα αποτελέσματα σε ένα .csv αρχείο από το «Export to .csv» button, να ακυρώσει την όλη διαδικασία από το «Cancel» button, αλλά και να εμφανίσει τις μετρικές αξιολόγησης του πλέον ταξινομημένου συνόλου σε Modal, μέσω του «Show Metrics» button (Σχήμα 5.27).



Σχήμα 5.26: Πίνακας Ταξινομημένων Στιγμοτύπων του Συνόλου Δεδομένων



Σχήμα 5.27: Modal με τις Μετρικές Αξιολόγησης του Ταξινομημένου Συνόλου

Κεφάλαιο 6

Συμπεράσματα και Μελλοντικές Επεκτάσεις

6.0.1 Συμπεράσματα

Στα πλαίσια της παρούσας διπλωματικής εργασίας του τμήματος, υλοποιήθηκε η διαδικτυακή εφαρμογή «MultilabelDTree», η οποία αποτελεί ένα χρήσιμο εργαλείο ανοικτού κώδικα που διευκολύνει την κατηγοριοποίηση των δεδομένων πολλαπλών ετικετών (multi-label), χρησιμοποιώντας τα Δέντρα Απόφασης (Decision Trees). Η εφαρμογή βασίζεται στις δημοφιλείς προσεγγίσεις μετασχηματισμού προβλήματος: BinaryRelevance, LabelPowerset και ClassifierChain, δίνοντας έτσι τη δυνατότητα στους χρήστες να δημιουργούν μοντέλα που ανταποκρίνονται στις απαιτήσεις τους. Τα αποθηκευμένα μοντέλα μπορούν να χρησιμοποιηθούν μετέπειτα τόσο για την πρόβλεψη νέων multi-label δεδομένων όσο και για την απεικόνιση της δομής των Δέντρων Απόφασης.

Η διεπαφή χρήστη (UI) της εφαρμογής «MultilabelDTree» έχει σχεδιαστεί με γνώμονα τη φιλικότητα και απλότητα, ώστε να εξυπηρετεί τόσο αρχάριους όσο και έμπειρους χρήστες. Μέσα από ένα καθαρό και λειτουργικό περιβάλλον, οι χρήστες μπορούν εύκολα να ανεβάζουν τα δεδομένα τους, να επιλέγουν τις προσεγγίσεις μετασχηματισμού προβλήματος που επιθυμούν αλλά και να ρυθμίζουν τις παραμέτρους για την κατηγοριοποίηση. Τέλος, η εφαρμογή είναι και responsive, εξασφαλίζοντας έτσι μία άνετη χρήση τόσο μέσα από υπολογιστές όσο και από φορητές συσκευές.

6.0.2 Μελλοντικές Επεκτάσεις

Η διαδικτυακή εφαρμογή «MultilabelDTree» προσφέρει ένα ολοκληρωμένο περιβάλλον με πλούσιες δυνατότητες και επιλογές για την κατηγοριοποίηση. Παρόλο που η εφαρμογή καλύπτει ήδη αρκετές ανάγκες, υπάρχουν περιθώρια για περαιτέρω βελτίωση και εξέλιξη, με την ενσωμάτωση πρόσθετων λειτουργιών στο μέλλον. Ορισμένες από αυτές θα μπορούσαν να περιλαμβάνουν:

Ανάπτυξη Εφαρμογής για Κινητές Συσκευές

Αρχικά, μία μελλοντική βελτίωση θα μπορούσε να είναι η ανάπτυξη της εφαρμογής για κινητές συσκευές, τόσο για Android όσο και για iOS. Μια τέτοια δυνατότητα θα επιτρέψει στους χρήστες να έχουν πρόσβαση στην εφαρμογή από οποιαδήποτε κινητή συσκευή, ενισχύοντας έτσι την ευκολία και την προσβασιμότητα.

Ανάπτυξη Εφαρμογών με Διαφορετικές Μεθόδους και Αλγορίθμους

Μία ακόμα προοπτική, είναι η δημιουργία εφαρμογών με διαφορετικές μεθόδους και αλγορίθμους για την κατηγοριοποίηση των δεδομένων. Αυτές θα μπορούσαν να περιλαμβάνουν τις Μεθόδους Προσαρμογής Αλγορίθμων (Algorithm Adaptation Methods), καθώς και τις Μεθόδους Συνόλου (Ensemble Methods), η χρήση των οποίων μπορεί να επιτευχθεί μέσω του scikit-multilearn. Με τη χρήση αυτών των τεχνικών, οι χρήστες θα μπορούν να εφαρμόζουν πιο σύνθετες προσεγγίσεις, βελτιώνοντας την ακρίβεια και την αποτελεσματικότητα των μοντέλων που δημιουργούν.

Βιβλιογραφία

- [1] G. Kesavaraj and S. Sukumaran, "A study on classification techniques in data mining," in *2013 Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT)*, pp. 1--7, 2013.
- [2] C. C. Aggarwal, *An Introduction to Data Mining*, pp. 1--26. Cham: Springer International Publishing, 2015.
- [3] A. Kumar, "Difference: Binary vs multiclass vs multilabel classification." <https://vitalflux.com/difference-binary-multi-class-multi-label-classification/>. Accessed: 13-08-2024.
- [4] G. Tsoumakas and I. Katakis, "Multi-label classification: An overview," *International Journal of Data Warehousing and Mining*, vol. 3, pp. 1--13, 09 2009.
- [5] S. Jain, "Solving multi-label classification problems (case studies included)." <https://www.analyticsvidhya.com/blog/2017/08/introduction-to-multi-label-classification/>. Accessed: 15-10-2024.
- [6] H. Jadhav, "Multiclass classification vs. multi-label classification." <https://harshjadhav100.medium.com/multiclass-classification-vs-multi-label-classification/>. Accessed: 31-07-2023.
- [7] Y. Chaurasia, "Multiclass classification vs multi-label classification." <https://www.geeksforgeeks.org/multiclass-classification-vs-multi-label-classification/>. Accessed: 06-01-2024.
- [8] W. Liu, H. Wang, X. Shen, and I. W. Tsang, "The emerging trends of multi-label learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 11, pp. 7955--7974, 2022.
- [9] H. J. Escalante, *Automated Machine Learning---A Brief Review at the End of the Early Years*, pp. 11--28. Cham: Springer International Publishing, 2021.
- [10] run:ai, "Automl: How it works." <https://www.run.ai/guides/automl>. Accessed: 2024.
- [11] Y.-W. Chen, Q. Song, and X. Hu, "Techniques for automated machine learning," *SIGKDD Explor. Newsl.*, vol. 22, p. 35--50, Jan. 2021.
- [12] Q. Yao, M. Wang, Y. Chen, W. Dai, Y.-F. Li, W.-W. Tu, Q. Yang, and Y. Yu, "Taking human out of learning applications: A survey on automated machine learning," *arXiv preprint arXiv:1810.13306*, vol. 31, 2018.

- [13] X. He, K. Zhao, and X. Chu, "Automl: A survey of the state-of-the-art," *Knowledge-Based Systems*, vol. 212, p. 106622, 2021.
- [14] K. Chauhan, S. Jani, D. Thakkar, R. Dave, J. Bhatia, S. Tanwar, and M. S. Obaidat, "Automated machine learning: The new wave of machine learning," in *2020 2nd International Conference on Innovative Mechanisms for Industry Applications (ICIMIA)*, pp. 205--212, 2020.
- [15] V. Kanade, "What is automated machine learning (automl)? meaning, types, and functions." <https://www.spiceworks.com/tech/artificial-intelligence/articles/what-is-automl>. Accessed: 05-09-2023.
- [16] R. Elshawi, M. Maher, and S. Sakr, "Automated machine learning: State-of-the-art and open challenges," *arXiv preprint arXiv:1906.02287*, 2019.
- [17] A. Danielkivych, "What is automl in machine learning? going beyond the basics." <https://forbytes.com/blog/what-is-automl/>. Accessed: 08-03-2024.
- [18] L. Kotthoff, C. Thornton, H. H. Hoos, F. Hutter, and K. Leyton-Brown, "Auto-weka 2.0: Automatic model selection and hyperparameter optimization in weka," *Journal of Machine Learning Research*, vol. 18, no. 25, pp. 1--5, 2017.
- [19] M. Feurer, K. Eggensperger, S. Falkner, M. Lindauer, and F. Hutter, "Auto-sklearn 2.0: Hands-free automl via meta-learning," 2022.
- [20] M.-A. VASILE, F. POP, M.-C. NIȚĂ, and V. CRISTEA, "Mlbox: Machine learning box for asymptotic scheduling," *Information Sciences*, vol. 433-434, pp. 401--416, 2018.
- [21] L. Jansen, "Evaluating automl performance of sagemaker autopilot and transmogrifai," 2022.
- [22] E. Bisong, *Google AutoML: Cloud Vision*, pp. 581--598. Berkeley, CA: Apress, 2019.
- [23] D. Mukunthu, P. Shah, and W. H. Tok, *Practical automated machine learning on Azure: using Azure machine learning to quickly build AI solutions*. O'Reilly Media, 2019.
- [24] V. Tidake and S. Sane, "Multi-label classification: a survey," *International Journal of Engineering & Technology*, vol. 7, p. 1045, 11 2018.
- [25] M.-L. Zhang, Y.-K. Li, X.-Y. Liu, and X. Geng, "Binary relevance for multi-label learning: an overview," *Frontiers of Computer Science*, vol. 12, pp. 191--202, 2018.
- [26] F. Wauthier, M. Jordan, and N. Jojic, "Efficient ranking from pairwise comparisons," in *Proceedings of the 30th International Conference on Machine Learning* (S. Dasgupta and D. McAllester, eds.), vol. 28 of *Proceedings of Machine Learning Research*, (Atlanta, Georgia, USA), pp. 109--117, PMLR, 17--19 Jun 2013.
- [27] K. Brinker and E. Hüllermeier, "Calibrated label-ranking," in *Proceedings of the NIPS-2005 Workshop on Learning to Rank*, pp. 1--6, Citeseer, 2005.
- [28] J. C. Junior, E. Faria, J. Silva, and R. Cerri, "Label powerset for multi-label data streams classification with concept drift," in *Proc. 5th Symp. Knowl. Discovery, Mining Learn*, pp. 97--104, 2017.
- [29] G. Tsoumakas, I. Katakis, and I. Vlahavas, "Random k-labelsets for multilabel classification," *IEEE Transactions on Knowledge and Data Engineering*, vol. 23, no. 7, pp. 1079--1089, 2011.

- [30] J. Read, B. Pfahringer, G. Holmes, and E. Frank, "Classifier chains for multi-label classification," in *Machine Learning and Knowledge Discovery in Databases* (W. Buntine, M. Grobelnik, D. Mladenić, and J. Shawe-Taylor, eds.), (Berlin, Heidelberg), pp. 254--269, Springer Berlin Heidelberg, 2009.
- [31] C. A. Tawiah and V. S. Sheng, "A study on multi-label classification," in *Advances in Data Mining. Applications and Theoretical Aspects* (P. Perner, ed.), (Berlin, Heidelberg), pp. 137--150, Springer Berlin Heidelberg, 2013.
- [32] E. A. Cherman, M. C. Monard, and J. Metz, "Multi-label problem transformation methods: a case study," *CLEI Electronic Journal*, vol. 14, no. 1, pp. 4--4, 2011.
- [33] S. N. Madaraboina, S. Sharma, S. Singh, and V. Kumar, "Efficient multi-target classification for bug priority and resolution time prediction," *Multimedia Tools and Applications*, pp. 1--30, 2024.
- [34] G. Tsoumakas, I. Katakis, and I. Vlahavas, *Mining Multi-label Data*, pp. 667--685. Boston, MA: Springer US, 2010.
- [35] M. M. ARAT, "Metrics for multilabel classification." https://mmuratarat.github.io/2020-01-25/multilabel_classification_metrics. Accessed: January 25-01-2020.
- [36] D. Ganda and R. Buch, "A survey on multi label classification," *Recent Trends in Programming Languages*, vol. 5, no. 1, pp. 19--23, 2018.
- [37] WixEncyclopedia, "Application programming interface (api)." <https://www.wix.com/encyclopedia/definition/application-programming-software-api>. Accessed: 10-11-2024.
- [38] MuleSoft, "What is an api (application programming interface)?" <https://www.mulesoft.com/api/what-is-an-api>. Accessed: 10-11-2024.
- [39] MuleSoft, "What is a restful api?" <https://www.mulesoft.com/api/rest/what-is-rest-api>. Accessed: 10-11-2024.
- [40] K. Chris, "What is php? the php programming language meaning explained." <https://www.freecodecamp.org/news/what-is-php-the-php-programming-language-meaning-explained/>. Accessed: 30-08-2024.
- [41] FlatCoding, "Php syntax." <https://flatcoding.com/tutorials/php-programming/php-syntax-guide-and-tags/>. Accessed: 21-10-2024.
- [42] PHP.net, "Php documentation." <https://www.php.net/manual/en>. Accessed: 14-01-2024.
- [43] J. Erickson, "Mysql: Understanding what it is and how it's used." <https://www.oracle.com/mysql/what-is-mysql/>. Accessed: 29-08-2024.
- [44] MySQL, "Mysql documentation." <https://dev.mysql.com/doc/refman/8.0/en/what-is-mysql.html>. Accessed: 01-11-2024.
- [45] A. Rawat, "A review on python programming," *International Journal of Research in Engineering, Science and Management*, vol. 3, p. 8--11, Dec. 2020.

-
- [46] Composer, "Composer documentation." <https://getcomposer.org/doc/00-intro.md>. Accessed: 16-11-2024.
- [47] D. Shcherbakov and I. Djuric, "Phpmailer guide: Configuration, smtp setup, and email sending." <https://mailtrap.io/blog/phpmailer/>. Accessed: 16-11-2024.
- [48] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, *et al.*, "Scikit-learn: Machine learning in python," *the Journal of machine Learning research*, vol. 12, pp. 2825--2830, 2011.
- [49] O. Kramer, *Scikit-Learn*, pp. 45--53. Cham: Springer International Publishing, 2016.
- [50] P. Szymański and T. Kajdanowicz, "A scikit-based python environment for performing multi-label classification," 2018.
- [51] M. D. N. M. W. Docs, "Html: Hypertext markup language." <https://developer.mozilla.org/en-US/docs/Web/HTML>. Accessed: 19-11-2024.
- [52] Investopedia, "Hypertext markup language (html): What it is and how it works." <https://www.investopedia.com/terms/h/html.asp>. Accessed: 19-11-2024.
- [53] M. D. N. M. W. Docs, "Css: Cascading style sheets." <https://developer.mozilla.org/en-US/docs/Web/CSS>. Accessed: 19-11-2024.
- [54] Javatpoint, "What is css." <https://www.javatpoint.com/what-is-css>. Accessed: 19-11-2024.
- [55] J. Alexandra, "What is bootstrap?." <https://www.hostinger.com/tutorials/what-is-bootstrap/>. Accessed: 19-05-2024.
- [56] A. Ouellette, "What is bootstrap: A beginner's guide." <https://careerfoundry.com/en/blog/web-development/what-is-bootstrap-a-beginners-guide/#what-is-bootstrap>. Accessed: 02-02-2023.
- [57] WixEncyclopedia, "Javascript." <https://www.wix.com/encyclopedia/definition/javascript>. Accessed: 21-11-2024.
- [58] JQuery, "Jquery documentation." <https://jquery.com>. Accessed: 21-11-2024.
- [59] Javatpoint, "What is jquery." <https://www.javatpoint.com/what-is-jquery>. Accessed: 21-11-2024.
- [60] HeidiSQL, "Heidisql documentation." <https://www.heidisql.com>. Accessed: 29-11-2024.