ΔΙΕΘΝΕΣ ΠΑΝΕΠΙΣΤΗΜΙΟ ΤΗΣ ΕΛΛΑΔΟΣ

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

HCvision
Διαδικτυακή εφαρμογή αυτοματοποιημένης μηχανικής μάθησης για απλούστερη ιεραρχική συσταδοποίηση

Φοιτητής:
Μωυσίδης Γεώργιος
Student ID: 174881

Επιβλέπων:
Ουγιάρογλου Στέφανος

29 January 2024

Τίτλος Π.Ε. AutoHierarchicalClusterer: Διαδικτυακή εφαρμογή αυτοματοποιημένης μηχανικής μάθησης για απλούστερη ιεραρχική συσταδοποίηση
Κωδικός Π.Ε. 23288
Ονοματεπώνυμο φοιτητή Μωυσίδης Γεώργιος
Ονοματεπώνυμο εισηγητή Ουγιάρογλου Στέφανος
Ημερομηνία ανάληψης Π.Ε. 27-10-2023
Ημερομηνία περάτωσης Π.Ε. 29-01-2024

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.

Αφιέρωση

Εκφράζω τις θερμότερες ευχαριστίες μου στον καθηγητή Στέφανο Ουγιάρογλου για την εμπιστοσύνη και την καθοδήγηση κατά τη διάρκεια της εργασίας. Εκφράζω επίσης βαθιά ευγνωμοσύνη για την ακλόνητη στήριξη της οικογένειάς μου κατά τη διάρκεια των σπουδών μου.

Πρόλογος

Στο σύγχρονο τοπίο, η σημασία της ανάλυσης δεδομένων και της εξαγωγής πληροφοριών είναι κρίσιμη για την πλοήγηση στις πολυπλοκότητες του κόσμου μας και την επίλυση πραγματικών προβλημάτων. Η πρόσβαση στην επιστήμη της μηχανικής μάθησης, ωστόσο, παραμένει πρόκληση για τους μη ειδικούς. Αυτή η εργασία αντιμετωπίζει αυτό το κενό αναπτύσσοντας μια εφαρμογή AutoML εστιασμένη στην ιεραρχική συσσωρευτική συσταδοποίηση, απαιτώντας ελάχιστες τεχνικές γνώσεις. Με τον τρόπο αυτό, διαμορφώντας την πρόσβαση σε προηγμένα εργαλεία μηχανικής μάθησης, ο στόχος είναι να εξουσιοδοτηθούν οι μη ειδικοί χρήστες, επιτρέποντάς τους να αποκομίσουν εργαλεία και να λάβουν ενημερωμένες αποφάσεις, προάγοντας έτσι μια πιο ενδελεχή και γνωστική κοινωνία.

Περίληψη

Η συσταδοποίηση, μια θεμελιώδης έννοια στην ανάλυση δεδομένων, περιλαμβάνει την ομαδοποίηση παρόμοιων σημείων δεδομένων με βάση ορισμένων κριτηρίων. Ένας ευρέως χρησιμοποιούμενος αλγόριθμος συσταδοποίησης είναι η ιεραρχική συσταδοποίηση, η οποία κατασκευάζει ένα δενδρόγραμμα συγχωνεύοντας επαναληπτικά σημεία δεδομένων σε συστάδες. Ο προσδιορισμός του αριθμού των συστάδων εξαρτάται από την κοπή του δενδρογράμματος σε ένα συγκεκριμένο ύψος για να προκύψουν διακριτές συστάδες. Παρά την επικράτηση της ιεραρχικής συσταδοποίησης, υπάρχει ένα αξιοσημείωτο κενό στα αυτοματοποιημένα εργαλεία για τον αποτελεσματικό προσδιορισμό τόσο του τύπου σύνδεσης όσο και του βέλτιστου αριθμού συστάδων. Αντιμετωπίζοντας αυτό το κενό, δημιουργήσαμε το HCvision, ένα εργαλείο που έχει σχεδιαστεί για την αυτοματοποίηση της ιεραρχικής συσταδοποίησης. Το HCvision απευθύνεται σε χρήστες που δεν έχουν γνώσεις μηχανικής μάθησης και εξόρυξης δεδομένων, όπως και το ότι δεν γνωρίζουν προγραμματισμό, χρήση βιβλιοθηκών και χρήση ειδικών εργαλείων που απαιτούν εγκατάσταση στον υπολογιστή. Με την απλοποίηση της διαδικασίας συσταδοποίησης και την παροχή ολοκληρωμένης ανάλυσης των αποτελεσμάτων, το HCvision επιτρέπει σε ανειδίκευτους χρήστες να εξάγουν σημαντικές πληροφορίες από τα δεδομένα τους, χωρίς την ανάγκη περαιτέρω τεχνικής εκπαίδευσης. Αξιοποιώντας τον μέγιστο συντελεστή ασυνέπειας, το HCvision προσδιορίζει αντικειμενικά το βέλτιστο ύψος για την κοπή του δενδρογράμματος, προσδιορίζοντας τόσο τον τύπο μέτρησης της απόστασης όσο και τον αριθμό συστάδων. Επιπλέον, το HCvision παρέχει μια ολοκληρωμένη ανάλυση μέσω δενδρογραμμάτων και γραφημάτων παράλληλων συντεταγμένων, προσφέροντας πολύτιμες γνώσεις για τις δομές ομαδοποίησης και διευκολύνοντας τις αναθέσεις συστάδων. Σε μια εποχή όπου κυριαρχούν τα δεδομένα, το HCvision αποτελεί κρίσιμη πηγή για την εξαγωγή ουσιαστικών πληροφοριών από πολύπλοκα σύνολα δεδομένων, γεφυρώνοντας το χάσμα μεταξύ της ιεραρχικής συσταδοποίησης και των αναγκών πρακτικής ανάλυσης δεδομένων.

Abstract

Clustering, a fundamental concept in data analysis, involves grouping similar data points together based on certain criteria. One widely used clustering algorithm is hierarchical clustering, which constructs a dendrogram by iteratively merging data points or clusters. The determination of the number of clusters hinges on cutting the dendrogram at a specific height to yield distinct clusters. Despite the prevalence of hierarchical clustering, a notable void exists in automated tools for efficiently identifying both the linkage type and optimal number of clusters. Addressing this gap, this study introduces HCvision, an innovative tool designed to automate agglomerative hierarchical clustering. Leveraging the Maximum Inconsistency Coefficient, HCvision objectively identifies the optimal height for cutting the dendrogram, determining both linkage and cluster count. Additionally, HCvision provides a comprehensive analysis through dendrogram visualizations and parallel coordinated plots, offering valuable insights into clustering structures and facilitating cluster assignments. In an era dominated by data, HCvision stands as a crucial resource for extracting meaningful information from complex datasets, bridging the gap between hierarchical clustering and practical data analysis needs.

Περιεχόμενα

Κατάλογος Σχημάτων

Κατάλογος Πινάκων

Κατάλογος Κώδικα

Κεφάλαιο 1

Εισαγωγή

1.1 Συσταδοποίηση Δεδομένων

Στον κόσμο της μηχανικής μάθησης, δύο βασικά παραδείγματα, η επιβλεπόμενη και η μη επιβλεπόμενη μάθηση, διαδραματίζουν κρίσιμο ρόλο στην ανάκληση νοηματικών συμπερασμάτων από τα δεδομένα. Η επιβλεπόμενη μάθηση περιλαμβάνει τη χρήση ετικεταρισμένων συνόλων δεδομένων, όπου κάθε σημείο δεδομένων συσχετίζεται με μια αντίστοιχη κλάση ή ετικέτα. Ο κύριος στόχος είναι η κατασκευή ενός προβλεπτικού μοντέλου που να είναι ικανό να κατηγοριοποιεί νέα, μη έγκυρα δεδομένα σε προκαθορισμένες κλάσεις. Αντίθετα, η μη επιβλεπόμενη μάθηση λειτουργεί σε μη ετικεταρισμένα σύνολα δεδομένων, με σκοπό να αποκαλύψει κρυφά πρότυπα ή δομές εντός των δεδομένων χωρίς προηγούμενη γνώση των ετικετών κλάσης[1].

Η Συσταδοποίηση, η οποία αποτελεί κεντρική τεχνική στην μη επιβλεπόμενη μάθηση, αποτελεί τον πυρήνα αυτής του της εργασίας. Στην ουσία της, η συσταδοποίηση είναι η διαδικασία της ομαδοποίησης των σημείων δεδομένων σε υποσύνολα, ή συστάδες, με βάση την ομοιότητα των χαρακτηριστικών τους[1]. Αντίθετα από την κατηγοριοποίηση, ο στόχος της συσταδοποίησης είναι περιγραφικός, επιδιώκοντας να εντοπίσει φυσικές ομάδες εντός των δεδομένων. Οι θεμελιώδη αρχές της συσταδοποίησης περιλαμβάνουν τη μεγιστοποίηση της ομοιότητας εντός των συστάδων, ταυτόχρονα με την ελαχιστοποίηση της ανομοιότητας μεταξύ τους. Τα στάδια της συσταδοποίησης περιλαμβάνουν την εξαγωγή και επιλογή χαρακτηριστικών, τον σχεδιασμό αλγορίθμου συσταδοποίησης, την αξιολόγηση του αποτελέσματος και την πρακτική ερμηνεία των αποτελεσμάτων.

Η σημασία της συσταδοποίησης έγκειται στην ικανότητά της να αναγνωρίζει και να οργανώνει αυτόματα τα δεδομένα σε νοηματικές κατηγορίες, συνεισφέροντας σε μια πιο αποτελεσματική αναπαράσταση του υποκείμενου πληθυσμού που εξετάζεται. Η συσταδοποίηση βρίσκει εφαρμογές σε διάφορους επιστημονικούς τομείς, από τη βιολογία και τη γενετική μέχρι τη μηχανική και πέρα. Οι αλγόριθμοι που χρησιμοποιούνται στη συσταδοποίηση συμμορφώνονται με τις αρχές της μεγιστοποίησης της εντός συστάδας ομοιότητας και της μειονεκτικότητας μεταξύ συστάδων, καθιστώντας την ένα ευέλικτο εργαλείο για την ανίχνευση προτύπων και την εξαγωγή γνώσης[1]. Το παρόν κεφάλαιο θα εξερευνήσει τις λεπτομέρειες της συσταδοποίησης, διευκρινίζοντας τις ορισμούς της, τις λειτουργίες της, καθώς και τις τεχνικές πτυχές της, συνεισφέροντας έτσι στην κατανόηση αυτής της καθοριστικής τεχνικής της μη επιβλεπόμενης μάθησης.

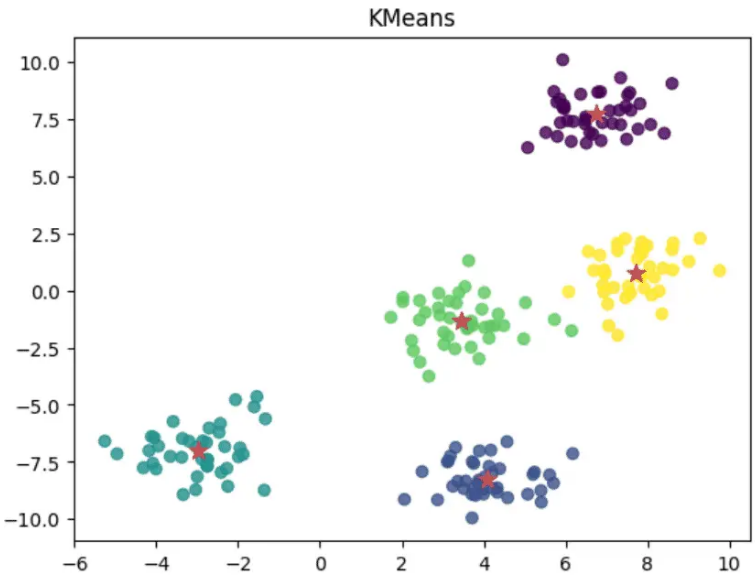
1.2 Κατηγορίες Αλγορίθμων Συσταδοποίησης

Οι μέθοδοι συσταδοποίησης διαδραματίζουν κεντρικό ρόλο στην οργάνωση των δεδομένων σε νοηματικές ομάδες, αποκαλύπτοντας πρότυπα και διευκολύνοντας την εξαγωγή γνώσης σε διάφορους τομείς. Μερικές από τις ποιο σημαντικές μεθόδους χωρίζονται στις παρακάτω κατηγορίες.

Συσταδοποίηση Με Βάση την Κατάτμηση
Η βασική ιδέα πίσω από τους αλγορίθμους συσταδοποίησης με βάση την κατάτμηση είναι να βελτιώνουν επαναληπτικά τον τρόπο ανάθεσης των σημείων δεδομένων σε συστάδες μετακινώντας αντικείμενα από μια ομάδα σε μια άλλη[2]. Δύο πολύ γνωστοί αλγόριθμοι αυτής της κατηγορίας είναι οι K-means και K-medoids.

Ο αλγόριθμος K-means ενημερώνει επαναληπτικά τα κέντρα των συστάδων, που αντιπροσωπεύονται από τα μέσα των σημείων δεδομένων εντός κάθε συστάδας, μέχρι να πληρούνται κάποια κριτήρια σύγκλισης. Πρόκειται για έναν ευρέως χρησιμοποιούμενο και ευέλικτο αλγόριθμο, κατάλληλο για ποικίλους τομείς προβλημάτων. Χρησιμοποιείται η έννοια του κεντροειδούς, που είναι το μέσο ή η μέση τιμή ενός ομίλου σημείων. Ωστόσο, σημειώνεται ότι το κεντροειδές μπορεί να μην αντιστοιχεί πάντα σε ένα πραγματικό σημείο δεδομένων. Στο σχήμα 1.1 παρατηρείται η γραφική παράσταση των συστάδων που δημιουργήθηκαν μαζί με τα κέντρα τους χρησιμοποιώντας των αλγόριθμο Kmeans[3].

Ο K-medoids, μια βελτίωση του K-means, σχεδιάστηκε για την αντιμετώπιση διακριτικών δεδομένων. Αντίθετα με το K-means, το K-medoids λαμβάνει το σημείο δεδομένων που βρίσκεται πιο κοντά στο κέντρο των σημείων ως τον εκπρόσωπο της αντίστοιχης συστάδας. Αυτό καθιστά τον K-medoids πιο ανθεκτικό σε περιπτώσεις όπου η μέση τιμή ή ο μέσος όρος δεν είναι κατάλληλα αντιπροσωπευτικά.



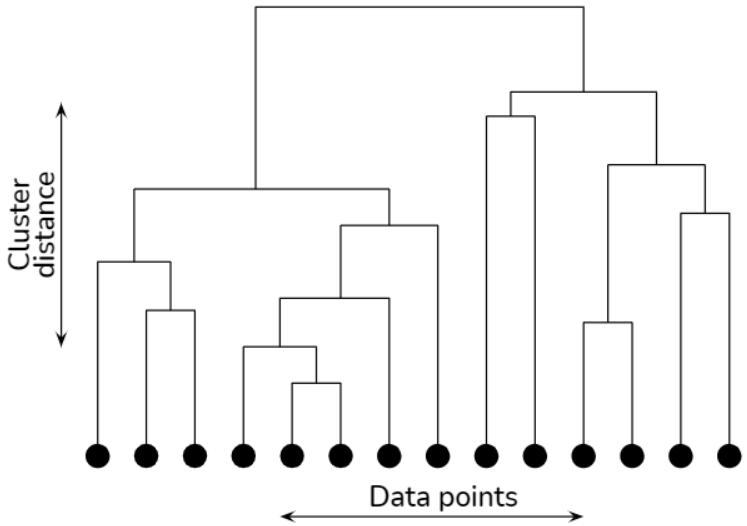
Σχήμα 1.1: Συσταδοποίηση με βάση την κατάτμηση

Άλλοι σημαντικοί αλγόριθμοι συσταδοποίησης με βάση την κατάτμηση περιλαμβάνουν τον PAM (Partitioning Around Medoids), τον CLARA (Clustering LARge Applications), και τον CLARANS (Clustering Large Applications based upon RANdomized Search). Ο PAM επικεντρώνεται στους μεσόδους αντί των μέσων, προσφέροντας καλύτερη ανθεκτικότητα στις ακραίες τιμές. Ο CLARA χρησιμοποιεί μια μεθοδολογία βασισμένη σε δείγματα για την αντιμετώπιση μεγάλων συνόλων δεδομένων, ενώ ο CLARANS συνδυάζει μια μέθοδο τύπου K-medoids με αναζήτηση βασισμένη σε τυχαίες επιλογές, ενισχύοντας τόσο την ποιότητα όσο και την επεκτασιμότητα.

Συνοπτικά, οι αλγόριθμοι συσταδοποίησης με βάση την κατάτμηση λειτουργούν με τον τρόπο της επαναληπτικής βελτίωσης της ανάθεσης των σημείων δεδομένων σε συστάδες, βασιζόμενοι στην έννοια ότι ένα κεντρικό σημείο μπορεί να αντιπροσωπεύει αποτελεσματικά μια συστάδα. Αυτοί οι αλγόριθμοι είναι εξαιρετικά χρήσιμοι σε διάφορες εφαρμογές, παρέχοντας ενδιαφέρουσες πληροφορίες για τη δομή των συνόλων δεδομένων.

Συσταδοποίηση με Βάση την Ιεραρχία

Οι ιεραρχικοί αλγόριθμοι συσταδοποίησης, όπως δηλώνει και το όνομά τους, δημιουργούν μια ιεραρχία εμφωλιασμένων συσταδοποιήσεων[4]. Δηλαδή, συστάδες περιέχουν μεμονωμένα στοιχεία και άλλες συστάδες, οι οποίες με τη σειρά τους μπορεί να περιέχουν και αυτές άλλες, μικρότερες συστάδες, δημιουργώντας έτσι τα επίπεδα της ιεραρχίας. Στο σχήμα 2.1 φαίνεται ένα δενδρόγραμμα ιεραρχικής συσταδοποίησης[5].



Σχήμα 1.2: Δενδρόγραμμα ιεραρχικής συσταδοποίησης

Οι ιεραρχικοί αλγόριθμοι διακρίνονται σε δύο υποκατηγορίες: τους συσσωρευτικούς και τους διαιρετικούς. Οι αλγόριθμοι μπορούν να αναπαρασταθούν πλήρως με δενδρογράμματα, δηλαδή με δενδρικά διαγράμματα, τα οποία παρουσιάζουν τη διάταξη των συστάδων που δημιουργήθηκαν από την ιεραρχική συσταδοποίηση. Ουσιαστικά, κάθε επίπεδο ενός δενδρογράμματος ορίζει ένα βήμα του αλγορίθμου.

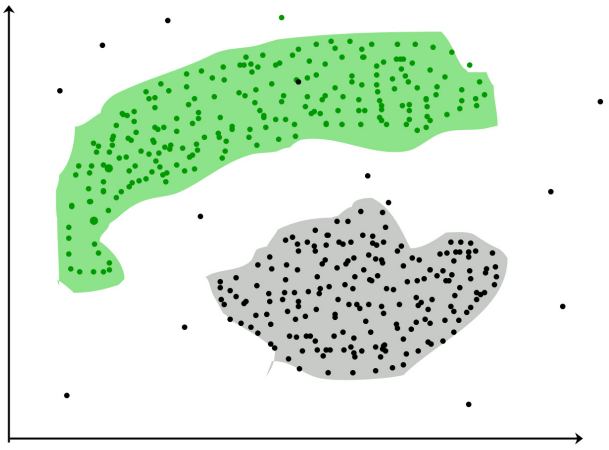
- Συσσωρευτική Ιεραρχία (Agglomerative Hierarchical Clustering): Στην συσσωρευτική ιεραρχία, ξεκινάμε με κάθε δείγμα σε ξεχωριστή συστάδα και στη συνέχεια συνενώνουμε τις πιο παρόμοιες συστάδες μέχρι να φτάσουμε σε μια μεγάλη συστάδα που περιέχει όλα τα δείγματα. Η διαδικασία επαναλαμβάνεται, συνενώνοντας συστάδες σε όλο και μεγαλύτερα επίπεδα, μέχρι να δημιουργηθεί το ιεραρχικό δέντρο[6].

- Διαιρετική Ιεραρχία (Divisive Hierarchical Clustering): Στην κάτω προς τα πάνω ιεραρχία, ξεκινάμε με μια μεγάλη συστάδα που περιέχει όλα τα δείγματα και διαιρούμε αυτήν την συστάδα σε μικρότερες και πιο παρόμοιες συστάδες. Η διαδικασία συνεχίζεται αποσπώντας υποσυστάδες σε όλο και μικρότερα επίπεδα, μέχρι να φτάσουμε στα φύλλα του ιεραρχικού δέντρου[6].

Το βασικό πλεονέκτημα των ιεραρχικών αλγορίθμων είναι ότι δεν χρειάζεται να υποθέσουμε ένα συγκεκριμένο αριθμό συστάδων, αφού οποιοσδήποτε αριθμός μπορεί να επιτευχθεί, απλά κόβοντας το δενδρόγραμμα στο κατάλληλο επίπεδο. Θα μιλήσουμε εκτενώς για τους αλγόριθμους αυτούς στο Κεφάλαιο 2.

Συσταδοποίηση με Βάση την Πυκνότητα

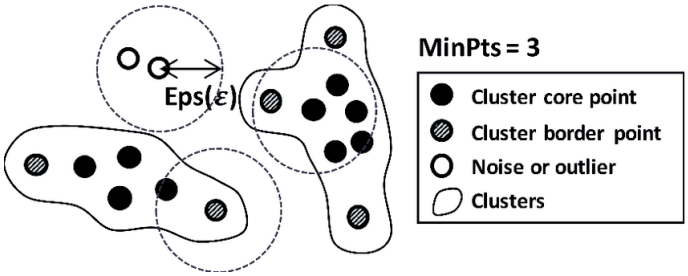
Οι αλγόριθμοι συσταδοποίησης βασισμένοι στην πυκνότητα λειτουργούν με βάση την ιδέα ότι τα δεδομένα που βρίσκονται σε περιοχές υψηλής πυκνότητας του χώρου θεωρούνται ότι ανήκουν στην ίδια συστάδα. Κλασικοί αλγόριθμοι αυτής της κατηγορίας περιλαμβάνουν τον DBSCAN, τον OPTICS και τον Mean-shift[4]. Στο Σχήμα 1.3 παρατηρούμαι συστάδες που δημιουργήθηκαν με βάση την πυκνότητα[7].



Σχήμα 1.3: Συσταδοποίηση με Βάση την Πυκνωτητα

Ο αλγόριθμος DBSCAN (Density-Based Spatial Clustering of Applications with Noise) είναι ο πιο γνωστός αλγόριθμος συσταδοποίησης βασισμένος στην πυκνότητα. Κατασκευάζει συστάδες εξετάζοντας την πυκνότητα των δεδομένων. Ένα σημείο θεωρείται πυκνό-προσεγγίσιμο (density-reachable) από ένα άλλο σημείο αν υπάρχει μονοπάτι από το ένα στο άλλο, τέτοιο ώστε κάθε σημείο του μονοπατιού είναι πυκνό-προσεγγίσιμο από το προηγούμενό του. Εάν ένα σημείο είναι πυκνό-προσεγγίσιμο από έναν ορισμένο αριθμό γειτονικών σημείων (κεντρικό σημείο), τότε σχηματίζει μια συστάδα. Σημεία που δεν είναι πυκνό-προσεγγίσιμα καλούνται ακραία. Παράδειγμα γραφικής αναπαράστασης του DBSCAN παρατηρείται στο σχήμα 1.4[8].

Η Χρονική Πολυπλοκότητα είναι συνήθως χαμηλή, αλλά εξαρτάται από την υλοποίηση και τη δομή των δεδομένων. Είναι αποτελεσματικός για ανίχνευση συστάδων υψηλής πυκνότητας, ευέλικτος ως προς το σχήμα των συστάδων. Αλλά είναι ευαίσθητος στις παραμέτρους, όπως ο αριθμός των γειτονικών σημείων, χαμηλή ποιότητα σε περιπτώσεις όπου η πυκνότητα δεν είναι ομοιόμορφη. Οι μέθοδοι βασισμένες στην πυκνότητα υποθέτουν ότι τα σημεία που ανήκουν σε κάθε συστάδα προέρχονται από μια συγκεκριμένη κατανομή πιθανότητας. Στόχος τους είναι να εντοπίσουν τις συστάδες και τις παραμέτρους της κατανομής. Είναι κα-

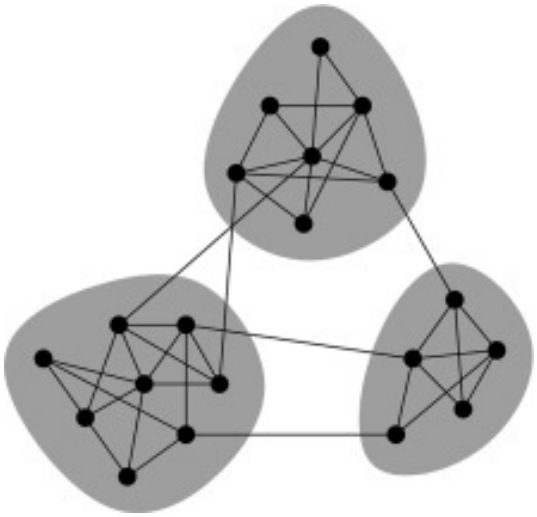


Σχήμα 1.4: Συσταδοποίηση με DBSCAN

τάλληλες για τον εντοπισμό συστάδων με αυθαίρετο σχήμα, που δεν είναι απαραίτητα κυρτές.

Συσταδοποίηση με Βάση τους Γράφους
Αυτοί οι αλγόριθμοι συσταδοποίησης βασίζονται στη θεωρία γράφων, όπου ο κόμβος θεωρείται ως το σημείο δεδομένων και η ακμή ως η σχέση μεταξύ των σημείων δεδομένων. Τυπικοί αλγόριθμοι αυτού του τύπου είναι ο CLICK και ο MST-based clustering[4].

Ο αλγόριθμος CLICK βασίζεται στην ελαχιστοποίηση του βάρους των ακμών στο γράφο με επαναληπτικό τρόπο προκειμένου να δημιουργήσει τις συστάδες. Ουσιαστικά, ο αλγόριθμος χωρίζει το γράφο σε συστάδες με βάση τη συνδεσιμότητα των κόμβων. Στο σχήμα 1.5 βλέπουμε διαχωρισμό του γράφου σε συστάδες[9].



Σχήμα 1.5: Συσταδοποίηση με Βάση τους Γράφους

Η χρονική πολυπλοκότητα χρόνου εξαρτάται από τον αριθμό των κόμβων (v) και των ακμών (e), καθώς και από την πολυπλοκότητα υπολογισμού ελάχιστης τομής (f(v, e)). Είναι αποτελεσματική στη συσταδοποίηση με υψηλή ακρίβεια. Αλλά η πολυπλοκότητα του χρόνου αυξάνεται δραματικά με την αύξηση της πολυπλοκότητας του γράφου.

Οι μέθοδοι βασισμένες στη θεωρία γράφων παράγουν συστάδες μέσω γράφων, όπου οι ακμές του γράφου συνδέουν τα αντικείμενα που αντιπροσωπεύονται ως κόμβοι. Ο αλγόριθμος MST (Ελάχιστο Συνδεδεμένο Δέντρο) είναι ένας γνωστός αλγόριθμος βασισμένος στη θεωρία γράφων, καθώς κατασκευάζει ένα δέντρο που συνδέει όλους τους κόμβους με το ελάχιστο δυνατό κόστος.

1.3 Αυτόματη Μηχανική Μάθηση (AutoML)

Η Αυτοματοποιημένη Μηχανική Μάθηση (AutoML) αποτελεί μια κεντρική πτυχή του τομέα της μηχανικής μάθησης, με στόχο την απλοποίηση και τον εκσυγχρονισμό της διαδικασίας ανάπτυξης μοντέλων. Καλύπτει μια σειρά από τεχνικές και εργαλεία που σχεδιάστηκαν για να αυτοματοποιήσουν ολοκληρωμένα τη διαδικασία κατασκευής μοντέλων μηχανικής μάθησης, από την προεπεξεργασία των δεδομένων και τη μηχανική εξαγωγή χαρακτηριστικών μέχρι την επιλογή μοντέλου, τη βελτιστοποίηση υπερπαραμέτρων και την τελική ερμηνεία του μοντέλου. Η ιστορία του AutoML εκτείνεται σε δεκαετίες, παρακολουθώντας την εξέλιξη διαφόρων βιβλιοθηκών και εργαλείων, όπως το Weka[10], το RapidMiner[11], το Scikit-learn[12] και το Spark MLlib[13], με σημαντικές συνεισφορές από ακαδημαϊκά ιδρύματα και startups[14].

Τα εργαλεία AutoML συνήθως ακολουθούν τρία περίπλοκα στάδια. Πρώτον, η προεπεξεργασία των δεδομένων και η μηχανική εξαγωγή χαρακτηριστικών περιλαμβάνει την αντιμετώπιση καθηκόντων, όπως η ανίχνευση τύπου και σχήματος των δεδομένων, με διαφορετικούς βαθμούς αυτοματοποίησης ανάμεσα στα εργαλεία. Για παράδειγμα, το TransmogrifAI[15] ξεχωρίζει στην υποστήριξη λεπτομερούς ανίχνευσης τύπου δεδομένων, ενώ άλλα, όπως το Auto-sklearn, βασίζονται περισσότερο στην είσοδο του χρήστη. Δεύτερον, στο στάδιο της επιλογής μοντέλου, της βελτιστοποίησης υπερπαραμέτρων και της αναζήτησης αρχιτεκτονικής, εκπαιδεύονται πολλά διαφορετικά μοντέλα, με κάθε εργαλείο να χρησιμοποιεί διάφορους αλγορίθμους, όπως η αναζήτηση πλέγματος, η τυχαία αναζήτηση και η βαϊανή βελτιστοποίηση[16].

Παραδείγματος χάρη, το Auto-Weka ήταν ένα από τα πρώτα εργαλεία AutoML που χρησιμοποιούσε αλγόριθμους από τη βιβλιοθήκη Weka. Προχωρώντας, τα εργαλεία AutoML έχουν συνεχώς τελειοποιηθεί για να αντιμετωπίσουν περιορισμούς στην προεπεξεργασία των δεδομένων και τη μηχανική εξαγωγή χαρακτηριστικών, με προόδους στην επιλογή μοντέλου και τεχνικές βελτιστοποίησης.

Τα πλεονεκτήματα των εργαλείων AutoML είναι πολύπλευρα. Μειώνουν σημαντικά τη χειρωνακτική προσπάθεια και την ειδίκευση που απαιτείται για την ανάπτυξη μοντέλων μηχανικής μάθησης, δημοκρατοποιώντας την πρόσβαση σε ισχυρή προβλεπτική ανάλυση. Τα εργαλεία AutoML επιτρέπουν την ταχύτερη ανάπτυξη μοντέλων, διευκολύνουν την αποτελεσματική χρήση υπολογιστικών πόρων μέσω της περικοπής του χώρου παραμέτρων και βελτιώνουν την ερμηνεία και την ανάλυση των προβλέψεων με λεπτομερείς απεικονίσεις και μετρικές σημασίας των χαρακτηριστικών. Επιπλέον, αυτά τα εργαλεία εξυπηρετούν τόσο τους επαγγελματίες όσο και τις νεοφυείς επιχειρήσεις, επιτρέποντας τους να αξιοποιούν τη δύναμη της μηχανικής μάθησης χωρίς εκτεταμένες γνώσεις στον τομέα. Κατά την ουσία, το AutoML αποτελεί μια μετασχηματιστική δύναμη στο να καθιστά τη μηχανική μάθηση προσβάσιμη, αποδοτική και ερμηνεύσιμη για ένα ευρύτερο φάσμα χρηστών[17].

Ο κύριος στόχος αυτής της εργασίας είναι η ανάπτυξη ενός εργαλείου AutoML που θα συμβάλει στην απλοποίηση και τον εκσυγχρονισμό της διαδικασίας ανάπτυξης μοντέλων μηχανικής μάθησης, εστιάζοντας ειδικά στην ιεραρχική συσταδοποίηση.

1.4 Κίνητρο και Συνεισφορά

Η ανάγκη για την ανάπτυξη της εφαρμογής HCvision πηγάζει από την αυξανόμενη πολυπλοκότητα των σύγχρονων αναλύσεων δεδομένων λόγω της εκθετικής αύξησης των διαθέσιμων δεδομένων. Η ανάγκη για αποτελεσματικούς και αυτοματοποιημένους αλγορίθμους είναι πιεστική. Στον τομέα της επιστήμης των δεδομένων και της αναλυτικής πληροφορικής, η διερεύνηση των δεδομένων απαιτεί προηγμένα εργαλεία για την εφαρμογή και τη χρήση αλγορίθμων μηχανικής μάθησης.

Ένας κύριος λόγος για την ανάγκη της εφαρμογής HCvision είναι η αναγκαία εγκατάσταση λογισμικού για την εκτέλεση του αλγορίθμου Ιεραρχικής Συσταδοποίησης. Ο αλγόριθμος αυτός, παρότι αποτελεσματικός στον εντοπισμό συστάδων, απαιτεί σημαντικούς υπολογιστικούς πόρους. Η εφαρμογή HCvision αντιμετωπίζει αυτήν την πρόκληση, προσφέροντας μια διεπαφή που επιτρέπει στους χρήστες να εφαρμόζουν τον αλγόριθμο χωρίς την ανάγκη περίπλοκων εγκαταστάσεων.
Ένα επιπλέον πλεονέκτημα της εφαρμογής HCvision είναι η προσφορά προτεινόμενων τιμών για τις παραμέτρους του αλγορίθμου, όπως ο τρόπος σύνδεσης (linkage) και ο αριθμός των συστάδων που πρόκειται να δημιουργηθούν. Αυτή η λειτουργία επιτρέπει στους χρήστες να ξεκινούν την ανάλυσή τους με βάση συνιστώμενες ρυθμίσεις, καθιστώντας τον αλγόριθμο προσιτό και για μη εξειδικευμένους χρήστες.

Σημειώνεται επίσης ότι η εφαρμογή δημιουργεί επιπλέον γραφικά στοιχεία, όπως το δέντρο συσταδοποίησης (dendrogram) και το παράλληλο διάγραμμα συντεταγμένων (parallel coordinates), ενώ παρέχει επίσης τις αναθέσεις στις συστάδες.

Η εφαρμογή HCvision συνεισφέρει στον τομέα της ανάλυσης δεδομένων προσφέροντας μια φιλική προς τον χρήστη διεπαφή και επιτρέποντας γρήγορη και αποτελεσματική εφαρμογή του αλγορίθμου Ιεραρχικής Συσταδοποίησης. Με την απλοποίηση της διαδικασίας εκτέλεσης του αλγορίθμου, ενθαρρύνει την ευρύτερη χρήση του σε διάφορες εφαρμογές και ενισχύει την πρόοδο στον τομέα της ανάλυσης δεδομένων.

Τέλος, το HCvision απευθύνεται σε χρήστες που διαθέτουν δεδομένα και επιθυμούν να τα αναλύσουν, χωρίς ωστόσο να διαθέτουν τις απαραίτητες γνώσεις στον τομέα της μηχανικής μάθησης και της εξόρυξης δεδομένων. Αυτοί οι χρήστες συνήθως δεν διαθέτουν επίσης τις δεξιότητες προγραμματισμού ή την εμπειρία στη χρήση βιβλιοθηκών και ειδικού λογισμικού που απαιτεί περίπλοκη εγκατάσταση και παραμετροποίηση, με πολλές φορές αυτό το λογισμικό να μην είναι διαθέσιμο δωρεάν.

1.5 Οργάνωση της εργασίας

Η δομή της εργασίας οργανώνεται στις παρακατω ενότητες, και σε αυτό το κείμενο θα παρουσιάσουμε μια επισκόπηση των κύριων ενοτήτων χωρίς λεπτομέρειες.

Στο Κεφάλαιο 2, προσεγγίζουμε το θέμα της Ιεραρχικής Συσταδοποίησης. Παρουσιάζουμε τον αλγόριθμο Ιεραρχικής Συσταδοποίησης, ο οποίος χρησιμοποιείται για την κατασκευή των συστάδων. Εξετάζουμε επίσης τη σημασία των παραμέτρων και τα πλεονεκτήματα/μειονεκτήματα του αλγορίθμου.

Στο Κεφάλαιο 3, εξετάζουμε την επιλογή τεχνολογιών, συμπεριλαμβανομένων των τεχνολογιών που χρησιμοποιούνται στον server και τον client για την υλοποίηση του HCvision.

Στο Κεφάλαιο 4, εστιάζουμε στον Σχεδιασμό και την Υλοποίηση του HCvision. Παρουσιάζουμε τις λειτουργικές απαιτήσεις, την αρχιτεκτονική, και την υλοποίηση του εξυπηρετητή και του πελάτη.

Στο Κεφάλαιο 5, παρουσιάζουμε το HCvision και αναλύουμε διάφορες σελίδες και λειτουργίες του, όπως την αρχική σελίδα, την εγγραφή χρήστη, τη σύνδεση, την επεξεργασία προσωπικών στοιχείων, τη διαγραφή λογαριασμού, και λειτουργίες που σχετίζονται με τη ιεραρχική συσταδοποίηση.

Στο Κεφάλαιο 6, αξιολογούμε το HCvision ως προς την απόδοση του συστήματος καθως και την εμπειρία χρήσης μέσω της μετρικής SUS.

Τέλος, στο Κεφάλαιο 7, παρουσιάζουμε τα συμπεράσματα της εργασίας και αναφέρουμε μελλοντικές επεκτάσεις και βελτιώσεις που μπορούν να εφαρμοστούν στο HCvision.

Κεφάλαιο 2

Ιεραρχική Συσταδοποίηση

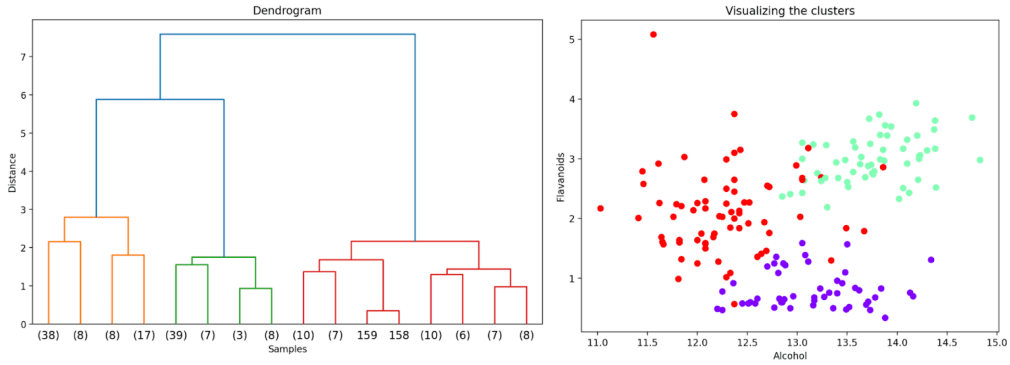
2.1 Εισαγωγή

Η ιεραρχική συσταδοποίηση είναι μια ισχυρή τεχνική που χρησιμοποιείται στην ανάλυση δεδομένων και την αναγνώριση προτύπων, η οποία έχει σχεδιαστεί για να αποκαλύπτει την εγγενή δομή σε σύνολα δεδομένων οργανώνοντας τα σημεία δεδομένων σε μια ιεραρχική δομή που μοιάζει με δέντρο. Σε αντίθεση με άλλες μεθόδους συσταδοποίησης, η ιεραρχική συσταδοποίηση παρέχει μια λεπτομερή αναπαράσταση των σχέσεων μεταξύ των σημείων δεδομένων, επιτρέποντας την ολοκληρωμένη κατανόηση της υποκείμενης δομής.

Η ιεραρχική συσταδοποίηση μπορεί να κατηγοριοποιηθεί σε γενικές γραμμές σε δύο κύριες προσεγγίσεις: τη συσσωρευτική και τη διαχωριστική. Η συσσωρευτική προσέγγιση, γνωστή και ως συσταδοποίηση από κάτω προς τα πάνω, ξεκινά με μεμονωμένα σημεία δεδομένων ως ξεχωριστές συστάδες και συγχωνεύει επαναληπτικά τις πλησιέστερες συστάδες έως ότου όλα τα σημεία δεδομένων ανήκουν σε μια ενιαία συστάδα[18]. Από την άλλη πλευρά, η διαιρετική προσέγγιση ή συσταδοποίηση από πάνω προς τα κάτω, ξεκινάει με όλα τα σημεία δεδομένων σε μια ενιαία συστάδα και τα διαιρεί αναδρομικά σε μικρότερες συστάδες μέχρι κάθε σημείο δεδομένων να σχηματίσει τη δική του συστάδα.

Η διαδικασία της ιεραρχικής συσταδοποίησης βασίζεται στην έννοια της εγγύτητας ή της ανομοιότητας μεταξύ των σημείων δεδομένων. Η επιλογή της κατάλληλης μετρικής εγγύτητας και της μεθόδου σύνδεσης επηρεάζει σημαντικά τις προκύπτουσες συστάδες[19]. Οι μετρικές εγγύτητας, όπως η ευκλείδεια απόσταση ή οι συντελεστές συσχέτισης, ποσοτικοποιούν τη διαφορετικότητα μεταξύ σημείων δεδομένων. Η μέθοδος σύνδεσης καθορίζει στη συνέχεια τον τρόπο με τον οποίο οι συστάδες συγχωνεύονται ή χωρίζονται με βάση αυτές τις ανομοιότητες.

Η ιεραρχική συσταδοποίηση προσφέρει επιπλέον μια οπτική αναπαράσταση των σχέσεων μέσω δενδρογραμμάτων. Ένα δενδρόγραμμα είναι ένα δενδροδιάγραμμα που απεικονίζει τη σταδιακή συγχώνευση ή διαίρεση των συστάδων, παρέχοντας ένα πολύτιμο εργαλείο για τους ερευνητές για την ερμηνεία της ιεραρχικής δομής μέσα στα δεδομένα[20]. Στο σχήμα 2.1 παρατηρούμαι ένα τυπικό δενδρόγραμμα καθώς και την αντίστοιχη γραφική απεικόνιση του συνόλου δεδομένων στο επίπεδο.



Σχήμα 2.1: Δενδρόγραμμα ιεραρχικής συσταδοποίησης

Η ευελιξία της ιεραρχικής συσταδοποίησης την καθιστά εφαρμόσιμη σε διάφορους τομείς, όπως η βιολογία, τα οικονομικά, οι κοινωνικές επιστήμες και η ανάλυση εικόνας. Η ικανότητά της να αποκαλύπτει ιεραρχικές σχέσεις και τη λεπτομερή δομή των συνόλων δεδομένων την καθιστά πολύτιμο εργαλείο για τη διερευνητική ανάλυση δεδομένων και την αναγνώριση προτύπων σε διάφορους τομείς[21].

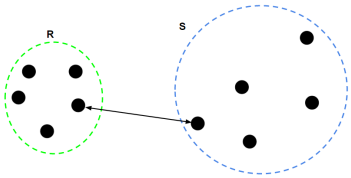
Συμπερασματικά, η ιεραρχική συσταδοποίηση αποτελεί μια ευέλικτη και διορατική μέθοδο για την κατανόηση της δομής που ενυπάρχει στα σύνολα δεδομένων. Η προσαρμοστικότητά της σε διαφορετικούς τομείς, σε συνδυασμό με την ικανότητα παροχής λεπτομερούς ιεραρχικής αναπαράστασης, την καθιστά θεμελιώδες εργαλείο στην εργαλειοθήκη του επιστήμονα δεδομένων.

2.2 Μέθοδος Σύνδεσης (Linkage)

Η σύνδεση παίζει καθοριστικό ρόλο στην ιεραρχική συσταδοποίηση, επηρεάζοντας τον τρόπο μέτρησης της ανομοιότητας ή της εγγύτητας μεταξύ των συστάδων κατά τη διαδικασία συσταδοποίησης. Η επιλογή της μεθόδου σύνδεσης επηρεάζει σημαντικά την προκύπτουσα ιεραρχική δομή και η κατανόηση αυτών των μεθόδων είναι απαραίτητη για την ουσιαστική ερμηνεία των αποτελεσμάτων της συσταδοποίησης.

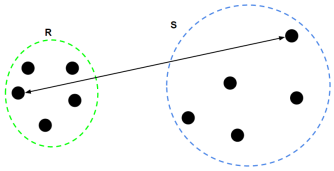
Απλή σύνδεση

Η μέθοδος της απλής σύνδεσης (single-linkage), γνωστή και ως μέθοδος του πλησιέστερου γείτονα, υπολογίζει την απόσταση μεταξύ δύο συστάδων με βάση την ελάχιστη απόσταση μεταξύ δύο οποιωνδήποτε σημείων δεδομένων από διαφορετικές συστάδες(Σχήμα 2.2). Αυτή η μέθοδος τείνει να σχηματίζει επιμήκεις συστάδες και είναι ευαίσθητη στις ακραίες τιμές λόγω του φαινομένου της ”αλυσίδας”[22].



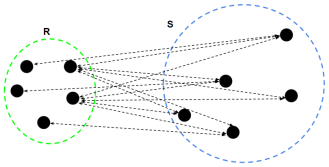
Σχήμα 2.2: Απλή Σύνδεση

Πλήρης σύνδεση
Η μέθοδος της πλήρους σύνδεσης (Complete-linkage), που ονομάζεται επίσης μέθοδος του πιο απομακρυσμένου γείτονα, υπολογίζει την απόσταση μεταξύ δύο συστάδων ως τη μέγιστη απόσταση μεταξύ δύο οποιωνδήποτε σημείων δεδομένων από διαφορετικές συστάδες(Σχήμα 2.3). Οι συστάδες που σχηματίζονται με πλήρη σύνδεση τείνουν να είναι πιο συμπαγείς και σφαιρικές, γεγονός που την καθιστά λιγότερο ευαίσθητη στις ακραίες τιμές σε σύγκριση με την απλή σύνδεση[23].



Σχήμα 2.3: Πλήρης σύνδεση

Σύνδεση μέσου όρου
Ο μέθοδος της σύνδεσης μέσου όρου υπολογίζει την απόσταση μεταξύ δύο συστάδων ως τη μέση απόσταση μεταξύ όλων των ζευγών σημείων δεδομένων από διαφορετικές συστάδες(Σχήμα 2.4). Στοχεύει στην επίτευξη ισορροπίας μεταξύ των τάσεων της απλής σύνδεσης και της πλήρους σύνδεσης, παράγοντας συστάδες που επηρεάζονται λιγότερο από τις ακραίες τιμές[24].



Σχήμα 2.4: Σύνδεση μέσου όρου

Σύνδεση Ward
Η μέθοδος σύνδεσης ward, επίσης γνωστή ως μέθοδος ελάχιστης διακύμανσης, ελαχιστοποιεί τη διακύμανση εντός κάθε συστάδας όταν αποφασίζει πώς θα συγχωνευθεί. Υπολογίζει την αύξηση της διακύμανσης που προκύπτει από τη συγχώνευση δύο συστάδων και επιλέγει τη συγχώνευση που επηρεάζει ελάχιστα τη συνολική διακύμανση[25].

2.3 Μετρικές Αποστάσεων (Distance Metrics)

Αφού καθοριστεί η κατάλληλη μέθοδος σύνδεσης, το επόμενο βήμα περιλαμβάνει τον υπολογισμό των αποστάσεων μεταξύ των συστάδων, ο οποίος απαιτεί την επιλογή μιας κατάλληλης

μετρικής απόστασης. Αυτή η μετρική παίζει καθοριστικό ρόλο στην ποσοτικοποίηση της ανομοιότητας ή της ομοιότητας μεταξύ μεμονωμένων σημείων δεδομένων, επιτρέποντας έτσι στον αλγόριθμο ιεραρχικής συσταδοποίησης να συγχωνεύει αποτελεσματικά τις συστάδες με βάση την εγγύτητά τους στο χώρο των χαρακτηριστικών. Μερικές από τις ποιο διαδεδομένες μετρικές είναι[26]:

Ευκλείδεια Απόσταση
Η ευκλείδεια απόσταση είναι ίσως η πιο διαδεδομένη μετρική στην ανάλυση δεδομένων, ειδικά σε γεωμετρικά προβλήματα. Υπολογίζει την απόσταση μεταξύ δύο σημείων σε ένα πολυδιάστατο χώρο. Η ευκλείδεια απόσταση μεταξύ δύο σημείων X_i και X_j σε έναν k-διάστατο χώρο δίνεται από τον τύπο:

$$D(X_i, X_j) = \sqrt{\sum_{k=1}^{n}(X_{ik} - X_{jk})^2}$$

όπου X_{ik} και X_{jk} αντιπροσωπεύουν την kη διάσταση των σημείων X_i και X_j αντίστοιχα.

Απόσταση Manhattan
Η απόσταση Manhattan, επίσης γνωστή ως απόσταση τετραγώνου πόλης ή απόσταση ταξί, υπολογίζει τις απόλυτες διαφορές μεταξύ των συντεταγμένων ενός ζεύγους αντικειμένων δεδομένων. Είναι ιδιαίτερα χρήσιμη όταν ασχολούμαστε με δομές πλέγματος. Η απόσταση Manhattan μεταξύ δύο σημείων X_i και X_j υπολογίζεται ως:

$$D_{Manhattan}(X_i, X_j) = \sum_{k=1}^{n} |X_{ik} - X_{jk}|$$

Απόσταση Minkowski
Η απόσταση Minkowski είναι μια γενικευμένη μετρική που περιλαμβάνει τόσο την ευκλείδεια όσο και την απόσταση Manhattan ως ειδικές περιπτώσεις. Ορίζεται ως:

$$D_{Minkowski}(X_i, X_j) = \left(\sum_{k=1}^{n} |X_{ik} - X_{jk}|^p\right)^{\frac{1}{p}}$$

όπου p είναι ένας παράμετρος που καθορίζει τη συγκεκριμένη μετρική απόσταση. Όταν $p = 2$, η απόσταση Minkowski μειώνεται στην ευκλείδεια απόσταση, ενώ $p = 1$ αντιστοιχεί στην απόσταση Manhattan.

Απόσταση Cosine
Η απόσταση Cosine μετρά τον συνημιτόνο της γωνίας μεταξύ δύο διανυσμάτων, παρέχοντας μια μέτρηση ομοιότητας αντί γεωμετρικής απόστασης. Χρησιμοποιείται συχνά στην ανάλυση κειμένου και σε δεδομένα υψηλής διάστασης. Η απόσταση Cosine μεταξύ των διανυσμάτων A και B δίνεται από τον τύπο:

$$\theta = \arccos\left(\frac{A \cdot B}{\|A\|\|B\|}\right)$$

όπου θ είναι η γωνία μεταξύ των διανυσμάτων, και $A \cdot B$ αναπαριστά το εσωτερικό γινόμενο των διανυσμάτων A και B.

Στην εφαρμογή μας, επιλέγουμε να χρησιμοποιήσουμε τη μετρική της ευκλείδειας απόστασης λόγω της ευρείας της αποδοχής και της εφαρμογής της σε διάφορους τομείς. Επιπλέον, αξίζει να σημειωθεί ότι αυτή η μετρική προσφέρει εξαιρετικά ακριβή αποτελέσματα σε διαφορετικά σύνολα δεδομένων και είναι ευρέως αποδεκτή από την επιστημονική κοινότητα της μηχανικής μάθησης.

2.4 Αλγόριθμος Συσσωρευτικής Ιεραρχικής Συσταδοποίησης

Η συσσωρευτική ιεραρχική συσταδοποίηση είναι μια επαναληπτική, από κάτω προς τα πάνω προσέγγιση για το σχηματισμό συστάδων, όπου τα μεμονωμένα σημεία δεδομένων συγχωνεύονται σταδιακά σε μεγαλύτερες συστάδες[18]. Ο αλγόριθμος προχωρά ως εξής:

1. Αρχικοποίηση: Αρχικά, κάθε σημείο δεδομένων αντιμετωπίζεται ως μοναδική συστάδα

2. Υπολογισμός απόστασης ανά ζεύγη: Υπολογισμός της ασυμφωνίας ανά ζεύγη ή της απόστασης μεταξύ όλων των συστάδων. Μπορούν να χρησιμοποιηθούν διάφορες μετρικές απόστασης, όπως η ευκλείδεια απόσταση ή η συσχέτιση, ανάλογα με τη φύση των δεδομένων.

3. Επιλογή ζευγών συστάδων: Προσδιορισμός των δύο συστάδων με τη μικρότερη ανομοιότητα. Η επιλογή του μέτρου ανομοιότητας και της μεθόδου σύνδεσης καθορίζει τον τρόπο υπολογισμού αυτής της ανομοιότητας.

4. Συστάδες συγχώνευσης: Συνδυάζει τις επιλεγμένες συστάδες σε μια νέα, μεγαλύτερη συστάδα. Αυτό μειώνει τον συνολικό αριθμό των συστάδων κατά μία.

5. Ενημέρωση του πίνακα αποστάσεων: Υπολογίστε εκ νέου τον πίνακα ανομοιότητας για να αντικατοπτρίζει τη συγχώνευση των συστάδων. Πρέπει να προσδιοριστεί η απόσταση μεταξύ της νέας συστάδας που δημιουργήθηκε και των υπόλοιπων συστάδων.

6. Επαναληπτική διαδικασία: Επαναλάβετε τα βήματα 1 έως 5 επαναληπτικά έως ότου απομείνει μόνο μία συστάδα που περιέχει όλα τα σημεία δεδομένων. Σε κάθε επανάληψη, ενημερώνεται ο πίνακας ανομοιότητας και συγχωνεύονται οι δύο πιο παρόμοιες συστάδες.

7. Κατασκευή δενδρογράμματος: Καθ' όλη τη διάρκεια της διαδικασίας, παρακολουθείτε την ιεραρχία των συγχωνεύσεων συστάδων για την κατασκευή ενός δενδρογράμματος. Το δενδρόγραμμα αναπαριστά οπτικά τη σειρά και το ύψος στο οποίο συγχωνεύονται οι συστάδες, παρέχοντας πληροφορίες για την ιεραρχική δομή των δεδομένων.

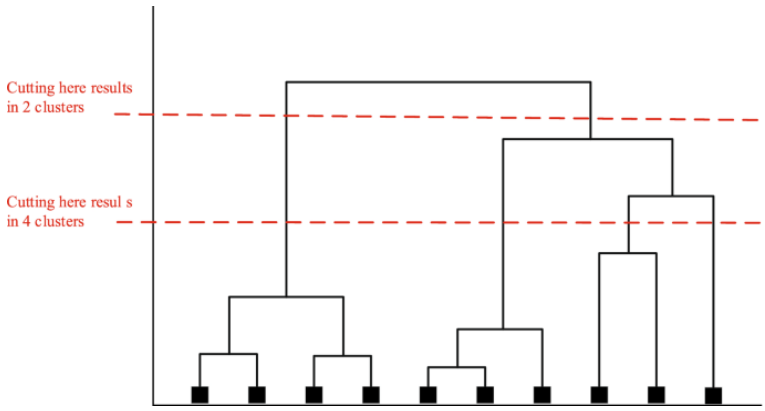
8. Κριτήριο διακοπής: Τερματισμός του αλγορίθμου όταν επιτευχθεί ο επιθυμητός αριθμός συστάδων ή όταν ικανοποιηθεί ένα συγκεκριμένο κριτήριο, όπως ένα προκαθορισμένο κατώφλι ανομοιότητας.

2.5 Επιλογή Μεθόδου Σύνδεσης και Συστάδων

Η ιεραρχική συσταδοποίηση παρέχει ένα πολύτιμο πλαίσιο για την αποκάλυψη μοτίβων μέσα σε σύνολα δεδομένων, αλλά ο καθορισμός του βέλτιστου αριθμού συστάδων και η επιλογή της κατάλληλης μεθόδου σύνδεσης είναι κρίσιμες αποφάσεις που επηρεάζουν σημαντικά τα αποτελέσματα. Το δενδρόγραμμα, μια δενδροειδής απεικόνιση, διαδραματίζει κεντρικό ρόλο στην καθοδήγηση αυτών των αποφάσεων.

Δενδρόγραμμα και εξαγωγή συστάδων

Το δενδρόγραμμα είναι μια γραφική αναπαράσταση των ιεραρχικών σχέσεων μεταξύ των συστάδων στο σύνολο δεδομένων. Σε αυτή τη δενδρική δομή, κάθε φύλλο αναπαριστά ένα μεμονωμένο σημείο δεδομένων, και καθώς κινείται κανείς προς τα πάνω στο δέντρο, οι συστάδες συγχωνεύονται διαδοχικά. Οι κάθετες γραμμές στο δενδρόγραμμα υποδεικνύουν την απόσταση ή τη διαφορετικότητα στην οποία συγχωνεύονται οι συστάδες. Για να προσδιοριστεί ο αριθμός των συστάδων, εξετάζονται οι κάθετες γραμμές και προσδιορίζεται το ύψος στο οποίο πραγματοποιείται η συγχώνευση(Σχήμα 2.5). Ο αριθμός των κάθετων γραμμών που τέμνονται από μια οριζόντια γραμμή που χαράσσεται σε ένα συγκεκριμένο ύψος αντιστοιχεί στον αριθμό των συστάδων[20].



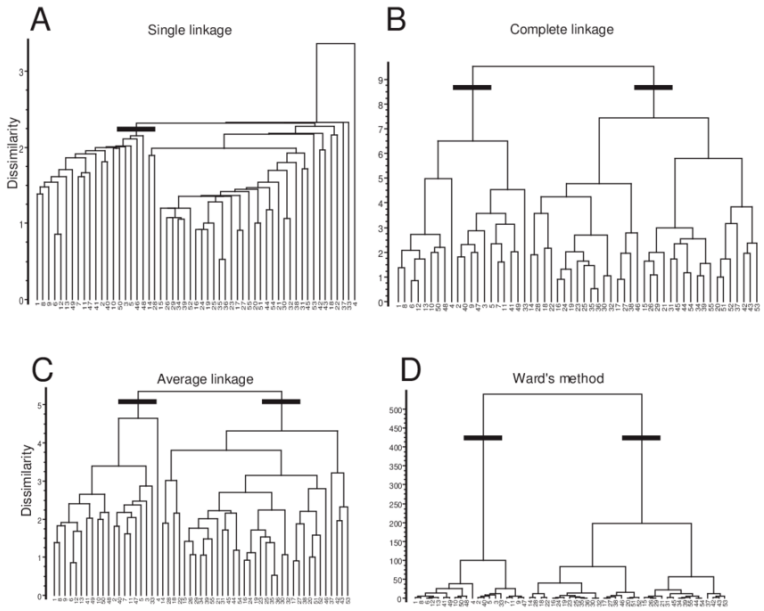
Σχήμα 2.5: Εξαγωγή πλήθους συστάδων από δενδρόγραμμα

Επιλογή μεθόδου σύνδεσης από δενδρόγραμμα

Η επιλογή της μεθόδου σύνδεσης επηρεάζει σε μεγάλο βαθμό το σχήμα του δενδρογράμματος και, κατά συνέπεια, τις προκύπτουσες συστάδες. Όπως αναφέρθηκε προηγουμένως οι συνήθεις μέθοδοι σύνδεσης περιλαμβάνουν την απλή σύνδεση (single-linkage), την πλήρη σύνδεση (complete-linkage), τη μέση σύνδεση (average-linkage) και τη σύνδεση Ward (Ward-linkage). Η απλή σύνδεση τείνει να δημιουργεί επιμήκεις συστάδες, η πλήρης σύνδεση παράγει πιο συμπαγείς συστάδες, η μέση σύνδεση επιτυγχάνει ισορροπία και η σύνδεση Ward ελαχιστοποιεί τη διακύμανση εντός των συστάδων. Η επιλογή της καταλληλότερης μεθόδου σύνδεσης εξαρτάται από τα χαρακτηριστικά των δεδομένων και τους στόχους της ανάλυσης[18]. Για παράδειγμα, η απλή σύνδεση μπορεί να είναι κατάλληλη για την ανίχνευση επιμήκων δομών, ενώ η πλήρης σύνδεση μπορεί να προτιμάται για πιο συμπαγείς, σφαιρικές συστάδες. Στο παρακάτω σχήμα 2.6 παρατηρούνται τα δενδρογράμματα που παράγονται με την επιλογή διαφορετικής σύνδεσης.

Σκέψεις στην ανάλυση δενδρογράμματος

Ενώ η ανάλυση δενδρογράμματος είναι ισχυρή, απαιτεί προσεκτική ερμηνεία. Οι ερευνητές



Σχήμα 2.6: Παραγόμενο Δενδρόγραμμα βάση μεθόδου σύνδεσης

πρέπει να λαμβάνουν υπόψη το πλαίσιο των δεδομένων και τους στόχους της ανάλυσης. Η επιλογή του ύψους κοπής του δενδρογράμματος περιλαμβάνει ένα συμβιβασμό: υψηλότερες κοπές οδηγούν σε λιγότερες συστάδες με μεγαλύτερες διακυμάνσεις εντός της συστάδας, ενώ χαμηλότερες κοπές οδηγούν σε περισσότερες συστάδες με δυνητικά μικρότερες διακυμάνσεις εντός της συστάδας[27]. Η κρίση του εμπειρογνώμονα και η γνώση του τομέα είναι ζωτικής σημασίας για την επίτευξη μιας ουσιαστικής και ερμηνεύσιμης λύσης συσταδοποίησης.

Συμπερασματικά, η διαδικασία επιλογής της σύνδεσης και του αριθμού των συστάδων στην ιεραρχική συσταδοποίηση περιλαμβάνει την εξαγωγή πληροφοριών από το δενδρόγραμμα, την εξέταση της φύσης των δεδομένων και τη χρήση αυτοματοποιημένων τεχνικών για αυξημένη αντικειμενικότητα. Οι αποφάσεις αυτές θα πρέπει να λαμβάνονται με προσεκτική εξέταση του πλαισίου και των στόχων, διασφαλίζοντας ότι οι προκύπτουσες συστάδες ευθυγραμμίζονται με τα υποκείμενα μοτίβα των δεδομένων.

2.6 Αυτοματοποιημένη Επιλογή Μεθόδου Σύνδεσης και Πλήθους Συστάδων

Για να προτείνουμε τη βέλτιστη μέθοδο σύνδεσης και τον αριθμό των συστάδων, η μεθοδολογία μας αξιοποιεί την έννοια της ασυνέπειας στην ιεραρχική συσταδοποίηση. Οι συντελεστές ασυνέπειας, που μετρούν τη μεταβλητότητα των αποστάσεων μεταξύ συστάδων σε διαφορετικά επίπεδα δενδρογράμματος, καθοδηγούν τη διαδικασία επιλογής. Μια νέα μέθοδος για την εύρεση του βέλτιστου αριθμού συστάδων χρησιμοποιεί τους συντελεστές ασυνέπειας για να προσδιορίσει το σημείο όπου η προσθήκη περισσότερων συστάδων παύει να παρέχει σημαντική βελτίωση.

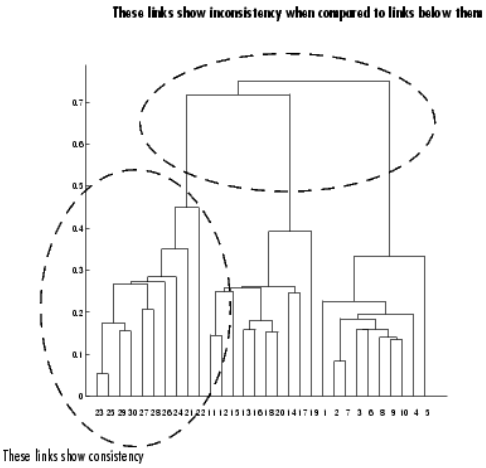
Στην ιεραρχική συσταδοποίηση, ο συντελεστής ασυνέπειας ποσοτικοποιεί τη μεταβλητότητα των αποστάσεων μεταξύ των συστάδων σε διαφορετικά επίπεδα δενδρογράμματος. Υπολογιζόμενος ως ο λόγος της διαφοράς μεταξύ των αποστάσεων σε δύο διαδοχικά επίπεδα προς τη μέση απόσταση, οι υψηλότεροι συντελεστές ασυνέπειας υποδηλώνουν μεγαλύτερη μετα-

βλητότητα, υποδηλώνοντας διακριτές συστάδες. Αυτή η μετρική βοηθά στον εντοπισμό των βέλτιστων σημείων αποκοπής στο δενδρόγραμμα όπου συμβαίνουν σημαντικές διαρθρωτικές αλλαγές.[28] Ο συντελεστής ασυνέπειας υπολογίζεται ως εξής:

$$I(c) = \frac{h(c) - \overline{h(c)}}{\sigma(c)}$$

όπου:
$I(c)$ είναι ο συντελεστής ασυνέπειας για το επίπεδο σύνδεσης c,
$h(c)$ είναι το ύψος σύνδεσης στο επίπεδο c,
$\overline{h(c)}$ είναι το μέσο ύψος σύνδεσης στο επίπεδο c,
$\sigma(c)$ είναι η τυπική απόκλιση των υψών σύνδεσης στο επίπεδο c.

Η επιλογή της βέλτιστης μεθόδου σύνδεσης καθοδηγείται από την αξιολόγηση των δενδρογραμμάτων και των σχετικών συντελεστών ασυνέπειας. Η μέθοδος σύνδεσης που δίνει συστάδες με τον μεγαλύτερο αριθμό συντελεστή ασυνέπειας προσδιορίζεται ως βέλτιστη. Αυτή η σχολαστική αξιολόγηση εξασφαλίζει μια διαδικασία επιλογής με γνώμονα τα δεδομένα, με αποτέλεσμα τη σύσταση της καταλληλότερης μεθόδου σύνδεσης για ένα δεδομένο σύνολο δεδομένων. Στο σχήμα 2.7 είναι εμφανές ότι ασυνέπεια μεταξύ συστάδων παρατηρείται σε συγχωνεύσεις των οποίων οι ακμές παρουσιάζουν μεγαλύτερο ύψος σε σχέση με συγχώνευσής προηγουμένων συστάδων.



Σχήμα 2.7: Απεικόνιση ακμών που υποδηλώνουν ασυνέπεια

Μια εξίσου κρίσιμη πτυχή είναι ο καθορισμός του βέλτιστου αριθμού συστάδων. Η διαδικασία αυτή περιλαμβάνει μια προσέγγιση με βάση τα δεδομένα, αξιοποιώντας τους συντελεστές ασυνέπειας που προκύπτουν από την ιεραρχική δομή συσταδοποίησης. Ο στόχος είναι να προσδιοριστεί ο αριθμός των συστάδων που παρέχουν ουσιαστικό διαχωρισμό εντός των δεδομένων. Διακρίνοντας το σημείο στο οποίο οι πρόσθετες συστάδες παύουν να προσφέρουν σημαντικά οφέλη, προτείνουμε έναν συγκεκριμένο αριθμό συστάδων ευθυγραμμισμένο με τα εγγενή πρότυπα που υπάρχουν στο σύνολο δεδομένων. Η προσέγγιση αυτή διασφαλίζει ότι η διαδικασία ιεραρχικής συσταδοποίησης όχι μόνο αποτυπώνει την υποκείμενη δομή των δεδομένων, αλλά και προσφέρει στους χρήστες πολύτιμες πληροφορίες σχετικά με τον κατάλληλο αριθμό συστάδων για τις συγκεκριμένες αναλυτικές ανάγκες τους.

Αυτή η μεθοδολογία χρησιμεύει ως βάση για την προσέγγιση της εφαρμογής HCvision για τη σύσταση μεθόδων σύνδεσης και τον προσδιορισμό του βέλτιστου αριθμού συστάδων στην

ιεραρχική συσταδοποίηση. Η HCvision χρησιμοποιεί μια στρατηγική βασισμένη στα δεδομένα, ενσωματώνοντας απρόσκοπτα θεωρητικές αρχές και εμπειρικές αναλύσεις. Λαμβάνοντας υπόψη τους συντελεστές ασυνέπειας και αξιολογώντας διάφορες μεθόδους σύνδεσης, η εφαρμογή καθοδηγεί τους χρήστες προς ουσιαστικά και ερμηνεύσιμα αποτελέσματα. Αυτή η στιβαρή μεθοδολογία εξασφαλίζει μια φιλική προς τον χρήστη εμπειρία, προωθώντας την τεκμηριωμένη λήψη αποφάσεων κατά την εξερεύνηση και ανάλυση πολύπλοκων δομών δεδομένων.

2.7 Πλεονεκτήματα

Ανακάλυψη ιεραρχικής δομής
Ένα σημαντικό πλεονέκτημα της ιεραρχικής συσταδοποίησης είναι η ικανότητά της να αποκαλύπτει την εγγενή ιεραρχική δομή εντός ενός συνόλου δεδομένων. Αυτή η ιεραρχική οργάνωση παρέχει πληροφορίες σχετικά με τις σχέσεις μεταξύ των σημείων δεδομένων σε διάφορα επίπεδα λεπτομέρειας, διευκολύνοντας τη διαφοροποιημένη κατανόηση της υποκείμενης δομής[18].

Οπτικοποίηση μέσω δενδρογραμμάτων
Η αναπαράσταση των αποτελεσμάτων της συσταδοποίησης μέσω δενδρογραμμάτων επιτρέπει τη διαισθητική οπτική ερμηνεία. Τα δενδρογράμματα παρέχουν σαφή απεικόνιση των ιεραρχικών σχέσεων μεταξύ των σημείων δεδομένων, βοηθώντας στον εντοπισμό συνεκτικών συστάδων και στη συνολική δομή του συνόλου δεδομένων[20].

Ευελιξία στην εξαγωγή συστάδων
Η ιεραρχική συσταδοποίηση παρέχει ευελιξία στην εξαγωγή συστάδων σε διαφορετικά επίπεδα λεπτομέρειας, κόβοντας το δενδρόγραμμα στα επιθυμητά ύψη. Αυτή η προσαρμοστικότητα επιτρέπει στους ερευνητές να διερευνήσουν τη δομή του συνόλου δεδομένων σε πολλαπλές αναλύσεις, ικανοποιώντας ποικίλους αναλυτικούς στόχους[28].

2.8 Μειονεκτήματα

Ευαισθησία στον θόρυβο και στις ακραίες τιμές
Η ιεραρχική συσταδοποίηση μπορεί να είναι ευαίσθητη στο θόρυβο και τις ακραίες τιμές, ιδίως όταν χρησιμοποιούνται μέθοδοι σύνδεσης όπως η απλή σύνδεση. Οι ακραίες τιμές ή τα θορυβώδη σημεία δεδομένων μπορεί να επηρεάσουν δυσανάλογα το σχηματισμό συστάδων, οδηγώντας σε μη βέλτιστα αποτελέσματα[29].

Υπολογιστική πολυπλοκότητα
Η υπολογιστική πολυπλοκότητα των αλγορίθμων ιεραρχικής συσταδοποίησης, ιδίως για μεγάλα σύνολα δεδομένων, μπορεί να αποτελέσει μειονέκτημα. Οι απαιτήσεις σε χρόνο και μνήμη μπορεί να αυξηθούν σημαντικά, επηρεάζοντας τη σκοπιμότητα εφαρμογής της ιεραρχικής συσταδοποίησης σε μαζικά σύνολα δεδομένων[30].

Έλλειψη επεκτασιμότητας
Καθώς το μέγεθος του συνόλου δεδομένων αυξάνεται, η ιεραρχική συσταδοποίηση μπορεί να καταστεί μη πρακτική λόγω της εγγενούς τετραγωνικής χρονικής πολυπλοκότητας. Αυτή η έλλειψη επεκτασιμότητας περιορίζει την εφαρμογή της σε σύνολα δεδομένων διαχειρίσιμου

μεγέθους[29].

2.9 Κλιμακωσιμότητα και Υπολογιστική Πολυπλοκότητα

Η συσσωρευτική ιεραρχική συσταδοποίηση περιλαμβάνει επαναληπτική συγχώνευση συστάδων, καθιστώντας την υπολογιστικά εντατική. Η χρονική πολυπλοκότητα της συσσωρευτικής συσταδοποίησης είναι συνήθως O του N στον κύβο, όπου N είναι ο αριθμός των σημείων δεδομένων[18]. Αυτή η τετραγωνική χρονική πολυπλοκότητα μπορεί να αποτελέσει τροχοπέδη για μεγάλα σύνολα δεδομένων.

Η κλιμακωσιμότητα είναι ένα κρίσιμο ζήτημα στην ιεραρχική συσταδοποίηση, ιδίως όταν πρόκειται για μεγάλα σύνολα δεδομένων. Η προσπάθεια ενίσχυσης της περιλαμβάνει στρατηγικές προσεγγίσεις για τον μετριασμό της υπολογιστικής πολυπλοκότητας με μεγάλα σύνολα δεδομένων. Οι τεχνικές υποδειγματοληψίας, όπου χρησιμοποιείται ένα αντιπροσωπευτικό υποσύνολο, μπορούν να μετριάσουν τον υπολογιστικό φόρτο. Η παραλληλοποίηση εκμεταλλεύεται αρχιτεκτονικές κατανεμημένων υπολογιστών για ταχύτερους υπολογισμούς αποστάσεων ανά ζεύγη. Οι μέθοδοι προσέγγισης θυσιάζουν την ακρίβεια για την ταχύτητα, βοηθώντας σε ταχύτερους χρόνους εκτέλεσης[18]. Αυτές οι στρατηγικές προσφέρουν συλλογικά λύσεις για τον χειρισμό των προκλήσεων που θέτει η επεκτασιμότητα στην ιεραρχική συσταδοποίηση.

2.10 Πραγματικές Εφαρμογές

Η ιεραρχική συσταδοποίηση, λόγω της προσαρμοστικότητας της και της ικανότητάς της να αποκαλύπτει περίπλοκες δομές σε σύνολα δεδομένων, βρίσκει εφαρμογές σε διάφορους τομείς, αναδεικνύοντας την ευελιξία της στην αντιμετώπιση σύνθετων αναλυτικών προκλήσεων.

Ανάλυση γονιδιωματικών δεδομένων στη βιολογία

Στον τομέα της βιολογίας, η ιεραρχική συσταδοποίηση έχει διαδραματίσει καθοριστικό ρόλο στην ανάλυση γονιδιωματικών δεδομένων. Έχει χρησιμοποιηθεί εκτενώς για την συσταδοποίηση γονιδίων ή δειγμάτων με βάση τα προφίλ έκφρασής τους, επιτρέποντας στους ερευνητές να εντοπίζουν μοτίβα και σχέσεις μέσα σε σύνολα γονιδιωματικών δεδομένων μεγάλης κλίμακας. Με την συσταδοποίηση γονιδίων ή δειγμάτων με παρόμοια πρότυπα έκφρασης, η ιεραρχική συσταδοποίηση βοηθά στην αποκάλυψη πιθανών λειτουργικών σχέσεων και στην κατανόηση των υποκείμενων γενετικών μηχανισμών[31].

Τμηματοποίηση πελατών στο μάρκετινγκ

Στον τομέα του μάρκετινγκ και της ανάλυσης πελατών, η ιεραρχική συσταδοποίηση εφαρμόζεται για την τμηματοποίηση πελατών με βάση ομοιότητες στην αγοραστική τους συμπεριφορά, τις προτιμήσεις ή τα δημογραφικά χαρακτηριστικά τους. Με την συσταδοποίηση πελατών με συγκρίσιμα χαρακτηριστικά, οι επιχειρήσεις μπορούν να προσαρμόζουν αποτελεσματικότερα τις στρατηγικές μάρκετινγκ, ενισχύοντας τη δέσμευση και την ικανοποίηση των πελατών. Η ιεραρχική δομή επιτρέπει τη διαφοροποιημένη τμηματοποίηση, επιτρέποντας στις επιχειρήσεις να στοχεύουν συγκεκριμένες ομάδες πελατών με εξατομικευμένες εκστρατείες μάρκετινγκ[18].

Διαφοροποίηση χαρτοφυλακίου στα χρηματοοικονομικά

Η ιεραρχική συσταδοποίηση χρησιμοποιείται στα χρηματοοικονομικά για τη διαφοροποίηση

χαρτοφυλακίου και τη διαχείριση κινδύνου. Με την συσταδοποίηση των χρηματοοικονομικών περιουσιακών στοιχείων με βάση την ιστορική τους απόδοση και το προφίλ κινδύνου, οι επενδυτές μπορούν να κατασκευάσουν διαφοροποιημένα χαρτοφυλάκια που εξισορροπούν τον κίνδυνο και την απόδοση. Η ιεραρχική συσταδοποίηση βοηθά στον εντοπισμό ομάδων περιουσιακών στοιχείων με παρόμοια συμπεριφορά στην αγορά, επιτρέποντας στους επενδυτές να λαμβάνουν τεκμηριωμένες αποφάσεις σχετικά με τη σύνθεση του χαρτοφυλακίου και τον μετριασμό του κινδύνου.

Τμηματοποίηση και Ανάλυση Εικόνας

Στον τομέα της ανάλυσης εικόνων, η ιεραρχική συσταδοποίηση χρησιμοποιείται για την τμηματοποίηση εικόνων, όπου παρόμοιες περιοχές σε μια εικόνα ομαδοποιούνται μαζί. Αυτό διευκολύνει την αναγνώριση αντικειμένων, την κατανόηση σκηνών και τις εφαρμογές υπολογιστικής όρασης. Με την ιεραρχική οργάνωση των συστατικών της εικόνας, οι ερευνητές μπορούν να εξάγουν δομές και σχέσεις με νόημα, συμβάλλοντας στην πρόοδο της ιατρικής απεικόνισης, της ανάλυσης δορυφορικών εικόνων και της διάγνωσης με τη βοήθεια υπολογιστή[32].

Ανάλυση κοινωνικών δικτύων

Στο πεδίο των κοινωνικών επιστημών, η ιεραρχική συσταδοποίηση εφαρμόζεται για την ανάλυση κοινωνικών δικτύων και τον εντοπισμό κοινοτικών δομών. Με την συσταδοποίηση των ατόμων με βάση τις κοινωνικές αλληλεπιδράσεις τους, οι ερευνητές αποκτούν γνώσεις σχετικά με τη δυναμική της κοινότητας, τους κόμβους με επιρροή και τη ροή πληροφοριών εντός των δικτύων. Η προσέγγιση αυτή έχει επιπτώσεις στην κοινωνιολογία, την ανθρωπολογία και την επιδημιολογία, όπου η κατανόηση των κοινωνικών δομών είναι ζωτικής σημασίας για τη μελέτη της ανθρώπινης συμπεριφοράς και της εξάπλωσης των μολυσματικών ασθενειών[33].

Κεφάλαιο 3

Γλώσσες και Τεχνολογίες

Το παρόν κεφάλαιο επικεντρώνεται σε μια σφαιρική ανάλυση των τεχνολογιών που χρησιμοποιήθηκαν για την ανάπτυξη της εφαρμογής HCvision. Η πολύπλοκη αρχιτεκτονική της εφαρμογής περιλαμβάνει ένα φάσμα τεχνολογιών, καθορίζοντας διακριτικά τις τεχνολογίες της πλευράς του διακομιστή (server) και της πλευράς του πελάτη (client).

Η ανάλυσή μας θα επικεντρωθεί στον απολογισμό των πιο κρίσιμων τεχνολογιών που διαμορφώνουν την ανθεκτικότητα και τη λειτουργικότητα τόσο του διακομιστή όσο και του πελάτη. Με τον λεπτομερή αυτόν εξερευνητικό χαρακτήρα, σκοπεύουμε να προσφέρουμε μια διεισδυτική κατανόηση των ρόλων, της επίδρασης και των συνεργιών αυτών των τεχνολογιών εντός του HCvision.

3.1 Τεχνολογίες Server side

Η επιλογή των κατάλληλων τεχνολογιών στην πλευρά του διακομιστή είναι κρίσιμη για την αποτελεσματικότητα και τη σταθερότητα μιας εφαρμογής AutoML. Οι επιλεγμένες τεχνολογίες διαδραματίζουν κεντρικό ρόλο στον προσδιορισμό του API της εφαρμογής, επηρεάζοντας την απόκριση του διακομιστή, την ταχύτητα, τη δυνατότητα ανταπόκρισης σε φορτίο και ταυτόχρονων αιτημάτων, καθώς και τη συνολική διαδικασία ανάπτυξης. Ένας σωστά επιλεγμένος συνδυασμός τεχνολογιών αποτελεί την βάση ενός αξιόπιστου και υψηλής απόδοσης διακομιστή, απαραίτητο για τη διαχείριση των ταυτόχρονων αιτημάτων και την παροχή μιας ομαλής εμπειρίας χρήστη. Στις επόμενες ενότητες, εξετάζουμε λεπτομερώς κάθε τεχνολογία που χρησιμοποιήθηκε για τη δημιουργία ενός ανθεκτικού και αποδοτικού διακομιστή για την εφαρμογή μας.

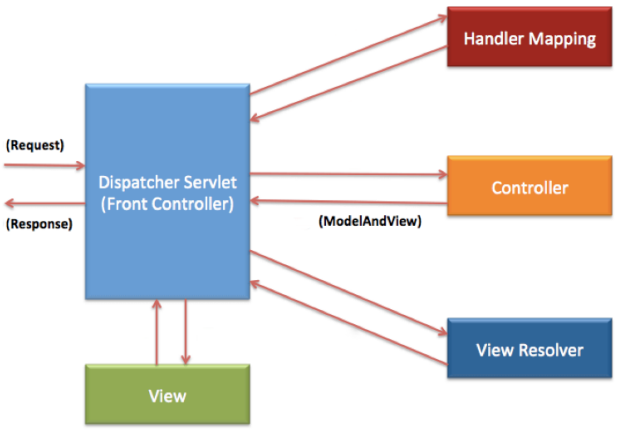
Spring Boot
Το Spring Boot, ως επέκταση του ευρέως χρησιμοποιούμενου Spring Framework, εισάγει μια επαναστατική προσέγγιση στην ανάπτυξη εφαρμογών βασισμένων σε Java. Αντίθετα με τις παραδοσιακές εφαρμογές Spring που απαιτούν σαφείς ρυθμίσεις για διάφορα στοιχεία, το Spring Boot απλοποιεί τη διαδικασία ανάπτυξης μέσω αυτόματης ρύθμισης. Εξαιρετικό στο να ανιχνεύει ευφυώς τον χαρακτήρα της εφαρμογής και να ρυθμίζει αυτόματα τα απαραίτητα στοιχεία, εξαλείφει την ανάγκη για χειροκίνητες ρυθμίσεις σε συνηθισμένα σενάρια. Οι εκκινητές εξαρτήσεων του Spring Boot απλοποιούν την επιλογή βιβλιοθηκών στην ώρα της χρήσης και της εκτέλεσης της εφαρμογής, μειώνοντας το μήκος των προδιαγραφών δημιουργίας[34].

Κάτω από το καπό, το Spring Boot εκμεταλλεύεται τις ισχυρές έννοιες των beans και του dependency injection. Επιτρέπει στους προγραμματιστές να ορίζουν και να ρυθμίζουν στοιχεία της εφαρμογής ως beans, χρησιμοποιώντας την αντιστροφή του ελέγχου για τη διαχείριση του κύκλου ζωής και των εξαρτήσεών τους. Με αναφορές όπως @Component, @Service, και @Repository, οι προγραμματιστές μπορούν να σημειώνουν τις κλάσεις για αυτόματη ανακάλυψη και εγγραφή ως Spring beans[34].

Η ρύθμιση στο Spring Boot μπορεί να επιτευχθεί μέσω αναφορών, XML ή αρχείων YAML. Αναφορές όπως @Configuration επιτρέπουν τη δημιουργία κλάσεων ρύθμισης βασισμένων σε Java, ενώ το @Value μπορεί να ενσωματώσει τιμές ιδιοκτησίας απευθείας στα beans. Αυτή η ευελιξία προσαρμόζεται σε διάφορες προτιμήσεις των προγραμματιστών και απαιτήσεις του έργου[34].

Το Spring Boot υιοθετεί μια αρχιτεκτονική ελεγκτών-υπηρεσιών-πόρων, όπου οι ελεγκτές διαχειρίζονται τις εισερχόμενες αιτήσεις, οι υπηρεσίες ενθυλακώνουν τη λογική της επιχείρησης, και οι πόροι διαχειρίζονται εξωτερικές αλληλεπιδράσεις ή λειτουργούν ως αποθηκευτήρια. Η χρήση αναφορών όπως @Controller και @RestController απλοποιεί τη δημιουργία API, επιτρέποντας στους προγραμματιστές να δημιουργήσουν αξιόπιστα και κλιμάκωσιμα συστήματα πίσω από τις σκηνές[34].

Σε βαθύτερο επίπεδο η διαδικασία λειτουργίας του Spring MVC ακολουθεί ένα συγκεκριμένο μοντέλο ροής. Όταν λαμβάνεται μια αίτηση, αυτή πρώτα φτάνει στον Front Controller, γνωστό και ως DispatcherServlet. Ο DispatcherServlet περνάει την αίτηση στον HandlerMapping, ο οποίος εντοπίζει τον κατάλληλο ελεγκτή (Controller) για την αίτηση. Έπειτα, ο HandlerMapping στέλνει τα στοιχεία του ελεγκτή στον DispatcherServlet. Ο DispatcherServlet καλεί τον ελεγκτή εντοπισμένο από τον HandlerMapping, ο οποίος επεξεργάζεται την αίτηση καλώντας την κατάλληλη μέθοδο και προετοιμάζοντας τα δεδομένα. Ο ελεγκτής στέλνει το ModelAndView (δεδομένα μοντέλου και όνομα προβολής) στον DispatcherServlet. Μόλις ο DispatcherServlet λάβει το αντικείμενο ModelAndView, το περνάει στον ViewResolver για να βρει την κατάλληλη προβολή. Ο ViewResolver αναγνωρίζει την προβολή και την στέλνει πίσω στον DispatcherServlet. Ο DispatcherServlet καλεί την κατάλληλη προβολή που εντοπίστηκε από τον ViewResolver. Η προβολή δημιουργεί την απόκριση σε μορφή HTML και την στέλνει στον DispatcherServlet. Τέλος, ο DispatcherServlet στέλνει την απόκριση στον περιηγητή. Ο περιηγητής απεικονίζει τον κώδικα HTML και τον εμφανίζει στον τελικό χρήστη[35]. Στο σχήμα 3.1 παρουσιάζεται η ροή που περιγράφηκε[35].



Σχήμα 3.1: Διάγραμμα ροής του Spring Boot

Η γραμμή εντολών του Spring Boot ενισχύει ακόμη περισσότερο την ανάπτυξη, επιτρέποντας τη δημιουργία εφαρμογών Spring σε Groovy με ελάχιστο θόρυβο και τυπικότητα συνηθισμένη στις εφαρμογές Java. Με το Actuator του Spring Boot, παρέχονται ολοκληρωμένες εισαγωγές στη λειτουργία μιας εκτελούμενης εφαρμογής, περιγράφοντας ακριβώς τα beans που βρίσκονται στο κέντρο της εφαρμογής Spring, τις αντιστοιχίσεις των ελεγκτών Spring MVC, τις διαθέσιμες ιδιότητες ρύθμισης κ.ά. Οι επιλογές αυτόματης ρύθμισης και διαχείρισης εξαρτήσεων που προσφέρει το Spring Boot το καθιστούν ιδανική επιλογή για προγραμματιστές που αναζητούν ένα πολυδιάστατο και ευέλικτο πλαίσιο[36].

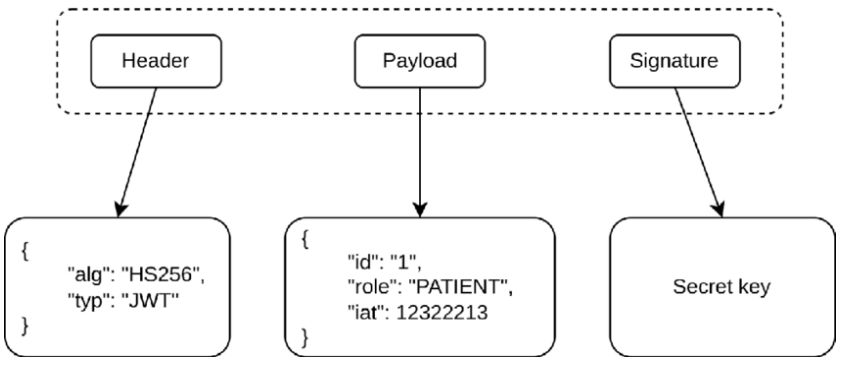
Η αποτελεσματικότητα του Spring Boot επεκτείνεται πέρα από την απλοποίηση της ανάπτυξης. Παρέχει γρήγορη εκκίνηση, γρήγορο κλείσιμο και βελτιστοποιημένη εκτέλεση από προεπιλογή. Με την υποστήριξη για αντιδραστικά (μη-μπλοκαριστά) μοντέλα προγραμματισμού, το framework ενισχύει ακόμη περισσότερο την αποτελεσματικότητα. Η ολοκληρωμένη συνεργασία του με το Spring Cloud διευκολύνει την ανάπτυξη αρχιτεκτονικών που βασίζονται σε υπηρεσίες, επιτρέποντας την άνετη αναπτυξιακή διαδικασία στον χώρο του cloud[37].

Το Spring, το υποκείμενο πλαίσιο, έχει μια πλούσια ιστορία και εμπιστεύεται ευρέως από προγραμματιστές σε όλο τον κόσμο, με συνεισφορές από μεγάλα ονόματα της τεχνολογίας όπως Alibaba, Amazon, Google και Microsoft[38]. Η έναρξη του Spring Boot καλλιέργησε ακόμη περισσότερο την εμπειρία του Spring, καθιστώντας την ακόμη πιο προσβάσιμη, γρήγορη και ασφαλή.

Συνολικά, το Spring Boot αναδεικνύεται ως ακρογωνιαίος λίθος στην ανάπτυξη εφαρμογών βασισμένων σε Java, προσφέροντας ανεπανάληπτα οφέλη σε θέματα απλότητας, ταχύτητας και ευελιξίας. Η αυτόματη ρύθμιση, οι εκκινητές εξαρτήσεων και τα εκτεταμένα χαρακτηριστικά το καθιστούν την ιδανική επιλογή για εφαρμογές διακομιστή, συμπεριλαμβανομένης της ανάπτυξης ανθεκτικών API.

JSON Web Tokens

Ένα JSON Web Token (JWT) είναι ένα JSON αντικείμενο που καθορίζεται στο RFC 7519, λειτουργώντας ως ασφαλής μέθοδος ανταλλαγής πληροφοριών μεταξύ δύο πλευρών. Αποτελείται από τρία στοιχεία: ένα κεφαλίδα, ένα φορτίο και μια υπογραφή(βλ. Σχήμα 3.2[39]). Η κεφαλίδα καθορίζει τον αλγόριθμο κατακερματισμού που θα χρησιμοποιηθεί για να υπολογιστεί η υπογραφή του JWT. Το φορτίο περιέχει δεδομένα, όπως αναγνωριστικά χρηστών ή email, ενώ η υπογραφή εξασφαλίζει την ακεραιότητα του περιεχομένου μέσω υπολογισμένου αλγορίθμου.



Σχήμα 3.2: Δομή JWT

Τα JWT βρίσκουν σημαντική χρησιμότητα σε διάφορα σενάρια, με την εξουσιοδότηση να είναι ένα κύριο παράδειγμα χρήσης. Αφού ένας χρήστης συνδεθεί, τα συνεχόμενα αιτήματα περιλαμβάνουν το JWT, παρέχοντας πρόσβαση σε διαδρομές, υπηρεσίες και πόρους με βάση τα δικαιώματα του τόκεν. Ευρέως χρησιμοποιούμενο στην Ενιαία Σύνδεση (Single Sign-On) λόγω της αποτελεσματικότητάς του σε διάφορους τομείς, τα JWT προσφέρουν πλεονέκτημα χάρη στη συμπαγή φύση τους, ευκολότερα μεταφερόμενα σε περιβάλλοντα HTML και HTTP σε σύγκριση με πρότυπα βασισμένα σε XML όπως το SAML.

Στην ταυτοποίηση, ένας χρήστης λαμβάνει ένα JWT κατά την επιτυχημένη σύνδεση, υπογραμμίζοντας την ανάγκη προσεκτικής χειριστικής των τόκεν, καθώς αποτελούν διαπιστευτήρια. Είναι ουσιώδες να μην διατηρούνται τα τόκεν περισσότερο από όσο είναι απαραίτητο και να μην αποθηκεύονται ευαίσθητα δεδομένα συνεδρίας στην αποθήκη του προγράμματος περιήγησης λόγω ανεπάρκειας ασφαλείας.

Όταν ένας χρήστης επιθυμεί να αποκτήσει πρόσβαση σε προστατευμένη διαδρομή ή πόρο, το πρόγραμμα χρήστη αποστέλλει το JWT, συνήθως στην κεφαλίδα Εξουσιοδότησης χρησιμοποιώντας το σχήμα Bearer. Αυτό εξασφαλίζει ασφαλή και εξουσιοδοτημένη επικοινωνία, αποδεικνύοντας την ευελιξία και την αξιοπιστία των JSON Web Tokens στη βελτίωση των διαδικασιών ανταλλαγής πληροφοριών και εξουσιοδότησης[40].

MySQL

Η MySQL, μια βάση δεδομένων ανοικτού κώδικα, ξεχωρίζει ως η δεύτερη πιο δημοφιλής βάση δεδομένων στον κόσμο, διαδεχόμενη μόνο τη βάση δεδομένων της Oracle σύμφωνα με το DB-Engines. Αναπτυγμένο για περισσότερο από 25 χρόνια σε συνεργασία με τους χρήστες, αποτελεί τη βάση πολλών σημαντικών εφαρμογών όπως το Facebook, το Twitter, το Netflix, το Uber κ.ά. Με φήμη για την ταχύτητα, την αξιοπιστία, την επεκτασιμότητα και την ευκολία χρήσης, το MySQL ανταποκρίνεται αποτελεσματικά σε μεγάλες βάσεις δεδομένων και ξεχωρίζει σε απαιτητικά περιβάλλοντα παραγωγής[41].

Ως ανοικτού κώδικα βάση δεδομένων, το MySQL εξασφαλίζει συνεχή ανάπτυξη με πλούσια σύνολα λειτουργιών, γεγονός που το καθιστά συμβατό με διάφορες εφαρμογές και γλώσσες προγραμματισμού. Η συνδεσιμότητά του, η ταχύτητά του και η ασφάλειά του το καθιστούν κατάλληλο για την πρόσβαση σε βάσεις δεδομένων στο Διαδίκτυο. Κύρια πλεονεκτήματα περιλαμβάνουν την ευκολία χρήσης, την αξιοπιστία, την επεκτασιμότητα, την απόδοση, την υψηλή διαθεσιμότητα και την ευελιξία, με το MySQL Document Store να υποστηρίζει τόσο παραδοσιακές SQL όσο και NoSQL εφαρμογές χωρίς σχήμα.

Η ταχύτητα του MySQL συνεχίζει να εξελίσσεται με βελτιώσεις στην InnoDB και τον βελτιστοποιητή ερωτημάτων. Προσφέρει απλότητα στην εγκατάσταση και διαχείριση, υποστηρίζοντας SQL και ενσωματώνοντας άψογα σε σύγχρονα συστήματα βάσεων δεδομένων. Η δικτυωμένη φύση του επιτρέπει την πρόσβαση σε βάσεις δεδομένων μέσω Διαδικτύου με αυστηρό έλεγχο πρόσβασης για την ασφάλεια δεδομένων. Η φορητότητα του MySQL επεκτείνει τη συμβατότητά του σε διάφορα λειτουργικά συστήματα. Η διαθεσιμότητα του υπό την άδεια GNU General Public License (GPL) επιτρέπει τη χρήση χωρίς κόστος για τις περισσότερες χρήσεις ενώ παρέχονται εμπορικές άδειες για οργανισμούς με συγκεκριμένες προτιμήσεις. Η ανοικτή διανομή και η διαθεσιμότητα του πηγαίου κώδικα καθιστούν το MySQL εύκολο στην απόκτηση, παρέχοντας διαφάνεια και ευκολία στους ελέγχους ασφαλείας. Το MySQL, που αρχικά αναπτύχθηκε ανεξάρτητα και αργότερα εξαγοράστηκε από την Oracle, παραμένει θεμέλιο στη διαχείριση βάσεων δεδομένων, υποστηρίζοντας εφαρμογές σε διάφορες βιομηχανίες[42].

Python
Το Python, δημιουργία του Guido van Rossum και πρώτη δημοσίευση του το 1991, αναδεικνύεται ως μια διερμηνευμένη, αντικειμενοστραφής, υψηλού επιπέδου γλώσσα προγραμματισμού με δυναμική σημασιολογία. Εξαιρετική στην Ανάπτυξη Εφαρμογών με Γρήγορο Ρυθμό (RAD), λειτουργεί επίσης ως ευέλικτη γλώσσα σεναρίων, συνδέοντας απροβλημάτιστα υπάρχουσες συνιστώσες. Η σύνταξή της, προσανατολισμένη στην αναγνωσιμότητα, μειώνει το κόστος συντήρησης των προγραμμάτων[43].

Το Python υποστηρίζει ενότητες και πακέτα, προωθώντας την τμηματοποίηση των προγραμμάτων και την επαναχρησιμοποίηση του κώδικα. Η αποτελεσματικότητά του στη διαχείριση υψηλού επιπέδου δομών δεδομένων, σε συνδυασμό με την απλή και αποτελεσματική προσέγγιση της αντικειμενοστραφούς προγραμματισμού, το θέτουν ως ιδανική επιλογή για σενάρια και γρήγορη ανάπτυξη εφαρμογών σε διάφορες πλατφόρμες. Η επεκτασιμότητά του επιτρέπει την προσθήκη νέων ενσωματωμένων συναρτήσεων ή ενοτήτων, καθιστώντας το προσαρμόσιμο για κρίσιμες λειτουργίες ή σύνδεση με βιβλιοθήκες δυαδικού κώδικα[44].

Το Python αποτελεί μία από τις πιο δημοφιλείς γλώσσες προγραμματισμού για την επιστήμη των δεδομένων, διαθέτοντας μια πληθώρα χρήσιμων βιβλιοθηκών που έχουν αναπτυχθεί από την ευρύτατη κοινότητά της. Παρά τη μικρότερη απόδοση σε υπολογιστικές εργασίες, η ύπαρξη βιβλιοθηκών επέκτασης, όπως η NumPy και η SciPy, εξασφαλίζει γρήγορες και διανυσματικές λειτουργίες. Για εργασίες προγραμματισμού μηχανικής μάθησης, αναφερόμαστε συχνά στη βιβλιοθήκη scikit-learn, μία από τις πιο δημοφιλείς βιβλιοθήκες ανοικτού κώδικα σήμερα[45].

3.2 Τεχνολογίες Client side

Στον κόσμο της ανάπτυξης λογισμικού, η επιλογή των τεχνολογιών στην πλευρά του πελάτη (client-side) κατέχει κρίσιμη σημασία, επηρεάζοντας όχι μόνο την εμπειρία του χρήστη, αλλά και τη συνολική αποτελεσματικότητα της διαδικασίας ανάπτυξης. Η επιλογή βέλτιστων τεχνολογιών εξασφαλίζει ένα ομαλό και αποκρίνεται στις ανάγκες του εφαρμογής, συμβάλλοντας σημαντικά στην ικανοποίηση του χρήστη. Αυτό το κεφάλαιο επικεντρώνεται σε μια λεπτομερή ανάλυση των τεχνολογιών που χρησιμοποιήθηκαν στην πλευρά του πελάτη για την εφαρμογή AutoML μας, φωτίζοντας τις ατομικές τους συνεισφορές και συνέργειες.

Angular
Το Angular, ένα καινοτόμο πλαίσιο σχεδιασμού εφαρμογών και πλατφόρμα ανάπτυξης, βρίσκεται στο επίκεντρο της δημιουργίας αποδοτικών και προηγμένων εφαρμογών μονής σελίδας. Προωθούμενο από τη Google, το Angular αποτελεί ένα πλαίσιο εφαρμογής ιστού με ανοικτό κώδικα που βασίζεται στην TypeScript, εξασφαλίζοντας δυναμικές, αποδοτικές και εξελιγμένες ιστοσελίδες. Αρχικά γνωστό ως AngularJS, το πλαίσιο υπέστη σημαντικές αλλαγές, οδηγώντας στο Angular 2 και σε μετέπειτα εκδόσεις. Το Angular δομείται γύρω από αλληλεξαρτώμενα συστατικά, καθένα με το δικό του HTML, CSS και ελεγκτή TypeScript, προωθώντας την modular και συντηρησιμότητα[46].

Αυτό το πλαίσιο εκμεταλλεύεται την TypeScript για τη δυνατότητά της για έλεγχο τύπων και βελτιωμένη δομή κώδικα, καθιστώντας το ισχυρό εργαλείο για τη δημιουργία πελατειακών

εφαρμογών. Το Angular δυναμώνει τη δημιουργία υπηρεσιών, οδηγιών και διάφορων άλλων στοιχείων, συμβάλλοντας στην ευελιξία και την επεκτασιμότητα του. Είναι μια δημοφιλής επιλογή μεταξύ των προγραμματιστών, σηματοδοτώντας τη δημοτικότητά του στη σύγχρονη προγραμματισμό εφαρμογών ιστού[47].

Τα έργα του Angular οργανώνονται γύρω από συστατικά, υπηρεσίες και οδηγίες, διευκολύνοντας έναν καθαρό και modulare κώδικα. Ως πακέτο Node, το Angular συνεργάζεται αρμονικά με το Node.js, εκμεταλλευόμενο τη δυνατότητά του για τη διαχείριση πακέτων. Το Node.js, ένα περιβάλλον εκτέλεσης JavaScript ανοικτού κώδικα, διαδραματίζει κρίσιμο ρόλο στην ανάπτυξη του Angular, εξασφαλίζοντας αποτελεσματική διαχείριση έργων και χειρισμό εξαρτήσεων.

Στο πλαίσιο της πελατειακής εφαρμογής της HCvision, το Angular υπηρέτησε ως κεντρική τεχνολογία, ενδυναμώνοντας τη δημιουργία μιας στιβαρής και αποκρίνεται διεπαφής χρήστη. Ο ρόλος του ήταν καθοριστικός για την παροχή μιας ομαλής εμπειρίας χρήστη, υπογραμμίζοντας τη σημασία του Angular στη σύγχρονη ανάπτυξη εφαρμογών ιστού[48].

Angular Material

Για την υλοποίηση του χρήστικού περιβάλλοντος (UI), επιλέχθηκε ο σχεδιασμός υλικού (Material Design). Το Material Design αποτελεί τη γλώσσα σχεδίασης της Google, παρέχοντας ένα πλούσιο σύνολο οδηγιών. Αυτή η γλώσσα σχεδίασης μιμείται αντικείμενα από τον πραγματικό κόσμο, επιτυγχάνοντας έτσι τη μεταφορική έννοια του ”υλικού”. Τα στοιχεία και τα χειριστήρια στο περιβάλλον χρήστη απεικονίζουν βάθος και υφή, προσθέτουν σκιές και χρησιμοποιούν την κίνηση με νόημα. Η Angular προσφέρει μια πολύ ισχυρή βιβλιοθήκη που ακολουθεί το υλικό σχεδιασμό.

Η βιβλιοθήκη Angular Material λειτουργεί ως ισχυρό πλαίσιο, προσφέροντας εκτεταμένη ευελιξία για τη δημιουργία μιας ξεχωριστής και αναγνωρίσιμης ταυτότητας για εφαρμογές διαδικτυακού χρήστη. Ένας σημαντικός συνεισφέρων στην επίτευξη αυτής της μοναδικότητας είναι η ενσωμάτωση θεμάτων, τα οποία διαδραματίζουν κεντρικό ρόλο. Οι χρήστες έχουν τη δυνατότητα να επιλέξουν από έναν προκαθορισμένο κατάλογο θεμάτων ή, εναλλακτικά, να δημιουργήσουν ένα προσαρμοσμένο θέμα που προσαρμόζεται στις συγκεκριμένες τους προτιμήσεις.

Επιπλέον, το Angular Material παρέχει μια σειρά προηγμένων συστατικών που συμμορφώνονται με τις αρχές του Material Design, εξυπηρετώντας διάφορες λειτουργίες σε τομείς όπως οι φόρμες δεδομένων, η πλοήγηση, οι διατυπώσεις και πολλοί άλλοι τομείς[49].

Type Script

Το TypeScript, δημιουργημένο από τη Microsoft και κυκλοφορημένο για πρώτη φορά το 2012, έχει εξελιχθεί σε ένα ισχυρό superset της JavaScript, προσφέροντας βελτιωμένη λειτουργικότητα με την εισαγωγή στατικού τύπου. Το Angular, το δημοφιλές πλαίσιο εφαρμογής για τον ιστό που αναπτύχθηκε από τη Google, γράφτηκε καθαυτό σε TypeScript. Ένα από τα κύρια πλεονεκτήματα του TypeScript είναι η δυνατότητά του να επιβάλλει πληροφορίες τύπου, επιτρέποντας στους προγραμματιστές να καθορίζουν τους τύπους τιμών που μπορούν να κρατούν οι μεταβλητές.

Το βασικό πλεονέκτημα του TypeScript είναι η ικανότητά του να επιβάλλει περιορισμούς στους τύπους μεταβλητών, εμποδίζοντας συνήθεις παγίδες στη JavaScript. Για παράδειγμα, μπορεί να ανιχνεύσει σφάλματα σε περιπτώσεις όπου μια μεταβλητή που προορίζεται για αριθμητι-

κές τιμές καταλήγει κατά λάθος να κρατάει έναν χαρακτήρα. Αυτή η ικανότητα γίνεται καίρια για τη διατήρηση της ακεραιότητας του κώδικα, ιδίως σε συνεργατικά ή εξελισσόμενα έργα.

Οι προγραμματιστές βρίσκουν αρκετούς πειστικούς λόγους για να υιοθετήσουν το TypeScript. Καταρχάς, ενισχύει την κατανοησιμότητα του κώδικα με τον καθορισμό τύπων μεταβλητών, καθιστώντας τον κώδικα πιο ευανάγνωστο για άλλους ή ακόμη και για τον ίδιο τον προγραμματιστή μετά από κάποιο χρονικό διάστημα. Επιπλέον, όταν χρησιμοποιείται με υποστηριζόμενα περιβάλλοντα επεξεργασίας, το TypeScript παρέχει έξυπνη υποστήριξη IntelliSense για τον κώδικα, προτείνοντας γνωστές μεταβλητές ή συναρτήσεις και πληροφορίες για τον τύπο τιμής που αναμένεται.

Επιπλέον, το TypeScript ανιχνεύει σφάλματα πριν την εκτέλεση του κώδικα, ελαχιστοποιώντας τον χρόνο ανατροφοδότησης κατά τη συγγραφή μη έγκυρου κώδικα.

Η χρήση του TypeScript είναι εντελώς προαιρετική, επιτρέποντας στους προγραμματιστές να επιλέγουν πότε να χρησιμοποιούν τους τύπους ανάλογα με τη σημασία τους. Είναι σημαντικό να σημειωθεί πως έχει γίνει στάνταρ επιλογή σε έργα που χρησιμοποιούν το Angular. Ο διαδικασία μεταγλώττισης περιλαμβάνει τη μετατροπή του κώδικα TypeScript σε JavaScript, συνήθως χρησιμοποιώντας περιβάλλοντα[50].

Postman

Το Postman αναδεικνύεται ως μια καινοτόμος πλατφόρμα API, σχεδιασμένη για να απλοποιεί και να βελτιστοποιεί κάθε στάδιο του κύκλου ζωής του API, προωθώντας τη συνεργασία και επιταχύνοντας τη δημιουργία υψηλής ποιότητας APIs. Αυτό το ευέλικτο εργαλείο παρέχει ένα κεντρικό περιβάλλον για την αποθήκευση, τον κατάλογο και τη συνεργασία σε όλα τα στοιχεία του API. Από προδιαγραφές και τεκμηρίωση μέχρι συνταγές ροής, περιπτώσεις δοκιμών και αποτελέσματα, μετρήσεις και άλλα συστατικά που σχετίζονται με τα APIs, το Postman διευκολύνει τη σωστή οργάνωση και διαχείριση.

Η πλατφόρμα Postman περιλαμβάνει ένα εκτενές σύνολο εργαλείων που παίζουν καθοριστικό ρόλο στην επιτάχυνση του κύκλου ζωής του API, καλύπτοντας διάφορα στάδια, όπως η σχεδίαση, οι δοκιμές, η τεκμηρίωση, η προσομοίωση και ο κοινοποιητικός χαρακτήρας των APIs. Το Postman επιτρέπει στους χρήστες να αποθηκεύουν και να διαχειρίζονται αποτελεσματικά τα στοιχεία του API, διευκολύνοντας τη διαδικασία ανάπτυξης και ενισχύοντας τη συνολική παραγωγικότητα.

Κατά τη διάρκεια της ανάπτυξης του HCvision, το Postman αποδείχθηκε αναντικατάστατο. Η επίδρασή του ήταν εξαιρετικά προφανής τόσο στις δοκιμές του API όσο και στην ανάπτυξη του πελάτη. Η φιλική προς τον χρήστη διεπαφή και το πλήρες σύνολο εργαλείων της πλατφόρμας διευκόλυναν τη διαδικασία δοκιμών, εξασφαλίζοντας την αξιοπιστία και την αποτελεσματικότητα των APIs. Επιπλέον, το Postman συνέβαλε σημαντικά στην ανάπτυξη του πελάτη, παρέχοντας ένα αξιόπιστο περιβάλλον για την εξερεύνηση, συνεργασία και επικύρωση των APIs. Συνολικά, το Postman αναδείχθηκε ως ένα βασικό εργαλείο για την επιτυχή πορεία του HCvision, διευκολύνοντας μια ομαλή και αποδοτική διαδικασία ανάπτυξης του API[51].

3.3 Εργαλεία ανάπτυξης

- **IDEAS** Για την ανάπτυξη του HCvision χρησιμοποιήθηκε ένας συνδυασμός IDEs της JetBrains. Το IntelliJ IDEA χρησιμοποιήθηκε για την ανάπτυξη από την πλευρά του διακομιστή, ενώ το WebStorm έπαιξε καθοριστικό ρόλο στη διαμόρφωση του front-end. Επιπλέον, το Visual Studio Code χρησιμοποιήθηκε για τη δημιουργία Python scripts, παρέχοντας ένα ευέλικτο σύνολο εργαλείων για διάφορες πτυχές του έργου.

- **maildev** Το MailDev χρησίμευσε ως πολύτιμο εργαλείο για τη δοκιμή των παραγόμενων μηνυμάτων ηλεκτρονικού ταχυδρομείου κατά τη διάρκεια της ανάπτυξης της HCvision. Προσφέρει μια απλή διαδικτυακή διεπαφή, που εκτελείται σε Node.js, διευκολύνοντας την εύκολη δοκιμή των email. Ανοιχτός κώδικας και προσβάσιμο στη διεύθυνση: `https://github.com/maildev/maildev`.

- **http-server** Το εργαλείο http-server, ένας απλός και χωρίς ρυθμίσεις στατικός διακομιστής HTTP γραμμής εντολών, χρησιμοποιήθηκε για διάφορες πτυχές της ανάπτυξης της HCvision. Περισσότερες λεπτομέρειες μπορούν να βρεθούν στη διεύθυνση: `https://github.com/http-party/http-server`.

- **DBeaver** Το DBeaver Community, ένα εργαλείο βάσεων δεδομένων πολλαπλών πλατφορμών, χρησιμοποιήθηκε από προγραμματιστές, διαχειριστές βάσεων δεδομένων και αναλυτές καθ' όλη τη διάρκεια της ανάπτυξης του HCvision. Υποστηρίζοντας δημοφιλείς βάσεις δεδομένων SQL, όπως η MySQL, η PostgreSQL και άλλες, αποδείχθηκε ένα ευέλικτο και αποτελεσματικό εργαλείο.

Κεφάλαιο 4

Σχεδίαση και Υλοποίηση του HCvision

4.1 Λειτουργικές Απαιτήσεις

Μια λειτουργική απαίτηση αποτελεί μια δήλωση που περιγράφει τις υπηρεσίες ή τις δυνατότητες που πρέπει να παρέχει ένα σύστημα λογισμικού. Αυτή η απαίτηση περιλαμβάνει την ανάλυση και τον προσδιορισμό των διαδικασιών εισόδου στο σύστημα, τη συμπεριφορά που πρέπει να επιδεικνύει το σύστημα απέναντι σε αυτές τις εισόδους, και τα αντίστοιχα αποτελέσματα που προκύπτουν από αυτό.

Οι λειτουργικές απαιτήσεις του HCvision είναι:

- **Εγγραφή νέου χρήστη:** Ένας χρήστης θα μπορεί να κάνει εγγραφή στο σύστημα μέσω μιας φόρμας η οποία θα αποτελείται από τέσσερα πεδία: Όνομα, Επίθετο, email, Κωδικός πρόσβασης.
- **Σύνδεση χρήστη στο σύστημα:** Ο χρήστης, με βάση τα στοιχεία που έκανε εγγραφή, θα έχει τη δυνατότητα να αποθηκεύει τα δικά του σύνολα δεδομένων για μελλοντική χρήση.
- **Σελίδα Profile:** Αυτή η σελίδα θα προσφέρει τη δυνατότητα στον χρήστη να επεξεργαστεί τα στοιχεία του.
- **Επεξεργασία προσωπικών στοιχείων και κωδικού πρόσβασης:** Ο χρήστης θα μπορεί να τροποποιήσει τα δικά του προσωπικά στοιχεία, όπως το όνομα και τον κωδικό πρόσβασης.
- **Διαγραφή Λογαριασμού:** Ο χρήστης θα έχει τη δυνατότητα να διαγράψει τον λογαριασμό του.
- **Ανάκτηση κωδικού πρόσβασης:** Ο χρήστης θα μπορεί να αλλάξει τον κωδικό πρόσβασης του σε περίπτωση που τον ξεχάσει, διαδικασία που θα πραγματοποιηθεί μέσω επιβεβαίωσης μέσω email.
- **Σελίδα Datasets:** Αυτή η σελίδα προσφέρει τη δυνατότητα στο χρήστη να διαχειρίζεται τα σύνολα δεδομένων του.
- **Ανέβασμα αρχείου:** Ο χρήστης θα μπορεί να ανεβάσει ένα σύνολο δεδομένων. Αποδεκτοί τύποι είναι csv και xlsx. Το αρχείο μπορεί να είναι είτε ιδιωτικό είτε δημόσιο ανάλογα με την επιλογή του χρήστη.

- **Ανάγνωση αρχείου:** Ο χρήστης θα έχει τη δυνατότητα να βλέπει το σύνολο δεδομένων του.
- **Διαγραφή αρχείου:** Ο χρήστης θα έχει τη δυνατότητα να διαγράψει κάποιο σύνολο δεδομένων δημόσιο ή ιδιωτικό ανάλογα με την κυριότητα του αρχείου.
- **Αρχική σελίδα:** Μια αρχική σελίδα που θα περιγράφει την εφαρμογή.
- **Σελίδα Hierarchical:** Αυτή θα είναι η σελίδα που οι χρήστες θα μπορούν να εκτελούν τις υπηρεσίες που προσφέρουμε σχετικά με τον αλγόριθμο Hierarchical. Η σελίδα θα είναι προσβάσιμη από συνδεδεμένους χρήστες. Οι χρήστες θα μπορούν να επιλέξουν ποια υπηρεσία θα θελήσουν να χρησιμοποιήσουν ανάμεσα στις προσφερόμενες. Οι προσφερόμενες υπηρεσίες είναι: Προσδιορισμός τύπου μέτρησης απόστασης (linkage) και πλήθους συστάδων καθώς και ανάλυση ιεραρχικής συσταδοποίησης.
- **Προσδιορισμός προτεινόμενων τιμών(Optimal):** Ο χρήστης θα μπορεί να επιλέξει το σύνολο δεδομένων και τα χαρακτηριστικά που επιθυμεί να συμπεριληφθούν στον αλγόριθμο, και θα του επιστραφούν προτάσεις για τον τύπο μέτρησης της απόστασης και τον αριθμό συστάδων, καθώς και το αντίστοιχο γράφημα που αντιστοιχεί στις προτάσεις.
- **Εκτέλεση ιεραρχικής συσταδοποίησης(Analysis):** Ο χρήστης θα μπορεί να εφαρμόσει ιεραρχική Συσταδοποίηση δίνοντας τιμές για τον τύπο μέτρησης της απόστασης και τον αριθμό συστάδων. Θα του επιστραφεί το παραγεμισμένο δενδρόγραμμα, παραλληλόγραμμο γράφημα, καθώς και ένας πίνακας με τα δεδομένα του αρχείου και μια επιπλέον στήλη για τη συστάδα που έχουν ενταχθεί. Ο χρήστης θα έχει τη δυνατότητα να κατεβάσει τα γραφήματα σε μορφή PNG καθώς και το αρχείο των αναθέσεων σε μορφή CSV.
- **Σελίδα History:** Αυτή η σελίδα περιλαμβάνει το ιστορικό εκτέλεσης των λειτουργιών της εφαρμογής.
- **Προεπισκόπηση επιλεγμένου ιστορικού εκτέλεσης:** Μέσω του ιστορικού εκτέλεσης ο χρήστης θα μπορεί να βλέπει τα αποτελέσματα των λειτουργιών της εφαρμογής.
- **Διαγραφή επιλεγμένου ιστορικού εκτέλεσης:** Ο χρήστης θα μπορεί να διαγράψει μεμονωμένη καταγραφή ιστορικού.
- **Public API:** Θα πρέπει να παρέχεται ελεύθερη πρόσβαση σε ένα WEB API για την εκτέλεση όλων των λειτουργιών μέσω αιτημάτων HTTP.
- **Αξιολόγηση Εμπειρίας Χρήσης:** Ο διαχειριστής της εφαρμογής θα πρέπει να λαμβάνει ανατροφοδότηση από τους χρήστες μέσω του System Usability Scale (SUS), ώστε να βελτιώνεται συνεχώς η εμπειρία χρήσης της εφαρμογής.

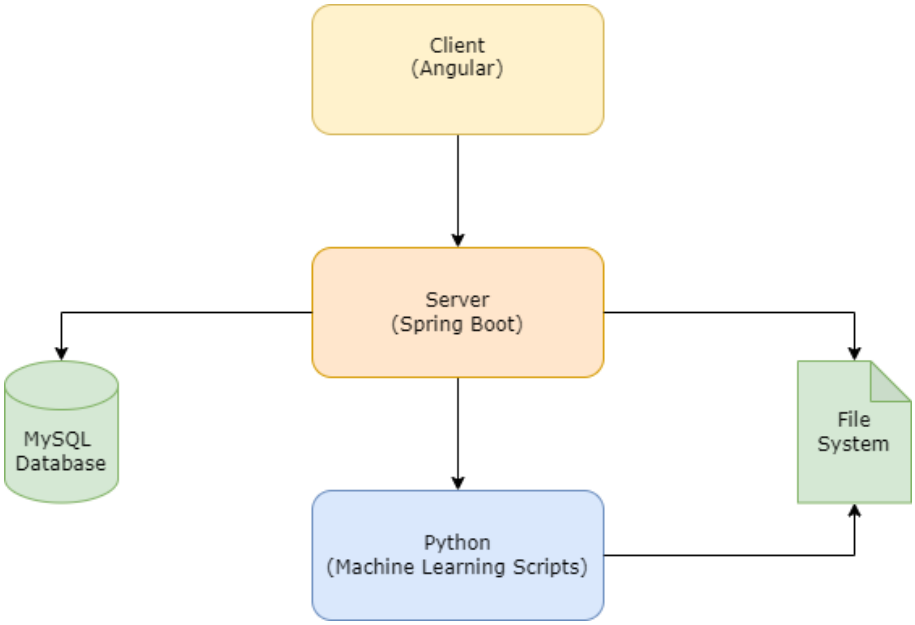
4.2 Αρχιτεκτονική του HCvision

Ο βασικός στόχος της εφαρμογής είναι η δημιουργία ενός πλήρους διαδικτυακού εργαλείου που επιτρέπει στους χρήστες να εκτελούν ανάλυση ιεραρχικής συσταδοποίησης σε σύνολα δεδομένων της επιλογής τους. Η εφαρμογή αποτελείται από ένα API με ελεύθερη πρόσβαση για όποιον επιθυμεί να το χρησιμοποιήσει, καθώς και μια υλοποίηση διεπαφής για τους χρήστες. Συγκεκριμένα, η εφαρμογή δημιουργεί ένα γράφημα που περιλαμβάνει το μέγιστο συντελεστή ασυνέπειας και τον αριθμό των συστάδων ανά τύπο μέτρησης της απόστασης. Προτείνει στον

χρήστη τον τύπο μέτρησης της απόστασης, καθώς και τον αριθμό των συστάδων που εμφανίζει τον μεγαλύτερο συντελεστή ασυνέπειας. Στη συνέχεια, ο χρήστης μπορεί να εφαρμόσει ιεραρχική συσταδοποίηση, δίνοντας τις προτεινόμενες παραμέτρους. Επιπλέον, η εφαρμογή υποστηρίζει ασύγχρονη εκτέλεση των λειτουργιών, καθώς και αποθήκευση των αποτελεσμάτων για τροφοδότηση σε μελλοντική εκτέλεση με ίδιες παραμέτρους. Οι χρήστες έχουν τη δυνατότητα να προεπισκοπούν τις προηγούμενες εκτελέσεις τους. Η ίδια διαδικασία μπορεί να γίνει μέσω ενός REST API.

4.2.1 Server

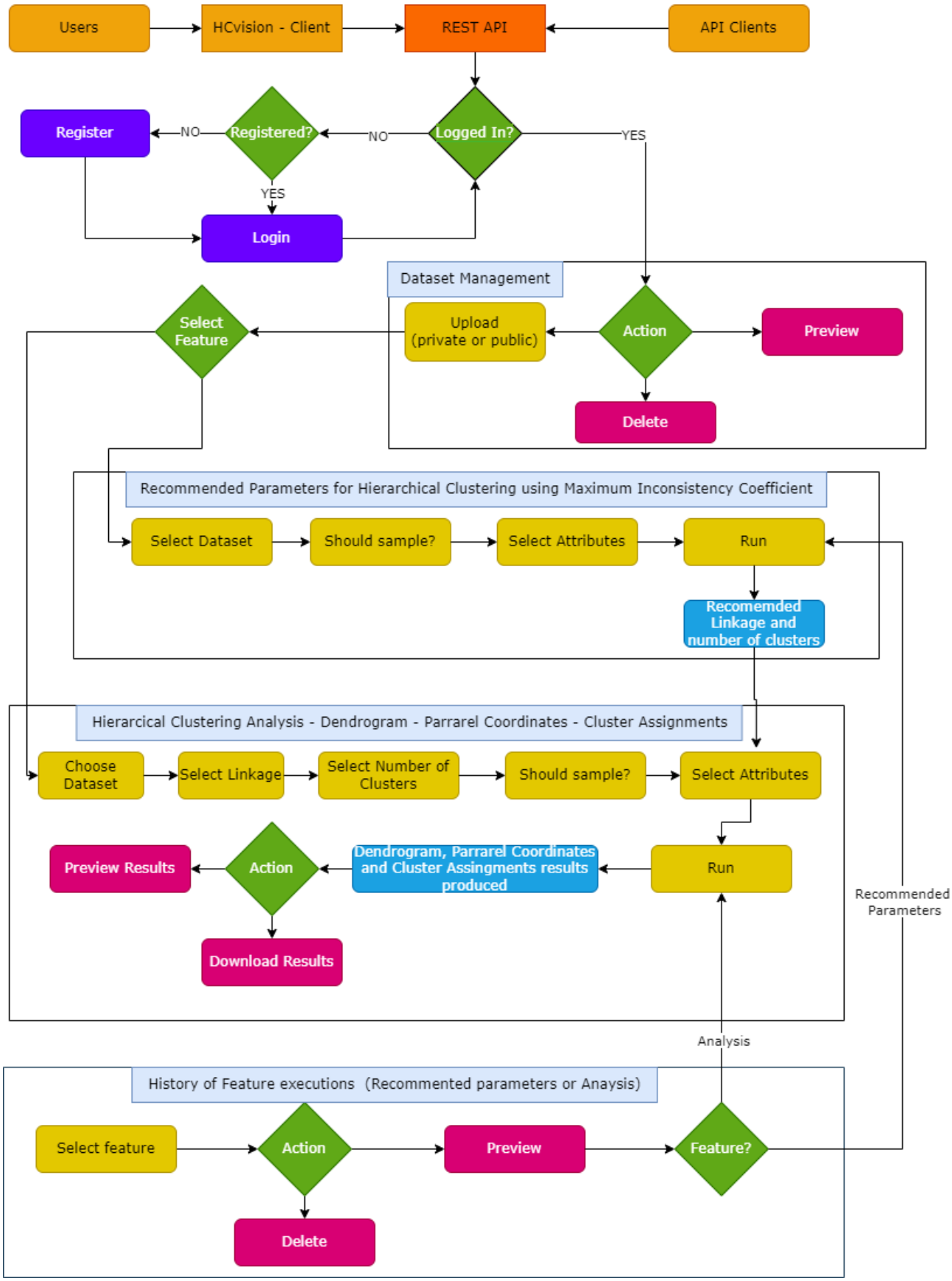
Όπως φαίνεται στο Σχήμα 4.1 ο server διαχειρίζεται τα API calls για όλες τις υπηρεσίες που προαναφέρθηκαν. Για τις υπηρεσίες της αυθεντικοποίησης αποθηκεύει τα απαραίτητα δεδομένα στην SQL βαση. Για την αποθήκευση των συνόλων δεδομένων χρησιμοποιεί το file system ενώ για την επεξεργασία τρέχει κάποια custom scripts σε python που χρησιμοποιούν τις βιβλιοθήκες της Scikit Learn και Scipy.



Σχήμα 4.1: Διάγραμμα αρχιτεκτονικής της εφαρμογής

4.2.2 Client

Ο χρήστης πρέπει να συνδεθεί με τα προσωπικά του στοιχεία (email, κωδικός πρόσβασης), και εάν δεν έχει λογαριασμό, πρέπει να εγγραφεί. Αφού συνδεθεί, θα έχει τη δυνατότητα να ανεβάσει σύνολα δεδομένων στον διακομιστή μέσω της διεπαφής. Στη συνέχεια, μπορεί να επιλέξει ένα από τα διαθέσιμα σύνολα δεδομένων και να τρέξει τη μέθοδο προσδιορισμού προτεινόμενων παραμέτρων για να λάβει τις προτεινόμενες τιμές του τύπου υπολογισμού απόστασης και αριθμού συστάδων για την Ιεραρχική Συσταδοποίηση. Εν συνεχεία, μπορεί να τρέξει τη συσταδοποίηση με τις επιλεγμένες παραμέτρους. Η διεπαφή θα εμφανίζει τα αποτελέσματα της συσταδοποίησης και θα παρέχει στον χρήστη την επιλογή να τα κατεβάσει στον υπολογιστή του. Επιπλέων ο χρήστης είναι σε θέση να επανεξετάσει τα αποτελέσματα αναλύσεων που έτρεξε στο παρελθόν. Η λογική αυτή περιγράφεται στο σχήμα 4.2



Σχήμα 4.2: Διάγραμμα Ροής

4.3 Υλοποίηση του Server

4.3.1 Βάση Δεδομένων

Στις περισσότερες περιπτώσεις, θα χρησιμοποιούσαμε τη βάση δεδομένων κυρίως για τη διαχείριση των χρηστών. Ωστόσο, οι απαιτήσεις για την ασύγχρονη εκτέλεση λειτουργιών της εφαρμογής, καθώς και η αποθήκευση των αποτελεσμάτων, δημιούργησαν την ανάγκη σχεδιασμού και υλοποίησης μιας εκτεταμένης σχεσιακής βάσης δεδομένων. Επιλέχθηκε η MySQL, καθώς αποτελεί ένα από τα πιο δημοφιλή συστήματα διαχείρισης σχεσιακών βάσεων δεδομένων (RDBMS), είναι ανοικτού κώδικα και προσφέρει υψηλή απόδοση, αξιοπιστία και ευελιξία. Μπορούμε να διακρίνουμε τη βάση δεδομένων του HCvision σε τρεις κατηγορίες και να αναλύσουμε εκτενώς κάθε μία από αυτές.

Διαχείριση των χρηστών
Οι σχετικοί πίνακες είναι ο *users* και ο *confirmation_token*. Ο πίνακας *users* (βλ. Πίνακα 4.1) αποτελείται από τα εξής πεδία

- Το πεδίο *id* λειτουργεί ως το κύριο κλειδί του πίνακα.
- Το πεδίο *first_name* αναπαριστά το όνομα του χρήστη.
- Το πεδίο *last_name* αναπαριστά το επώνυμο του χρήστη.
- Το πεδίο *email* είναι η διεύθυνση ηλεκτρονικού ταχυδρομείου που χρησιμοποιεί ο χρήστης για την εγγραφή του.
- Το πεδίο *password* αντιστοιχεί στον κωδικό του χρήστη σε κατακερματισμένη μορφή, απαραίτητος για την είσοδό του στην εφαρμογή. Ο κωδικός αποθηκεύεται σε κατακερματισμένη μορφή χρησιμοποιώντας τον αλγόριθμο Blowfish για λόγους ασφαλείας.
- Το πεδίο *role* προσδιορίζει το ρόλο του χρήστη (0 = ADMIN, 1 = USER).
- Το πεδίο *activated* καθορίζει εάν ο χρήστης έχει επιβεβαιώσει το email του.

Column	Data Type	Constraints	Description
id	bigint	NOT NULL AUTO_INCREMENT	User ID (Primary Key)
first_name	varchar(255)	DEFAULT NULL	User's first name
last_name	varchar(255)	DEFAULT NULL	User's last name
email	varchar(255)	DEFAULT NULL	User email address
password	varchar(255)	DEFAULT NULL	User password
role	tinyint	DEFAULT NULL	User role (0 or 1)
activated	bit(1)	NOT NULL	Account activation status

Πίνακας 4.1: Δομή πίνακα 'users'

Ο πίνακας *confirmation_token* (βλ. Πίνακα 4.2) αποθηκεύει τα tokens για την επιβεβαίωση των email και των αιτημάτων ανάκτησης του κωδικού πρόσβασης. Αποτελείται από τα εξής πεδία:

- Το πεδίο *id* είναι το κύριο κλειδί του πίνακα.

- Το πεδίο *type* προσδιορίζει τον τύπο του token (0 = UUID, 1 = OTP). Ως UUID αναφερόμαστε σε token επιβεβαίωσης email και ως OTP σε token αίτησης ανάκτησης του κωδικού πρόσβασης.
- Το πεδίο *token* είναι ένα τυχαίο string που χρησιμοποιείται για την επαλήθευση της επαναφοράς του κωδικού.
- Το πεδίο *confirmed_at* υποδεικνύει τη χρονική στιγμή επιβεβαίωσης.
- Το πεδίο *created_at* υποδεικνύει τη χρονική στιγμή δημιουργίας του token.
- Το πεδίο *expires_at* υποδεικνύει τη χρονική στιγμή λήξης του token.
- Το πεδίο *expiresAt* υπάρχει για να παρέχει μια ημερομηνία λήξης στο token για λόγους ασφαλείας. Του δίνουμε ως ημερομηνία την επόμενη ώρα από την ώρα του αιτήματος σε ms.
- Το πεδίο *user_id* υποδεικνύει τον χρήστη στον οποίον εκδόθηκε το token. Αποτελεί ξένο κλειδί που αναφέρεται στον πίνακα *users*.

Column	Data Type	Constraints	Description
id	bigint	NOT NULL	Token ID (Primary Key)
type	tinyint	DEFAULT NULL	Type of confirmation token
token	varchar(255)	NOT NULL	Confirmation token value
confirmed_at	datetime(6)	DEFAULT NULL	Date and time of confirmation
created_at	datetime(6)	NOT NULL	Date and time of creation
expires_at	datetime(6)	NOT NULL	Expiry date and time
user_id	bigint	NOT NULL	User ID (Foreign Key)

Πίνακας 4.2: Δομή πίνακα 'confirmation_token'

Διαχείριση των συνόλων δεδομένων

Τα αρχεία των συνόλων δεδομένων αποθηκεύονται στο σύστημα αρχείων. Παρόλα αυτά, σημαντικά στοιχεία από αυτά τα σύνολα αποθηκεύονται επίσης στη βάση δεδομένων. Αυτά τα στοιχεία περιλαμβάνουν το όνομα, τον τύπο πρόσβασης (δημόσιο ή ιδιωτικό), τον ιδιοκτήτη του συνόλου, τα αριθμητικά χαρακτηριστικά και τη διαδρομή του αρχείου στο σύστημα αρχείων (βλ. Πίνακα 4.3). Αυτή η πρακτική συμβάλλει στον περιορισμό πολλαπλών αναγνώσεων των αρχείων από το σύστημα αρχείων, οι οποίες μπορεί να είναι χρονοβόρες, καθώς και στον περιορισμό πολλαπλών αιτημάτων για επεξεργασία των συνόλων. Επιπλέον, ο διαχωρισμός των χαρακτηριστικών σε αριθμητικούς πραγματοποιείται μια φορά κατά τη μεταφόρτωση του αρχείου, και στη συνέχεια αυτή η πληροφορία προσπελαύνεται από τη βάση δεδομένων.

- Το πεδίο *id* λειτουργεί ως το κύριο κλειδί του πίνακα.
- Το πεδίο *access_type* προσδιορίζει το είδος προσβασιμότητας (0 = PUBLIC, 1 = PRIVATE).
- Το πεδίο *file_name* υποδεικνύει το όνομα του συνόλου δεδομένων.
- Το πεδίο *path* προσδιορίζει τη διαδρομή του συνόλου δεδομένων στο σύστημα αρχείων.
- Το πεδίο *numeric_cols* υποδεικνύει τα αριθμητικά χαρακτηριστικά του συνόλου δεδομένων.

- Το πεδίο *user_id* υποδεικνύει τον ιδιοκτήτη του συνόλου δεδομένων. Αποτελεί ξένο κλειδί που αναφέρεται στον πίνακα *users*.

Column	Data Type	Constraints	Description
id	bigint	NOT NULL	Dataset ID (Primary Key)
access_type	tinyint	DEFAULT NULL	Type of dataset access
file_name	varchar(255)	DEFAULT NULL	Name of the dataset file
path	varchar(255)	DEFAULT NULL	Path to the dataset
numeric_cols	varchar(255)	DEFAULT NULL	Numeric columns information
user_id	bigint	NOT NULL	User ID (Foreign Key)

Πίνακας 4.3: Δομή πίνακα 'dataset'

Διαχείριση των λειτουργιών συσταδοποίησης

Έχει υλοποιηθεί ένας πίνακας για κάθε λειτουργία του HCvision. Ο πίνακας *optimal* περιέχει πληροφορίες για τη λειτουργία περιορισμού προτεινόμενων παραμέτρων, ενώ ο πίνακας *analysis* αφορά την ιεραρχική συσταδοποίηση (βλ. Πίνακες 4.4 και 4.5). Κοινά πεδία και στους δύο πίνακες είναι:

- Το πεδίο *id* είναι το κύριο κλειδί του πίνακα.
- Το πεδίο *dataset_id* υποδεικνύει το σύνολο δεδομένων που χρησιμοποιείται. Αποτελεί ξένο κλειδί που αναφέρεται στον πίνακα *datasets*.
- Το πεδίο *user_id* υποδεικνύει τον ιδιοκτήτη του συνόλου δεδομένων. Αποτελεί ξένο κλειδί που αναφέρεται στον πίνακα *users*.
- Το πεδίο *sample* υποδεικνύει εαν ο αλγόριθμος τρέχει σε δειγμα του συνόλου δεδομένων.
- Το πεδίο *attributes* περιλαμβάνει τα χαρακτηριστικά που επιλέχθηκαν για την εκτέλεση της λειτουργίας.
- Το πεδίο *status* υποδεικνύει την κατάσταση της εκτέλεσης της λειτουργίας (0 = RUNNING, 1 = FINISHED, 2 = ERROR).
- Το πεδίο *started_at* υποδεικνύει την χρονική στιγμή που ξεκίνησε η επεξεργασία.
- Το πεδίο *ended_at* υποδεικνύει τη χρονική στιγμή λήξης της επεξεργασίας.
- Το πεδίο *duration* υποδεικνύει τη χρονική διάρκεια των δύο παραπάνω σε δευτερόλεπτα.

Επιπλέον, σε κάθε πίνακα περιλαμβάνονται πεδία που περιέχουν τη διαδρομή των αποτελεσμάτων στο σύστημα αρχείων, όπως φαίνεται και στο σχήμα.

Column	Data Type	Constraints	Description
id	bigint	NOT NULL	Optimal process ID (Primary Key)
user_id	bigint	NOT NULL	User ID (Foreign Key)
dataset_id	bigint	NOT NULL	Dataset ID (Foreign Key)
sample	bit(1)	NOT NULL	Sample indicator
attributes	varchar(255)	DEFAULT NULL	Information about attributes
inconsistency_coefficient	varchar(255)	DEFAULT NULL	Inconsistency coefficient
status	tinyint	DEFAULT NULL	Status of the optimal process
started_at	datetime(6)	DEFAULT NULL	Start date and time of the process
ended_at	datetime(6)	DEFAULT NULL	End date and time of the process
duration	bigint	NOT NULL	Duration of the process

Πίνακας 4.4: Δομή πίνακα 'optimal'

Column	Data Type	Constraints	Description
id	bigint	NOT NULL	Analysis process ID (Primary Key)
user_id	bigint	NOT NULL	User ID (Foreign Key)
dataset_id	bigint	NOT NULL	Dataset ID (Foreign Key)
sample	bit(1)	NOT NULL	Sample indicator
linkage	tinyint	DEFAULT NULL	Linkage type
num_clusters	int	NOT NULL	Number of clusters
attributes	varchar(255)	DEFAULT NULL	Information about attributes
cluster_assignment_result_path	varchar(255)	DEFAULT NULL	Path to cluster assignment result
dendrogram_result_path	varchar(255)	DEFAULT NULL	Path to dendrogram result
parallel_coordinates_result_path	varchar(255)	DEFAULT NULL	Path to parallel coordinates result
status	tinyint	DEFAULT NULL	Status of the analysis process
started_at	datetime(6)	DEFAULT NULL	Start date and time of the process
ended_at	datetime(6)	DEFAULT NULL	End date and time of the process
duration	bigint	NOT NULL	Duration of the process

Πίνακας 4.5: Δομή πίνακα 'analysis'

Διαχείριση του ιστορικού

Στον πίνακα *history*(βλ. Πίνακα 4.6) καταγράφονται όλες οι εκτελέσεις των δύο λειτουργιών που προσφέρονται στο HCvision για κάθε χρήστη.

- Το πεδίο *id* είναι το κύριο κλειδί του πίνακα.
- Το πεδίο *user_id* υποδεικνύει τον ιδιοκτήτη του συνόλου δεδομένων. Αποτελεί ξένο κλειδί που αναφέρεται στον πίνακα *users*.
- Το πεδίο *current_script* υποδεικνύει ποια από τις δύο λειτουργίες αναφέρεται το συγκεκριμένο ιστορικό.
- Το πεδίο *analysis_id* υποδεικνύει τα στοιχεία εκτέλεσης ιεραρχικής συσταδοποίησης. Αποτελεί ξένο κλειδί που αναφέρεται στον πίνακα *analysis*.
- Το πεδίο *optimal_id* υποδεικνύει τα στοιχεία εκτέλεσης περιορισμού προτεινόμενων παραμέτρων. Αποτελεί ξένο κλειδί που αναφέρεται στον πίνακα *optimal*.

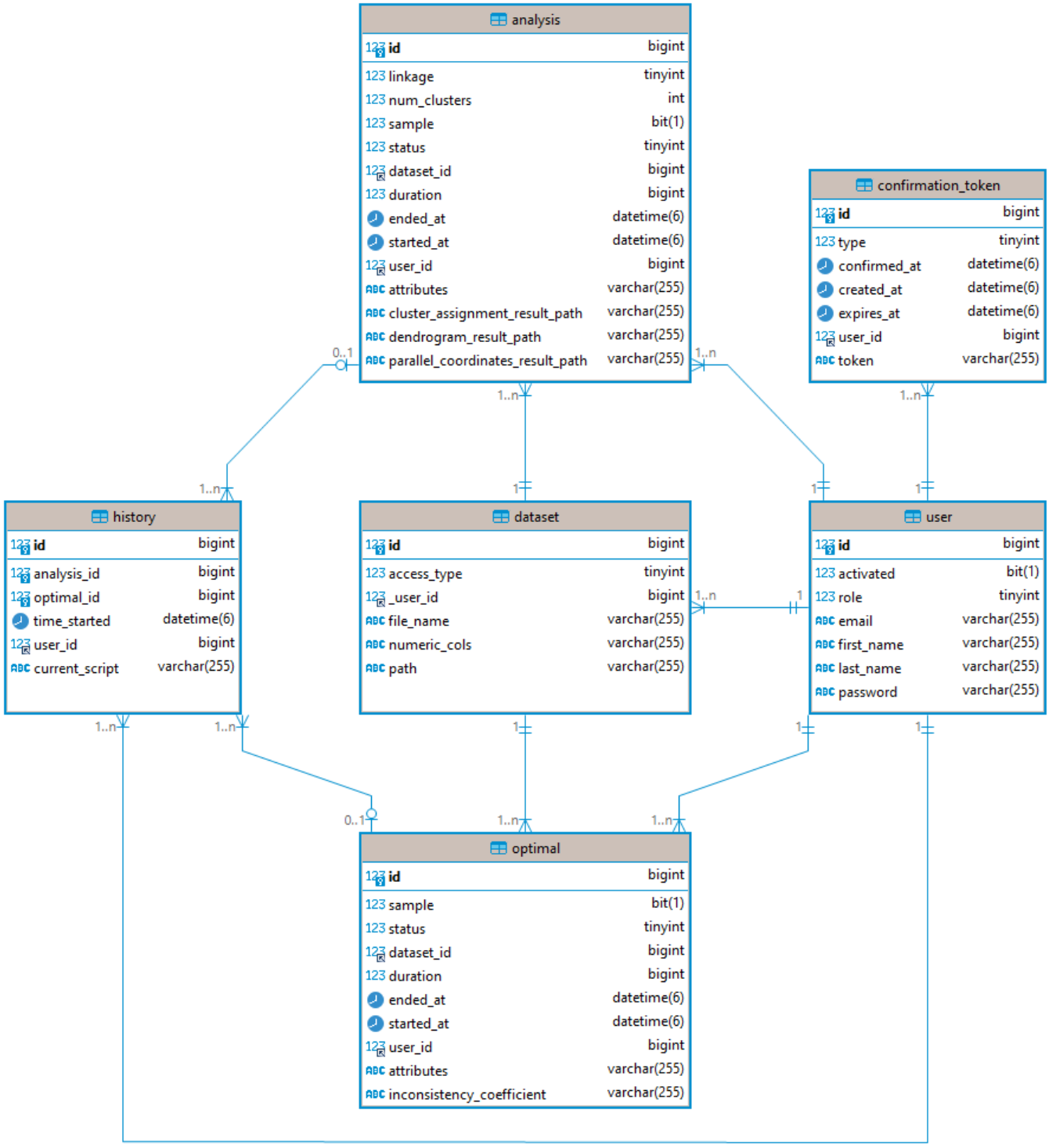
- Το πεδίο *time_started* υποδεικνύει τη χρονική στιγμή που ξεκίνησε η επεξεργασία.

Column	Data Type	Constraints	Description
id	bigint	NOT NULL	History entry ID (Primary Key)
user_id	bigint	NOT NULL	User ID (Foreign Key)
optimal_id	bigint	DEFAULT NULL	Optimal ID (Foreign Key)
analysis_id	bigint	DEFAULT NULL	Analysis ID (Foreign Key)
current_script	varchar(255)	DEFAULT NULL	Path to the current script
time_started	datetime(6)	DEFAULT NULL	Start date and time of the history entry

Πίνακας 4.6: Δομή πίνακα 'history'

Διάγραμμα οντοτήτων συσχετίσεων (ER)

Ένα διάγραμμα σχέσης οντοτήτων είναι ένα διάγραμμα που αναπαριστά σχέσεις μεταξύ οντοτήτων σε μια βάση δεδομένων. Είναι κοινώς γνωστό ως διάγραμμα ER. Ένα διάγραμμα ER στο DBMS παίζει σημαντικό ρόλο στο σχεδιασμό της βάσης δεδομένων. Το διάγραμμα ER για την εφαρμογή HCvision φαίνεται στο σχήμα 4.3.



Σχήμα 4.3: Διάγραμμα οντοτήτων συσχετίσεων

4.3.2 Παραμετροποίηση της Python

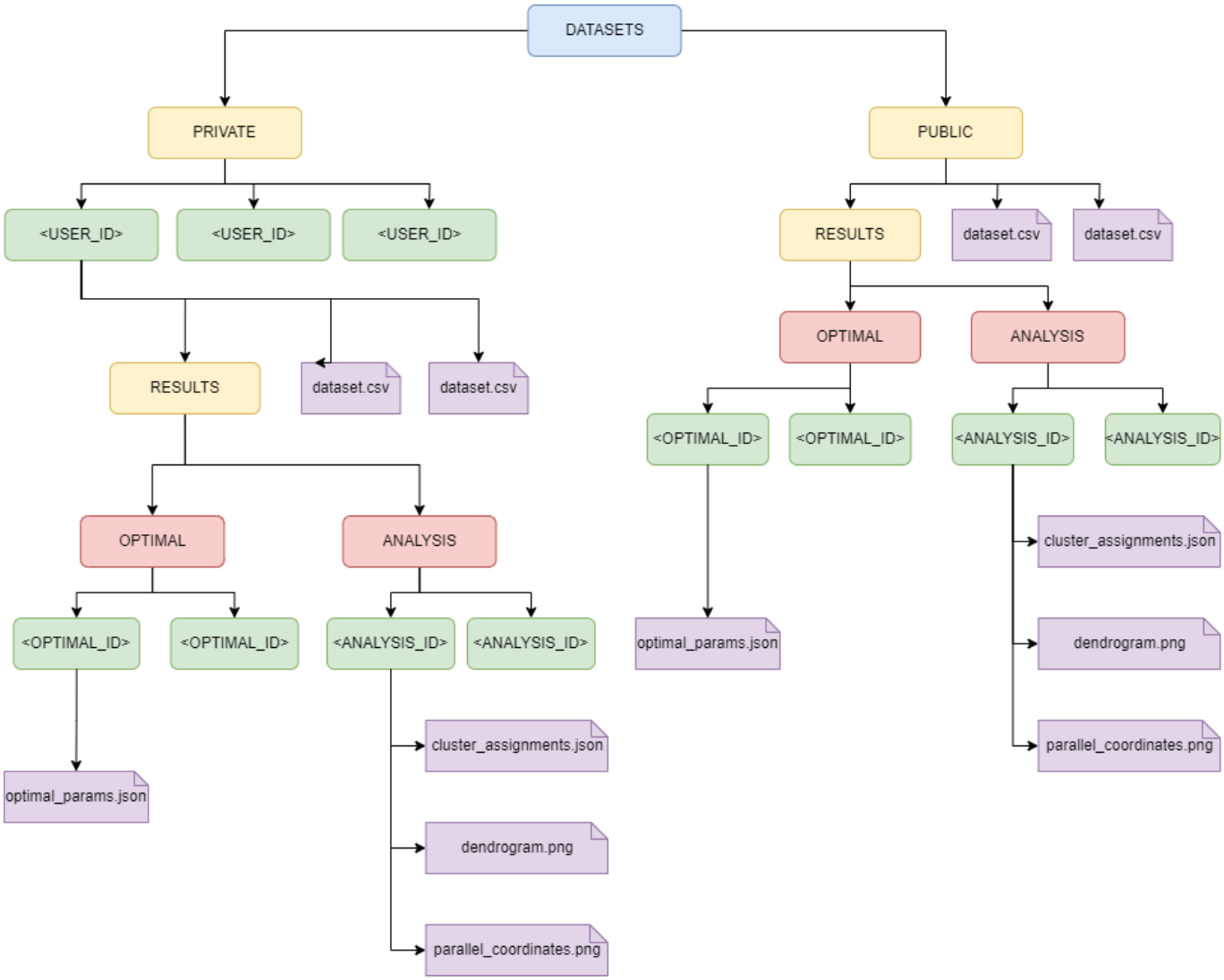
Κατά την ανάπτυξη της εφαρμογής HCvision, η διαμόρφωση του περιβάλλοντος Python στον διακομιστή ήταν ένα κρίσιμο βήμα για να εξασφαλιστεί ένα αξιόπιστο και απομονωμένο περιβάλλον εκτέλεσης. Ακολουθώντας τις βέλτιστες πρακτικές στην ανάπτυξη παραγωγής, χρησιμοποιήθηκε ένα εικονικό περιβάλλον (venv) για την Python. Αυτή η πρακτική περιλαμβάνει τη δημιουργία ενός αποκλειστικού εικονικού περιβάλλοντος για την εφαρμογή, απομονώνοντάς την από την προεπιλεγμένη εγκατάσταση Python του συστήματος. Η προσέγγιση αυτή ενισχύει την αξιοπιστία της εφαρμογής, αποτρέποντας συγκρούσεις μεταξύ διαφορετικών έργων που μοιράζονται τον ίδιο διακομιστή. Όλα τα απαιτούμενα πακέτα εγκαταστάθηκαν στη συνέχεια σε αυτό το εικονικό περιβάλλον, διασφαλίζοντας ότι η εφαρμογή βασίζεται αποκλειστικά στις ενθυλακωμένες εξαρτήσεις της. Αυτό όχι μόνο διευκολύνει την ευκολότερη διαχείριση των εξαρτήσεων, αλλά εγγυάται επίσης συνεπείς και αναπαραγώγιμες αναπτύξεις.

Η σημασία αυτής της πρακτικής υποστηρίζεται από τα πρότυπα και τις συστάσεις του κλάδου καθώς αναγνωρίζεται ευρέως στην κοινότητα ανάπτυξης λογισμικού, η χρήση εικονικών περιβαλλόντων για εξαρτήσεις συγκεκριμένου έργου σε περιβάλλοντα παραγωγής θεωρείται καλή πρακτική[43].

Δεδομένης της φύσης της εφαρμογής HCvision, η οποία περιλαμβάνει μηχανική μάθηση και ιεραρχική ομαδοποίηση με έμφαση στην οπτικοποίηση, επιλέχθηκαν και εγκαταστάθηκαν προσεκτικά συγκεκριμένα πακέτα Python. Στα αξιοσημείωτα πακέτα περιλαμβάνονταν το Pandas για την επεξεργασία δεδομένων, το Scipy για την ιεραρχική ομαδοποίηση και το Matplotlib για την οπτικοποίηση. Αυτή η επιλογή και διαμόρφωση των πακέτων έθεσε τα θεμέλια για την αποτελεσματική εκτέλεση της εφαρμογής και την απρόσκοπτη ενσωμάτωση των αλγορίθμων μηχανικής μάθησης, εξασφαλίζοντας βέλτιστες επιδόσεις στο χειρισμό της ιεραρχικής ομαδοποίησης και των σχετικών οπτικοποιήσεων.

4.3.3 Δομή File System

Για την αποθήκευση των datasets και των αποτελεσμάτων που προκύπτουν από την εκτέλεση των παρεχόμενων αλγορίθμων χρησιμοποιείται το σύστημα αρχείων. Συγκεκριμένα, τα αρχεία οργανώνονται με τέτοιο τρόπο ώστε να διαχωρίζονται τα ιδιωτικά από τα δημόσια αρχεία συνόλων δεδομένων. Επιπλέον, υπάρχει διακριτικότητα ως προς την ιδιοκτησία των αρχείων ανά χρήστη, ενώ παράλληλα κρατείται πληροφορία σχετικά με τον αλγόριθμο στον οποίο ανήκουν τα αποτελέσματα, καθώς και για το ποιο σύνολο παραμέτρων έχει εκτελεστεί. Όσον αφορά τα ιδιωτικά σύνολα δεδομένων, αυτά αποθηκεύονται σε έναν φάκελο που έχει ως όνομα το αναγνωριστικό *ID* του αντίστοιχου χρήστη στη βάση δεδομένων. Οι φάκελοι ανά χρήστη τοποθετούνται κάτω από τον κατάλογο *PRIVATE*. Αντίστοιχα, τα δημόσια σύνολα δεδομένων αποθηκεύονται στον φάκελο *PUBLIC*, διότι δεν υπάρχει συγκεκριμένη κυριότητα σε αυτά τα αρχεία. Τα αποτελέσματα αποθηκεύονται σε έναν φάκελο που έχει ως όνομα το αντίστοιχο αναγνωριστικό *ID* του script που εκτελέστηκε. Αυτός ο φάκελος τοποθετείται εντός ενός άλλου φακέλου που καθορίζεται από το είδος του script (π.χ., *OPTIMAL*, *ANALYSIS*). Στη συνέχεια, αυτός ο φάκελος είναι τοποθετημένος κάτω από τον φάκελο *RESULTS*. Για τα ιδιωτικά σύνολα δεδομένων, υπάρχει ένας φάκελος RESULTS για κάθε χρήστη, ενώ για τα δημόσια σύνολα δεδομένων τοποθετούνται κάτω από τον φάκελο PUBLIC. Στο Σχήμα 4.4 απεικονίζεται η δομή του συστήματος αρχείων όπως περιγράφηκε.



Σχήμα 4.4: Δομή File System

4.3.4 Rest API

Στον Πίνακα 4.7 βλέπουμε μια λίστα με όλα τα διαθέσιμα endpoints της εφαρμογής. Στη συνέχεια, παραθέτονται παραδείγματα HTTP αιτημάτων καθώς και οι αντίστοιχες απαντήσεις από τις πιο σημαντικές λειτουργίες του API.

Κατηγορία	**Endpoint**
Authentication	`/api/v1/auth`
POST	`/authenticate` - Σύνδεση Χρήστη
POST	`/register` - Εγγραφή Χρήστη
GET	`/confirm` - Επιβεβαίωση Email
GET	`/confirmation-link` - Αίτηση Email Επιβεβαίωσης
Users	`/API/v1/users`
GET	`/` - Ανάκτηση Πληροφοριών Χρήστη
POST	`/update` - Ενημέρωση Πληροφοριών Χρήστη
DELETE	`/delete` - Διαγραφή Χρήστη
POST	`/password/forgot` - Αίτηση Επαναφοράς Κωδικού
POST	`/password/reset` - Επαναφορά Κωδικού Χρήστη
Datasets	`/api/v1/datasets`
GET	`/` - Ανάκτηση Δημόσιων και Ιδιωτικών Dataset
DELETE	`/delete` - Διαγραφή Dataset
GET	`/download` - Λήψη Dataset
POST	`/upload` - Μεταφόρτωση Dataset
Clustering	`/api/v1/hierarchical`
GET	`/optimal` - Εκτέλεση Υπολογισμού προτεινόμενων τιμών
GET	`/analysis` - Εκτέλεση Ιεραρχικής Συσταδοποίησης
History	`/api/v1/hierarchical/history`
GET	`/` - Ανάκτηση Λίστας Ιστορικού
GET	`/{id}` - Ανάκτηση Πληροφοριών Συγκεκριμένου ιστορικού
GET	`/{id}` - Διαγραφή Ιστορικού
Results	`/api/v1/resources`
GET	`/analysis/` - Ανάκτηση Αποτελεσμάτων Συσταδοποίησης
GET	`/optimal/` - Ανάκτηση Αποτελεσμάτων προτεινόμενων παραμέτρων

Πίνακας 4.7: Κατάλογος Endpoints

Εγγραφή και αυθεντικοποίηση χρήστη

Στο Σχήμα 4.5 αρχικά παρουσιάζεται ένα HTTP αίτημα εγγραφής χρήστη, όπου παρέχονται τα στοιχεία του νέου χρήστη. Μετά την επιτυχή εγγραφή, ο χρήστης λαμβάνει ένα αντίστοιχο αντίγραφο με τα στοιχεία του χρήστη, τον ρόλο του ως χρήστης, καθώς και την ημερομηνία

εγγραφής. Στην συνέχεια παρουσιάζεται ένα HTTP αίτημα αυθεντικοποίησης χρήστη, όπου χρησιμοποιούνται τα διαπιστευτήρια του χρήστη (email και password). Μετά την επιτυχή αυθεντικοποίηση, ο χρήστης λαμβάνει ένα αντίγραφο με τα στοιχεία του, το ρόλο του, και ένα access token που χρησιμοποιείται για πρόσβαση σε ασφαλείς πόρους.

```
#User Registration

    POST /api/v1/auth/register HTTP/1.1
    Content-Type: application/json

    {
        "firstname": "John",
        "lastname": "Doe",
        "email": "johndoe@example.com",
        "password": "securepassword"
    }

    HTTP/1.1 200 OK
    Content-Type: application/json

    {
        "name": "John",
        "email": "johndoe@example.com",
        "role": "USER",
        "confirmed": false,
        "registeredAt": "2024-01-21T18:16:15.410843"
    }

#User Authentication

    POST /api/v1/auth/authenticate HTTP/1.1
    Content-Type: application/json

    {
        "email": "johndoe@example.com",
        "password": "securepassword"
    }

    HTTP/1.1 200 OK
    Content-Type: application/json

    {
        "name": "John",
        "email": "johndoe@example.com",
        "role": "USER",
        "confirmed": false,
        "access_token": "<token>",
        "issued_at": "2024-01-21T18:37:17.6691934",
        "expires_at": "2024-02-27T11:57:17.6691934"
    }
```

Σχήμα 4.5: Αιτήματα εγγραφής και αυθεντικοποίησης χρήστη

Μεταφόρτωση και ανάγνωση συνόλου δεοδομένων

Στο Σχήμα 4.6, αρχικά παρουσιάζεται ένα HTTP αίτημα μεταφόρτωσης συνόλου δεδομένων, όπου ο χρήστης αποστέλλει ένα αρχείο δεδομένων και καθορίζει τον τρόπο πρόσβασης στα

δεδομένα. Μετά την επιτυχή μεταφόρτωση, ο χρήστης λαμβάνει ένα μήνυμα επιτυχούς μεταφόρτωσης. Στην συνέχεια παρουσιάζεται ένα αίτημα ανάγνωσης συνόλου δεδομένων, όπου ο χρήστης ζητά τα δεδομένα του συνόλου με βάση το όνομα του και τον τρόπο πρόσβασης. Μετά την επιτυχή ανάγνωση, ο χρήστης λαμβάνει τα δεδομένα σε μορφή JSON, συμπεριλαμβανομένων των χαρακτηριστικών και των αντίστοιχων τιμών τους.

```
#Dataset Upload

    POST /api/v1/datasets/upload HTTP/1.1
    Content-Type: multipart/form-data
    Authorization: Bearer <token>

    {
        "file": <file contents>,
        "access_type": "PUBLIC"
    }

    HTTP/1.1 200 OK
    Content-Type: application/json

    {
        "success_msg": "File uploaded successfully."
    }

#Dataset Read

    GET /api/v1/datasets/read?dataset=iris.csv&access_type=PUBLIC HTTP/1.1
    Authorization: Bearer <token>

    HTTP/1.1 200 OK
    Content-Type: application/json

    {
        "attributes": [
            "sepal.length",
            "sepal.width",
            "petal.length",
            "petal.width"
        ],
        "dataset": [
            {
                "variety": "Setosa",
                "sepal.width": "3.5",
                "sepal.length": "5.1",
                "petal.width": ".2",
                "petal.length": "1.4"
            },
            ...
            ]
    }
```

Σχήμα 4.6: Αιτήματα δειαχείρησης των συνόλων δεδομένων

Αίτημα εύρεσης βέλτιστων παραμέτρων και ιεραρχικής ανάλυσης

Τα παραπάνω παραδείγματα αφορούν το αίτημα εύρεσης βέλτιστων παραμέτρων και την εκτέλεση ιεραρχικής ανάλυσης δεδομένων. Στο Σχήμα 4.7, παρουσιάζεται ένα HTTP αίτημα εύρεσης βέλτιστων παραμέτρων, όπου ο χρήστης καθορίζει το αρχείο δεδομένων, τον τρόπο

πρόσβασης σε αυτό, τη διεξαγωγή δειγματοληψίας και τα χαρακτηριστικά που θα ληφθούν υπόψη. Μετά την επιτυχή εκτέλεση, ο χρήστης λαμβάνει ένα μήνυμα επιβεβαίωσης, συμπεριλαμβανομένης της διάρκειας εκτέλεσης.

```
POST /api/v1/hierarchical/optimal HTTP/1.1
Content-Type: application/json
Authorization: Bearer <token>

{
    "filename": "iris.csv",
    "access_type": "PUBLIC",
    "sample": true,
    "attributes": ["sepal.length", "sepal.width", "petal.length", "petal.width
    "]
}

HTTP/1.1 200 OK
Content-Type: application/json

{
    "id": 1,
    "dataset": {
        "dataset": "iris.csv",
        "access_type": "PUBLIC"
    },
    "attributes": ["petal.length", "petal.width", "sepal.length", "sepal.width
    "],
    "sample": true,
    "status": "FINISHED",
    "duration": 2
    }
```

Σχήμα 4.7: Αίτημα εύρεσης βέλτιστων παραμέτρων

Στο Σχήμα 4.8 παρουσιάζεται ένα αίτημα εκτέλεσης ιεραρχικής ανάλυσης δεδομένων, όπου ο χρήστης καθορίζει το αρχείο δεδομένων, τον τρόπο πρόσβασης σε αυτό, τον τρόπο σύνδεσης (linkage), τον αριθμό των clusters, τη διεξαγωγή δειγματοληψίας και τα χαρακτηριστικά που θα ληφθούν υπόψη. Μετά την επιτυχή εκτέλεση, ο χρήστης λαμβάνει ένα μήνυμα επιβεβαίωσης, συμπεριλαμβανομένης της διάρκειας εκτέλεσης.

```
POST /api/v1/hierarchical/analysis HTTP/1.1
Content-Type: application/json
Authorization: Bearer <token>

{
    "filename": "iris.csv",
    "access_type": "PUBLIC",
    "linkage": "single",
    "n_clusters": "3",
    "sample": true,
    "attributes": ["sepal.length", "sepal.width", "petal.length", "petal.width
   "]
}

HTTP/1.1 200 OK
Content-Type: application/json

{
    "id": 1,
    "dataset": {
        "dataset": "iris.csv",
        "access_type": "PUBLIC"
    },
    "linkage": "ward",
    "n_clusters": 3,
    "attributes": ["petal.length", "petal.width", "sepal.length", "sepal.width
   "],
    "sample": true,
    "status": "FINISHED",
    "duration": 6
}
```

Σχήμα 4.8: Αίτημα εκτέλεσης ιεραρχικής ανάλυσης

Αίτηματα ανάκτησης ιστορικού εκτέλεσης

Στο Σχήμα 4.9, αρχικά παρουσιάζεται ένα αίτημα για τη λίστα των προηγούμενων εκτελέσεων, όπου επιστρέφονται οι πληροφορίες όπως το ID της εκτέλεσης, το σενάριο που εκτελέστηκε και η ημερομηνία έναρξης. Στην συνέχεια, παρουσιάζεται ένα αίτημα για την ανάκτηση λεπτομερειών μιας συγκεκριμένης εκτέλεσης. Εδώ επιστρέφονται πληροφορίες για το σενάριο που εκτελέστηκε, την ημερομηνία έναρξης, καθώς και λεπτομέρειες για την εκάστοτε διαδικασία, όπως το ID, το σύνολο δεδομένων, τον τρόπο πρόσβασης, τα χαρακτηριστικά, η κατάσταση και η διάρκεια της εκτέλεσης.

```
# History list

    GET /api/v1/hierarchical/history HTTP/1.1
    Authorization: Bearer <token>

    HTTP/1.1 200 OK
    Content-Type: application/json

    [
        {
            "id": "1",
            "script": "Optimal",
            "started_at": "2024-01-21T20:28:10.762882"
        },
        {
            "id": "2",
            "script": "Analysis",
            "started_at": "2024-01-21T20:29:31.304353"
        }
    ]

#Individual history

    GET /api/v1/hierarchical/history/1 HTTP/1.1
    Authorization: Bearer <token>

    HTTP/1.1 200 OK
    Content-Type: application/json

    {
        "id": "1",
        "current_script": "Optimal",
        "Started_at": "2024-01-21T20:28:10.762882",
        "optimal": {
            "id": 1,
            "dataset": {
                "dataset": "iris.csv",
                "access_type": "PUBLIC"
            },
            "attributes": [
                "petal.length",
                "petal.width",
                "sepal.length",
                "sepal.width"
            ],
            "sample": true,
            "status": "FINISHED",
            "duration": 2
        },
        "analysis": null
    }
```

Σχήμα 4.9: Αιτήματα ανάκτησης ιστορικού

Αίτηματα ανάκτησης αποτελεσμάτων

Στο Σχήμα 4.10, αρχικά παρουσιάζεται ένα αίτημα ανάκτησης των βέλτιστων παραμέτρων και στην συνέχεια, παρουσιάζεται ένα αίτημα ανάκτησης των αναθέσεων συστάδων, όπου επιστρέφονται οι τιμές των χαρακτηριστικών για κάθε δείγμα, καθώς και η ετικέτα της συ-

στάδας στην οποία ανήκει το δείγμα.

```
#Optimal parameters

    GET /api/v1/resources/optimal/1?resource=optimal_params HTTP/1.1
    Authorization: Bearer <token>

    HTTP/1.1 200 OK
    Content-Type: application/json

    {
        "all_results": [
            {
                "linkage": "ward",
                "clusters": 3,
                "max_inconsistency": 6.399406819518539
            },
            {
                "linkage": "complete",
                "clusters": 3,
                "max_inconsistency": 3.2109188716004646
            },
            {
                "linkage": "average",
                "clusters": 3,
                "max_inconsistency": 1.7855664820227883
            },
            {
                "linkage": "single",
                "clusters": 3,
                "max_inconsistency": 0.7348469228349535
            }
        ],
        "best_linkage": "ward",
        "best_clusters": 3
    }

#Cluster assignments

    GET /api/v1/resources/analysis/1?resource=cluster_assignments HTTP/1.1
    Authorization: Bearer <token>

    HTTP/1.1 200 OK
    Content-Type: application/json

    [
        {
            "sepal.length": 5.1,
            "sepal.width": 3.5,
            "petal.length": 1.4,
            "petal.width": 0.2,
            "variety": "Setosa",
            "Cluster_Labels": 1
        },
        ...
    ]
```

Σχήμα 4.10: Αιτήματα ανάκτησης αποτελεσμάτων

4.3.5 Προσδιορισμός Μεθόδου Σύνδεσης Συστάδων

Όπως αναφέραμε στα προηγούμενα κεφάλαια για να εκτελεστεί Ιεραρχική συσταδοποίηση σαν απαραίτητη προυπόθεση έχουμε τον καθορισμό των παραμέτρων μεθόδου σύνδεσης και πλήθους συστάδων. Ας αναλύσουμε τον σχετικό κώδικα (βλ. Κώδικα 4.1).

```
public Optimal.ProjectOptimal getOptimalParams(OptimalRequest request, String jwt) {
    User user = userService.getUserFromJwt(jwt);

    Dataset dataset = datasetService.getDataset(request.getFilename(), request.getAccessType(),
    user);
    if (dataset == null) throw new NotFoundException("Dataset not found");

    if (invalidParams(dataset, request.getAttributes()))
        throw new BadRequestException("Invalid attributes selected");

    List<Optimal.ProjectOptimal> reRun = optimalService.getOptimalReRun(user, dataset,
            request.isSample(), DatasetUtils.sortAttributes(request.getAttributes()));

    if (!reRun.isEmpty()) return reRun.get(0);

    Optimal optimal = optimalService.createOptimal(new Optimal(user, dataset, request.isSample(),
            DatasetUtils.sortAttributes(request.getAttributes()), ResultStatus.RUNNING));

    historyService.keepHistory(user, optimal);

    String command = "python " +
            getPythonScriptPath(OPTIMAL_SCRIPT) + " " +
            getBaseResultPathByPythonScript(optimal) + " " +
            "\"" + dataset.getPath() + "\" " +
            (request.isSample() ? "--sampling " : "") +
            DatasetUtils.encloseInDoubleQuotes(request.getAttributes()).replace(",", " ");

    maybeCreateResultDirectory(optimal);
    asyncPythonService.runScript(optimal, command);

    log.info("Python script execution started - ScriptID: {}", optimal.getId());
    return optimalService.refresh(optimal.getId());
}
```

Κώδικας 4.1: Μέθοδος getOptimalParams

Αφού ο server λάβει το HTTP αίτημα για την εύρεση προτεινόμενων τιμών, καλεί τη μέθοδο getOptimalParams. Αυτή η μέθοδος δέχεται ως παραμέτρους το OptimalRequest, το οποίο αποτελεί ένα αντικείμενο που περιέχει τις παραμέτρους του σώματος του αιτήματος, καθώς και το JWT κεφαλίδα αυθεντικοποίησης.

Στη συνέχεια, αποσπάται το επιλεγμένο σύνολο δεδομένων, τα χαρακτηριστικά και το flag που καθορίζει εάν ο χρήστης θέλει να εξετάσει ένα δείγμα του συνόλου ή ολόκληρο. Επίσης, από το JWT αποσπάται ο χρήστης που έκανε το αίτημα. Με αυτές τις πληροφορίες πραγματοποιούνται οι σχετικοί έλεγχοι εγκυρότητας και σε περίπτωση λανθασμένων τιμών, κατάλληλα μηνύματα επιστρέφονται.

Σε επόμενο στάδιο ελέγχεται εάν έχει γίνει ίδιο αίτημα στο παρελθόν. Σε περίπτωση που έχει γίνει, επιστρέφεται, ενώ αν όχι, δημιουργείται ένα νέο optimal session και αποθηκεύεται στη βάση. Σε αυτό το σημείο αποθηκεύεται επίσης στη βάση και το ιστορικό.

Τέλος, κατασκευάζεται η Python εντολή παίρνοντας όλες τις σχετικές παραμέτρους και καλείται η υπηρεσία ασύγχρονης εκτέλεσης του Python script (βλ. Κώδικα 4.2).

```
protected void runScript(PythonScript pythonScript, String command) {
    Runnable script;

    if (pythonScript instanceof Optimal) {

```

```
        script = () -> {
            try {
                ((Optimal) pythonScript).setStartedAt(LocalDateTime.now());
                ProcessBuilder processBuilder = new ProcessBuilder(command.split(" "));
                processBuilder.redirectErrorStream(true);
                Process process = processBuilder.start();

                StringBuilder output = new StringBuilder();
                BufferedReader reader = new BufferedReader(new InputStreamReader(process.
    getInputStream()));
                String line;
                while ((line = reader.readLine()) != null) {
                    output.append(line).append("\n");
                }

                int exitCode = process.waitFor();
                if (exitCode != 0) {
                    log.error("Python script execution failed for Optimal script - ScriptID: {}."
     +
                    "Exit code: {}. Output: {}",pythonScript.getId(), exitCode, output);
                    throw new IOException("Python script execution failed with exit code " +
    exitCode);
                } else {
                    log.info("Python script executed successfully for Optimal script - ScriptID:
    {}",
                    pythonScript.getId());
                }
                optimalService.saveResults((Optimal) pythonScript);

            } catch (IOException | InterruptedException e) {
                log.error("Error executing Python script for Optimal script - ScriptID: {}. Error
    : {}",
                pythonScript.getId(), e.getMessage());
                optimalService.informError((Optimal) pythonScript);
            }

        };
    }
    service.submit(script);
```

Κώδικας 4.2: Μέθοδος runScript

Δημιουργεί ένα αντικείμενο τύπου Runnable, το οποίο περιέχει το κομμάτι κώδικα που είναι υπεύθυνο για να ξεκινήσει μια διεργασία Python, χρησιμοποιώντας την εντολή που κατασκευάστηκε προηγουμένως από το αίτημα, με τη βοήθεια του ProcessBuilder. Τέλος, τοποθετεί το Runnable στη λίστα του thread pool και μόλις ένα thread είναι ελεύθερο, το εκτελεί.

Όσον αφορά το Python script, χρησιμοποιούνται βιβλιοθήκες για machine learning, όπως οι scikit-learn και scipy, καθώς και το matplotlib για την κατασκευή των γραφημάτων (βλ. Κώδικα 4.3).

```
import pandas as pd
import argparse
import json
import os
from scipy.cluster.hierarchy import linkage, fcluster
import numpy as np

def find_optimal_clusters(Z, max_d):
    clusters = fcluster(Z, t=max_d, criterion='distance')
    return len(np.unique(clusters))

def find_best_linkage_and_clusters(result_path, dataset_path, sampling, attributes):
    if dataset_path.endswith('.csv'):
        if sampling:
            dataset = pd.read_csv(dataset_path, nrows=1000)
        else:
            dataset = pd.read_csv(dataset_path)
    else:
        if sampling:
            dataset = pd.read_excel(dataset_path, nrows=1000)
```

```
        else:
            dataset = pd.read_excel(dataset_path)

    selected_attributes = dataset[attributes]

    all_results = []
    best_linkage = None
    best_clusters = 0

    for linkage_method in ['ward', 'complete', 'average', 'single']:
        Z = linkage(selected_attributes, linkage_method)
        distances = Z[:, 2]
        diff_distances = np.diff(distances, 2)
        max_d = distances[np.argmax(diff_distances)]

        n_clusters = find_optimal_clusters(Z, max_d)

        result = {
            "linkage": linkage_method,
            "clusters": n_clusters,
            "max_inconsistency": max_d
        }
        all_results.append(result)
        if n_clusters > best_clusters and max_d > 0:
            best_linkage = linkage_method
            best_clusters = n_clusters
    result = {
        "all_results": all_results,
        "best_linkage": best_linkage,
        "best_clusters": best_clusters
    }

    predict_json_path = os.path.join(result_path, "optimal_params.json")
    with open(predict_json_path, 'w') as f:
        json.dump(result, f, indent=4)

if __name__ == '__main__':
    parser = argparse.ArgumentParser()
    parser.add_argument("result_path", type=str)
    parser.add_argument("dataset_path", type=str)
    parser.add_argument("--sampling", action="store_true", default=False)
    parser.add_argument("attributes", type=str, nargs='+')
    args = parser.parse_args()
    find_best_linkage_and_clusters(args.result_path, args.dataset_path, args.sampling, args.
    attributes)
```

Κώδικας 4.3: Python Script εύρεσης προτεινόμενων παραμέτρων

Αρχικά, καλείται η μέθοδος find_best_linkage_and_clusters, στην οποία περνιούνται όλες οι παράμετροι. Το script ελέγχει εάν το αρχείο είναι σε έγκυρη μορφή. Εάν ο χρήστης επιθυμεί δειγματοληψία, το αρχείο διαβάζεται από τη διαδρομή που δόθηκε ως παράμετρος, περιορίζοντας το σύνολο δεδομένων μόνο στις στήλες/χαρακτηριστικά που επιλέχθηκαν.

Στη συνέχεια, δημιουργείται ένας πίνακας με τις διαφορές των αποστάσεων. Κατόπιν, καταχωρείται η απόσταση που βρίσκεται σε εκείνο το ύψος όπου παρατηρείται η μεγαλύτερη διαφορά. Με χρήση της συνάρτησης fcluster, γίνεται η αποσύνθεση του δενδρογράμματος στις συνενώσεις που δεν ξεπερνούν το όριο της μέγιστης απόστασης, παίρνοντας τον αριθμό των συστάδων που δημιουργήθηκαν για το συγκεκριμένο linkage.

Η διαδικασία αυτή επαναλαμβάνεται για όλα τα linkages, επιλέγοντας τις προτεινόμενες τιμές βάσει του μεγίστου αριθμού μη-μοναδικών συστάδων. Τα αποτελέσματα αποθηκεύονται σε μορφή JSON στο filesystem, προκειμένου να είναι προσβάσιμα από τα endpoints ανάκτησης αποτελεσμάτων.

4.3.6 Ανάλυση Ιεραρχικής Συσταδοποίησης

Η διαδικασία εκτέλεσης ιεραρχικής Συσταδοποίησης script γίνεται με τον ίδιο τρόπο που αναλύσαμε στην προηγούμενη ενότητα οπότε ας αναλύσουμε το ίδιο το python script για την ιεραρχική Συσταδοποίηση(βλ. Κώδικα 4.4).

```python
import pandas as pd
from pandas.plotting import parallel_coordinates
from scipy.cluster.hierarchy import dendrogram, linkage
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import AgglomerativeClustering
import matplotlib.pyplot as plt
import matplotlib.colors as mcolors

sys.setrecursionlimit(100000000)

def perform_hierarchical_clustering(result_path, dataset_path, linkage_type, num_clusters,
    sampling, attributes):

    //Identical file proccesing as the optimal script

    selected_data = dataset[attributes]
    data_standardized = StandardScaler().fit_transform(selected_data)

    clustering = AgglomerativeClustering(n_clusters=num_clusters, linkage=linkage_type)
    cluster_labels = clustering.fit_predict(data_standardized)
    dataset['Cluster_Labels'] = cluster_labels

    linkage_matrix = linkage(data_standardized, method=linkage_type)

    plt.figure(figsize=(15, 8))
    dendrogram(linkage_matrix, labels=cluster_labels, color_threshold=1.5)
    plt.title(f'{linkage_type} Hierarchical Clustering Dendrogram')
    dendrogram_path = os.path.join(result_path, "dendrogram.png")
    plt.savefig(dendrogram_path)
    plt.close()

    custom_cmap = plt.get_cmap('viridis')
    colors = [custom_cmap(i / num_clusters) for i in range(num_clusters)]
    hex_colors = [mcolors.to_hex(color) for color in colors]
    if len(selected_data) > 1:
        plt.figure(figsize=(15, 8))
        parallel_coordinates(selected_data.assign(Cluster_Labels=dataset['Cluster_Labels']),
        'Cluster_Labels', color=hex_colors)
        plt.title('Parallel Coordinates Plot')
        parallel_coordinates_path = os.path.join(result_path, "parallel_coordinates.png")
        plt.savefig(parallel_coordinates_path)
        plt.close()

    dataset['Cluster_Labels'] = cluster_labels
    json_data = dataset.to_json(orient='records')

    analysis_json_path = os.path.join(result_path, "cluster_assignments.json")
    with open(analysis_json_path, 'w') as json_file:
        json_file.write(json_data)

if __name__ == "__main__":
    parser = argparse.ArgumentParser()
    parser.add_argument("result_path")
    parser.add_argument("dataset_path")
    parser.add_argument("linkage_type")
    parser.add_argument("num_clusters", type=int)
    parser.add_argument("--sampling", action="store_true", default=False)
    parser.add_argument("attributes", nargs='+')

    args = parser.parse_args()
    perform_hierarchical_clustering(args.result_path, args.dataset_path, args.linkage_type, args.
    num_clusters, args.sampling,
    args.attributes)
```

Κώδικας 4.4: Python script ανάλυσης ιεραρχικής συσταδοποίησης

Το python scipt για την ιεραρχική συσταδοποίηση αρχικά φορτώνει το αρχείο δεδομένων και περιορίζει τις στήλες στα επιλεγμένα χαρακτηριστικά, ακριβώς όπως και το script των προτεινόμενων τιμών. Στη συνέχεια, δημιουργεί τις αναθέσεις των συστάδων, εκτελώντας τον αλγόριθμο ιεραρχικής συσταδοποίησης και χρησιμοποιώντας τον επιλεγμένο τύπο σύνδεσης (linkage) και τον αριθμό των επιθυμητών συστάδων που καθορίζει ο χρήστης. Οι αναθέσεις προστίθενται ως επιπλέον στήλη στο σύνολο δεδομένων.
Σε επόμενο επίπεδο, χρησιμοποιώντας τον πίνακα σύνδεσης (linkage matrix), δημιουργείται το δενδρόγραμμα και το γράφημα παράλληλων συντεταγμένων. Τέλος, τα γραφήματα αποθηκεύονται σε μορφή PNG και οι αναθέσεις αποθηκεύονται σε μορφή JSON στο σύστημα αρχείων, προκειμένου να είναι προσβάσιμα από τα σημεία ανάκτησης αποτελεσμάτων.

4.4 Υλοποίηση του Client

Η ανάπτυξη του client αποτελεί σημαντικό κομμάτι της εφαρμογής καθώς ο πρωταρχικός στόχος είναι να φερουμαι χρήστες με ελάχιστη ειδικευμένη γνώση σε επαφή με την ανάλυση δεδομένων και συγκεκριμένα την ιεραρχική συσταδοποίηση. Επομένως η ανάγκη για μια αποδοτικη και εύχρηστη διεπαφή είναι ζωτικής σημασίας. Για την υλοποίηση της διεπαφής αξιοποιήθηκαν σύγχρονες τεχνολογίες, συμπεριλαμβανομένων των Angular, Typescript, HTML και CSS.

Ας εισχωρήσουμε στις λεπτομέρειες για να κατανοήσουμε καλύτερα τον τρόπο με τον οποίο αυτή η υλοποίηση συμβάλλει στη συνολική λειτουργία και απόδοση της εφαρμογής HCvision.

4.4.1 Οργάνωση Αρχείων

Η δομή ενός Angular project ακολουθεί έναν οργανωμένο και συνεκτικό τρόπο, παρέχοντας ένα πλαίσιο για την ανάπτυξη δυναμικών ιστοσελίδων και εφαρμογών. Η δομή αυτή βοηθάει στην οργάνωση του κώδικα, την διαχείριση διάφορων λειτουργιών και τη διατήρηση μιας σαφής εικόνα του project. Ας ρίξουμε μια ματιά στην δομή του project (βλ. Σχήμα 4.11):

```
src
├── app
│   ├── components
│   ├── directives
│   ├── models
│   ├── modules
│   ├── services
│   ├── app.component.css
│   ├── app.component.html
│   ├── app.component.ts
│   └── app.module.ts
└── assets
    ├── index.html
    ├── main.ts
    └── styles.css
```

Σχήμα 4.11: Δομή angular project

Ενα angular project περιεχει:

- Συστατικά (Components): Αποτελούν τα βασικά στοιχεία κατασκευής που διαχειρίζονται το User Interface (UI), ενθυλακώνοντας HTML, CSS και λογική για συγκεκριμένα τμήματα λειτουργικότητας ή προβολές. Στο HCvision κάθε σελίδα αποτελεί ένα ξεχωριστό component με τα αντίστοιχα αρχεία για το λογικό μέρος(TypeScript), UI (HTML) και styling (CSS)(βλ. Σχήμα 4.12).

```
home
├── home.component.css
├── home.component.html
└── home.component.ts
```

Σχήμα 4.12: Αρχεία component

- Υπηρεσίες (Services): Αναλαμβάνουν τον ρόλο της κεντρικής διαχείρισης λειτουργικότητας και δεδομένων, παρέχοντας επαναχρησιμοποιήσιμες υπηρεσίες για τα συστατικά. Για παράδειγμα στην περίπτωσή μας έχουν συγκεντρωθεί όλα τα http αιτήματά προς τον server (βλ. Σχήμα 4.13).

```
services
├── auth.service.ts
├── dataset.service.ts
├── hierarchical.service.ts
├── history.service.ts
├── resource.service.ts
├── snackbar.service.ts
└── user.service.ts
```

Σχήμα 4.13: Αρχεία service

- Ενότητες (Modules): Δομούν την εφαρμογή σε λειτουργικές μονάδες, οργανώνοντας συστατικά, υπηρεσίες και άλλα στοιχεία. Το HCvision περιεχει modules όπως το routing που είναι υπευθυνω για την δρομολόγηση των σελίδων , interseptor το οποίο προσθετη την κεφαλίδα αργεντινοποίησης σε οποιαδήποτε http αίτημα γίνεται καθώς και το authGuard που επιβάλει αυθεντικοποιηση σε επιλεγμένες σελίδες (βλ. Σχήμα 4.14).

```
modules
├── auth.guard.ts
├── dataset.interceptor.ts
├── AuthModule.ts
└── routing.module.ts
```

Σχήμα 4.14: Αρχεία modules

- Μοντέλα (Models): Προσδιορίζουν τη δομή δεδομένων που χρησιμοποιούνται, προσφέροντας έλεγχο και προτυποποίηση των δεδομένων. Χρησιμοποειουνται απο το HCvision για να μετατρεπουν json απαντησεις απο τον server σε typescript objects.
- Πόροι (Assets): Στατικά αρχεία (όπως εικόνες, γραμματοσειρές) που ενσωματώνονται στην εφαρμογή, αποθηκευμένα συνήθως στον φάκελο assets.

4.4.2 Διαδικασία αυθεντικοποίησης χρήστη.

Όταν ένας χρήστης συνδεθεί στο σύστημα επιστρέφεται στην απάντηση του server ένα authorization token. Στη συνέχεια αποθηκεύει στο session του browser το συγκεκριμένο token. Αυτό εξυπηρετεί τον χρήστη όντας αποθηκευμένο στον browser την επόμενη φορά μέσα στο χρονικό διάστημα που θα είναι έγκυρο το token, να μην χρειαστεί να ξανά ακολουθήσει την διαδικασία σύνδεσης (βλ. Κώδικα 4.5).

```
login() {
   this.authService.login({email: this.email, password: this.password})
     .subscribe(response => {
       this.router.navigate(['']);
       this.authService.setToken(response.access_token);
     });
 }

setToken(access_token: string): void {
   sessionStorage.setItem('access_token', access_token);
   this.router.navigateByUrl(this.redirectUrl || '/');
   this.redirectUrl = null;
 }
```

Κώδικας 4.5: Αυθεντικοποίηση χρήστη

4.4.3 Http Αιτήματα

Τα HTTP αιτήματα αποστέλλονται με τη βοήθεια της βιβλιοθήκης HttpClient, η οποία αποτελεί βασικό μέρος της Angular. Για να αποστείλει αιτήσεις HTTP ο client προς τον server, χρησιμοποιείται η βιβλιοθήκη HttpClient.

Αρχικά, ο client δημιουργεί μια αίτηση HTTP. Συνήθως, μέσω της dependency injection, έχει πρόσβαση σε ένα instance της κλάσης HttpClient, ας την ονομάσουμε http. Στη συνέχεια, καλεί την αντίστοιχη μέθοδο για τον επιθυμητό τύπο HTTP αιτήματος, όπως http.get(), http.post() κλπ., καθορίζοντας το URL του endpoint, τα query params και το body. Στο Κώδικα 4.6 εικόνα βλέπουμε ένα παράδειγμα αιτήματος για την ανάγνωση ενός συνόλου δεδομένων.

```
readDataset(dataset: Dataset): Observable<DatasetResponse> {
  const url = `${this.baseUrl}/read?dataset=${dataset.dataset}&access_type=${dataset.
  access_type}`;
  return this.http.get<DatasetResponse>(url);
}
```

Κώδικας 4.6: Αίτημα ανάγνωσης συνόλου δεδομένων

Στη συνέχεια, ο διακομιστής επεξεργάζεται το αίτημα και αποστέλλει πίσω την ανταπόκρισή του, η οποία περιέχει τα απαιτούμενα δεδομένα. Μόλις η ανταπόκριση ληφθεί από τον client, μπορεί να γίνει επεξεργασία των δεδομένων, όπως η απόδοσή τους σε στοιχεία της εφαρμογής για εμφάνιση στο χρήστη (βλ. Κώδικα 4.7).

```
previewDataset(dataset: Dataset) {
  this.datasetPreviewLoading = true;
  if (dataset) {
    this.selectedDataset = dataset;
    this.datasetService.readDataset(dataset).subscribe({
        next: (response) => {
          this.datasetPreviewLoading = false;
          this.jsonData = response.dataset;
        },
        error: (error) => {
          console.error('Error fetching JSON data:', error);
        }
      }
    );
  }
```

```
16   }
```

Κώδικας 4.7: Διαχείρηση απάντησης αιτήματος ανάγνωσης συνόλου δεδομένων

4.4.4 Ασύγχρονη ανάκτηση αποτελεσμάτων

Ένα από τα κυριότερα χαρακτηριστικά του HCvision είναι η ασύγχρονη εκτέλεση και ανάκτηση των αποτελεσμάτων της ιεραρχικής ανάλυσης. Αυτό σημαίνει ότι τα αποτελέσματα εμφανίζονται στη διεπαφή όταν αυτά είναι έτοιμα. Αυτό επιτυγχάνεται στέλνοντας διαδοχικά αιτήματα κάθε ένα δευτερόλεπτο για την ανάκτηση της κατάστασης της python διεργασίας (βλ. Κώδικα 4.8).

```
1   private pollForStatus(requestData: any): void {
2     this.hierarchicalService.runAnalysis(requestData).subscribe((result) => {
3
4       if (result.status === 'RUNNING') {
5         setTimeout(() => this.pollForStatus(requestData), 1000);
6       } else if (result.status === 'FINISHED') {
7         this.duration = result.duration;
8         this.getAnalysisResults(result.id);
9       } else if (result.status === 'ERROR') {
10        this.error = true;
11        console.error('Unexpected status:', result.status);
12      }
13    });
14  }
```

Κώδικας 4.8: Ασύγχρονη ανάκτηση αποτελεσμάτων

4.4.5 Γραφικό περιβάλλον

Για την υλοποίηση του γραφικού περιβάλλοντος (UI) χρησιμοποιήθηκε η Angular Material η οποία είναι μια βιβλιοθήκη συστατικών UI για την κατασκευή web εφαρμογών με το Angular framework. Αναπτύχθηκε και διατηρείται από την ομάδα Angular στη Google, παρέχοντας ένα σύνολο προετοιμασμένων, καλά σχεδιασμένων και προσαρμόσιμων συστατικών UI που ακολουθούν τις οδηγίες του Material Design.

4.5 Github repository

Το GitHub repository (ή απλώς “repo”) αντιπροσωπεύει ένα αποθετήριο ηλεκτρονικών αρχείων που φιλοξενείται στην πλατφόρμα GitHub. Αυτό το αποθετήριο μπορεί να περιλαμβάνει κώδικα, έγγραφα, εικόνες, αρχεία δεδομένων και άλλα είδη πληροφοριών. Το GitHub αποτελεί ένα ευρέως διαδεδομένο εργαλείο συνεργασίας για προγραμματιστές και άλλους επαγγελματίες, παρέχοντας μια πλατφόρμα για τη διαχείριση και την κοινή χρήση κώδικα και έργων.

Ένα GitHub repository προσφέρει τα εξής χαρακτηριστικά:

- Διαχείριση Κώδικα: Το repo είναι ένας κεντρικός τόπος για την αποθήκευση, την παρακολούθηση και την επεξεργασία κώδικα. Οι προγραμματιστές μπορούν να κάνουν αναζήτηση στο ιστορικό αλλαγών, να δημιουργούν νέα υποκαταλόγους και να συγχωνεύουν τις αλλαγές.
- Συνεργασία: Πολλοί χρήστες μπορούν να συνεισφέρουν στο repo, κάνοντας αλλαγές, προσθέτοντας νέο κώδικα ή διορθώνοντας σφάλματα. Οι συνεισφορές γίνονται μέσω διαδικασιών όπως οι “pull requests.”

- Ιστορικό Αλλαγών: Κάθε αλλαγή στον κώδικα αποθηκεύεται, παρέχοντας μια λεπτομερή ανασκόπηση της εξέλιξης του έργου. Αυτό είναι χρήσιμο για την ανίχνευση σφαλμάτων, την παρακολούθηση προόδου και την επαναφορά σε προηγούμενες εκδόσεις.
- Σχολιασμός: Οι χρήστες μπορούν να σχολιάζουν τον κώδικα, τις συνεισφορές ή τα issues, βοηθώντας έτσι στην ανταλλαγή απόψεων και στην επίλυση προβλημάτων.
- Εκδόσεις: Το GitHub παρέχει ένα σύστημα για τη διαχείριση εκδόσεων του κώδικα, επιτρέποντας τη δημιουργία σταθερών και ασταθών εκδόσεων.
- Διαχείριση Προβλημάτων: Οι χρήστες μπορούν να δημιουργούν "issues" για τα προβλήματα που αντιμετωπίζουν, τις προτάσεις βελτίωσης και τις ιδέες, παρέχοντας ένα τρόπο για τη διαχείριση και την παρακολούθηση των ανοιχτών θεμάτων.

Τα GitHub repositories αποτελούν απαραίτητα εργαλεία για τη συνεργασία, τη διαμοιρασμό γνώσης και τη διαχείριση έργων σε ποικίλους τομείς, από τον ανοικτό κώδικα έως την επιχειρηματική ανάπτυξη και την ακαδημαϊκή έρευνα.

Ο κώδικας του παρόν project βρίσκεται στην σελίδα `https://github.com/Gemois/HCvision`. Επιπλέον, η εφαρμογή φιλοξενείται και είναι προσβάσιμη μέσω της διεύθυνσης `hcvision.iee.ihu.gr` για χρήση και δοκιμή της.

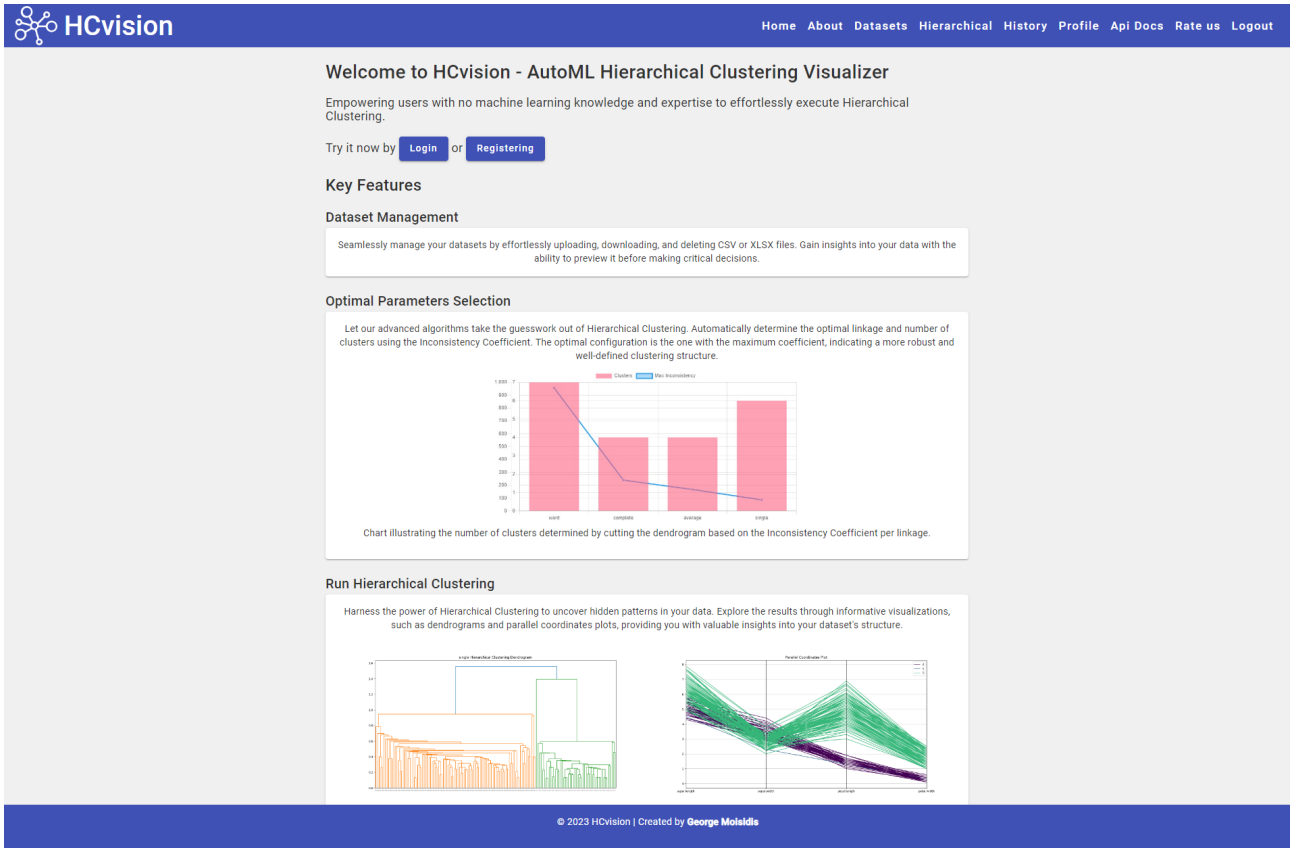
Το HCvision προσφέρει μια ευέλικτη λύση για εταιρείες ή οργανισμούς που ανησυχούν για την προστασία των δεδομένων τους. Ως εφαρμογή ανοιχτού κώδικα, το HCvision παρέχει τη δυνατότητα εγκατάστασης σε εσωτερικούς διακομιστές, εξασφαλίζοντας ότι τα ευαίσθητα δεδομένα παραμένουν ασφαλή στην υποδομή του οργανισμού. Η δυνατότητα αυτής της εσωτερικής εγκατάστασης επιτρέπει στις εταιρείες να χρησιμοποιούν τις ισχυρές λειτουργίες του HCvision για ανάλυση δεδομένων και συσταδοποίηση χωρίς την ανάγκη να εκθέτουν τα δεδομένα τους σε εξωτερικούς διακομιστές, συμμορφούμενες με αυστηρούς κανόνες απορρήτου και ασφάλειας.

Κεφάλαιο 5

Παρουσιάση του HCvision

5.1 Αρχική σελίδα

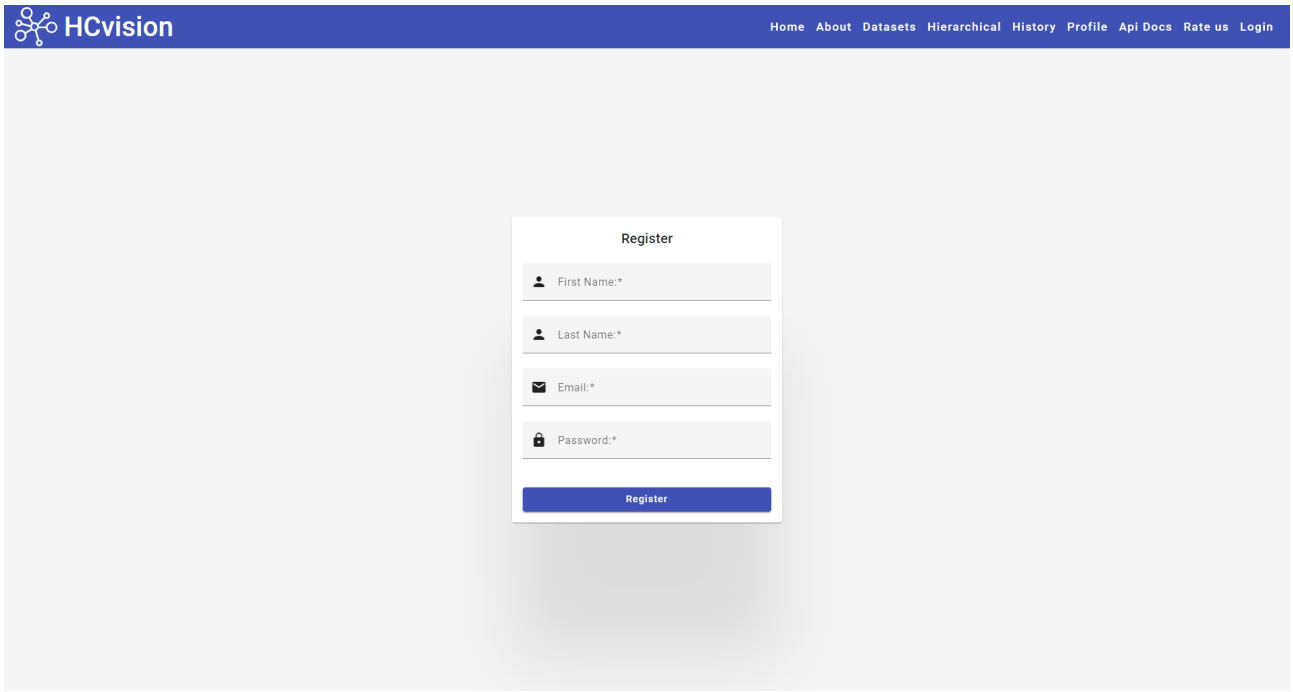
Στο Σχήμα 5.1 παρουσιάζεται η αρχική σελίδα της εφαρμογής. Εδώ, ο χρήστης μπορεί να βρει μια περιγραφή της εφαρμογής, η οποία περιλαμβάνει το όνομά της και μια σύντομη επεξήγηση των λειτουργιών που προσφέρει. Αυτές περιλαμβάνουν τη διαχείριση συνόλων δεδομένων, την αυτόματη εύρεση προτεινόμενων τιμών για μέθοδο σύνδεσης και πλήθους συστάδων, καθώς και την πλήρη ανάλυση ιεραρχικής συσταδοποίησης, συμπεριλαμβανομένων των παραγόμενων γραφημάτων και της ανάθεσης συστάδων. Η αρχική σελίδα παρουσιάζει επίσης στιγμιότυπα από τα αποτελέσματα που προσφέρει, βοηθώντας τον χρήστη να κατανοήσει το περιεχόμενο της εφαρμογής.



Σχήμα 5.1: Αρχική σελίδα

5.2 Εγγραφή νέου χρήστη

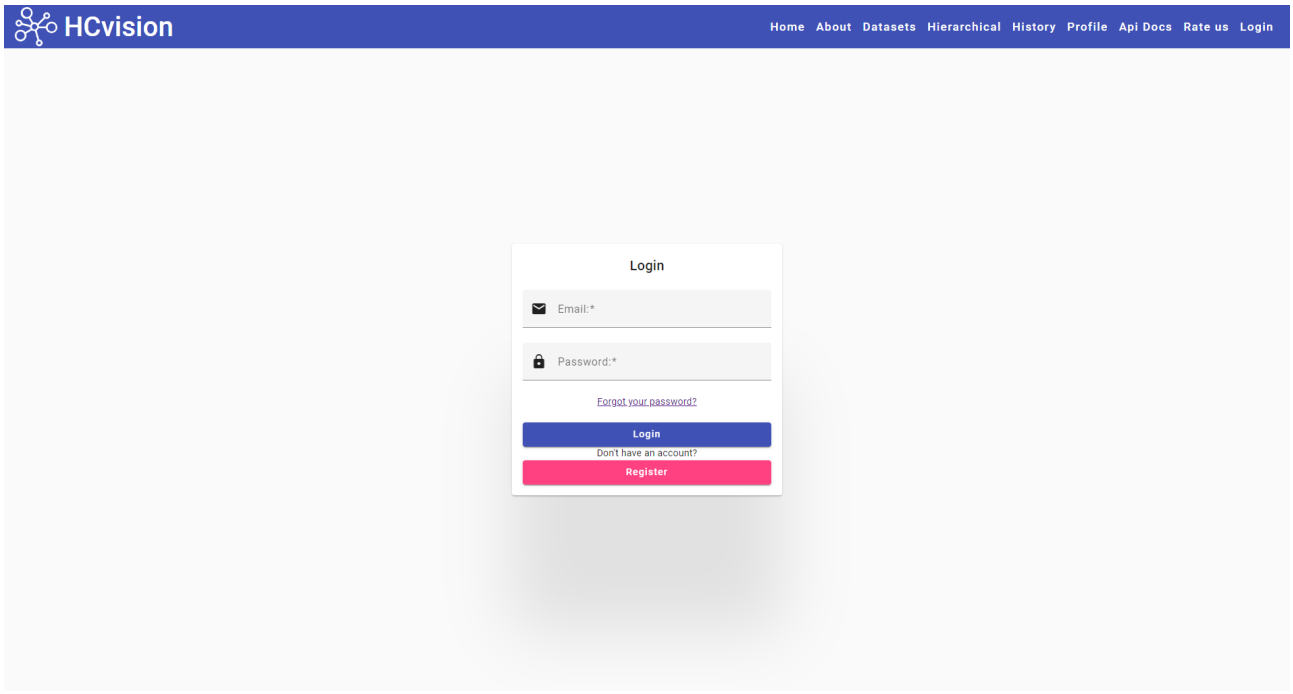
Στο σχήμα 5.2 φαίνεται η σελίδα εγγραφής χρήστη. Σε αυτήν τη σελίδα, ο χρήστης μπορεί να δημιουργήσει ένα νέο λογαριασμό στην εφαρμογή. Οι βασικές πληροφορίες που απαιτούνται για την εγγραφή περιλαμβάνουν το ονοματεπώνυμο του χρήστη, την ηλεκτρονική διεύθυνση και τον κωδικό πρόσβασης. Το email πρέπει να είναι μοναδικό, καθώς αποτελεί το αναγνωριστικό του χρήστη και με αυτό και τον κωδικό θα μπορεί να συνδεθεί στην εφαρμογή. Όλα τα πεδία πρέπει να είναι συμπληρωμένα για να είναι επιτυχής η διαδικασία εγγραφής.



Σχήμα 5.2: Δημιουργία λογαριασμού

5.3 Σύνδεση χρήστη στο σύστημα

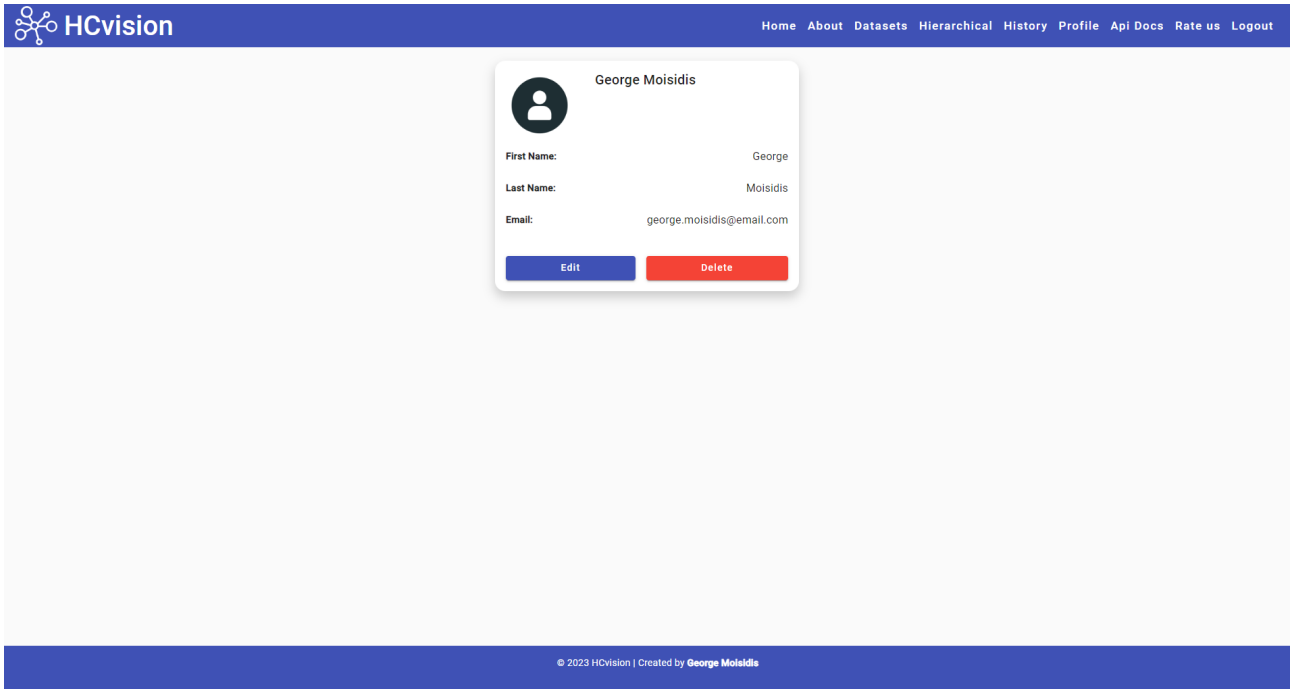
Στο σχήμα 5.3 παρουσιάζεται η σελίδα εισόδου. Οι υπάρχοντες χρήστες μπορούν να συνδεθούν στο σύστημα χρησιμοποιώντας την ηλεκτρονική τους διεύθυνση και τον κωδικό πρόσβασης. Σε αυτό το σημείο, οι χρήστες που δεν έχουν επιβεβαιώσει την ηλεκτρονική διεύθυνσή τους έχουν πρόσβαση μόνο στις εξής σελίδες: αρχική, σχετικά με εμάς, Api Docs, και Rate us. Σε όλες τις υπόλοιπες σελίδες εμφανίζεται ένα μήνυμα που παροτρύνει τον χρήστη να επιβεβαιώσει την ηλεκτρονική διεύθυνσή του. Τέλος, μετά από επιτυχή επιβεβαίωση της ηλεκτρονικής διεύθυνσης, όλες οι λειτουργίες είναι διαθέσιμες.



Σχήμα 5.3: Σύνδεση χρήστη στο σύστημα

5.4 Σελίδα Profile

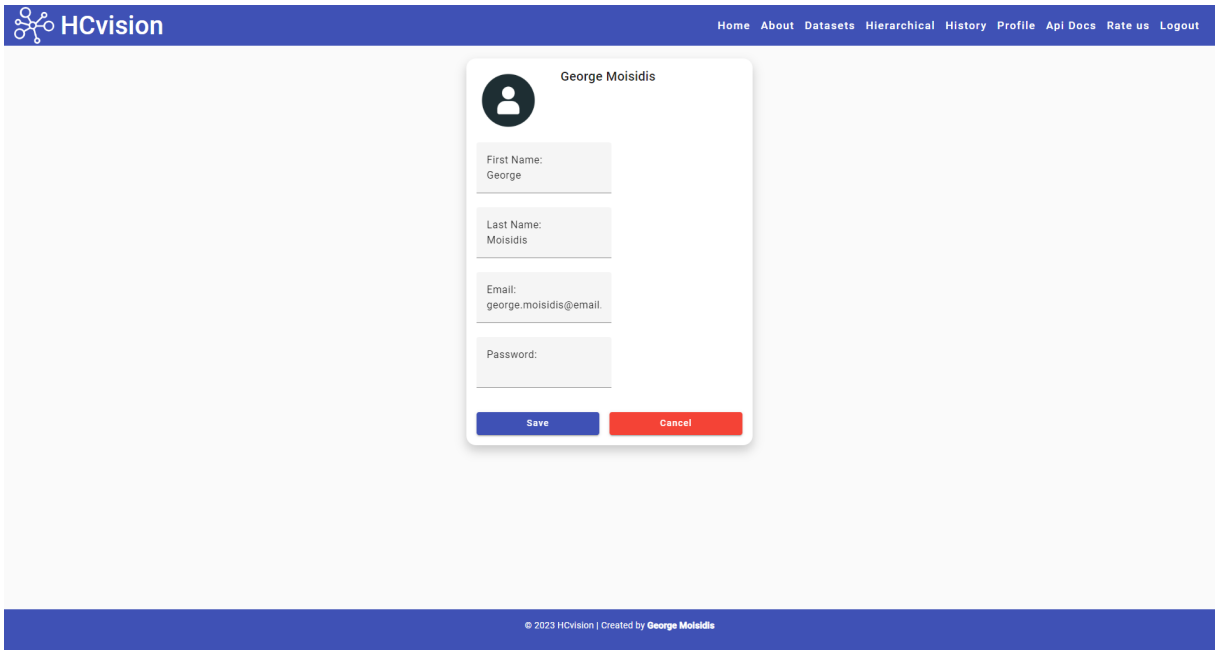
Στο σχήμα 5.4 παρουσιάζεται η σελίδα του χρήστη. Εδώ ο χρήστης βλέπει τα στοιχεία που έχει καταχωρίσει στο σύστημα, καθώς και τις επιλογές για ενέργειες επεξεργασίας και διαγραφής. Οι εν λόγω ενέργειες περιγράφονται παρακάτω.



Σχήμα 5.4: Επεξεργασία προσωπικών στοιχείων

5.4.1 Επεξεργασία προσωπικών στοιχείων και κωδικού πρόσβασης

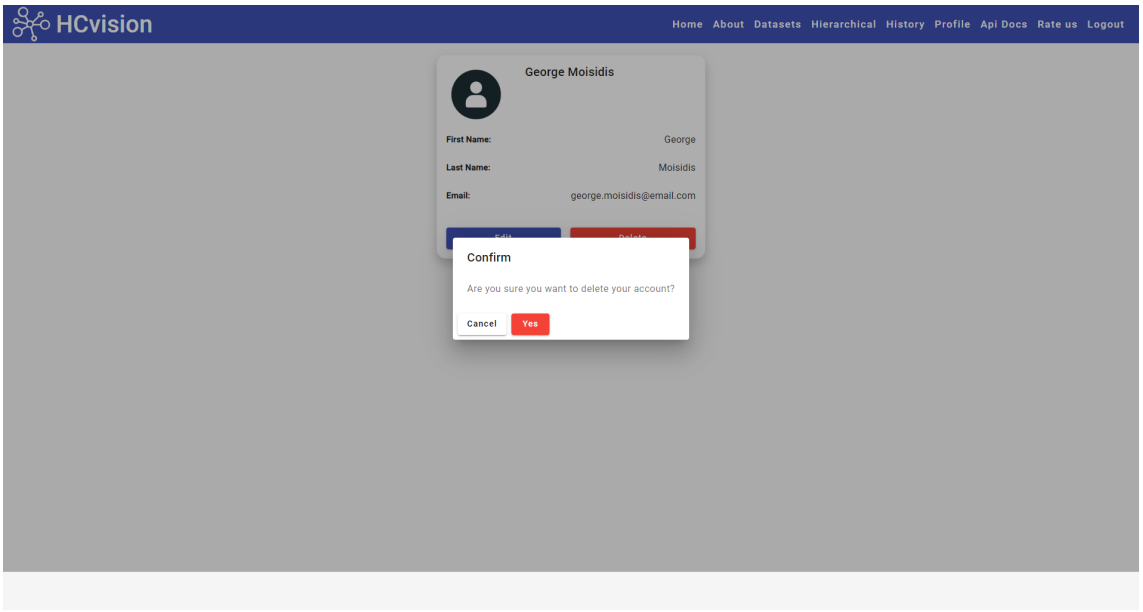
Στο σχήμα 5.5 φαίνεται η σελίδα του χρήστη. Εδω μπορεί να ενημερώσει ή να αλλάξει τα προσωπικά του στοιχεία, όπως το όνομα, την ηλεκτρονική διεύθυνση ή τον κωδικό πρόσβασης, για να διατηρήσει το προφίλ του ενημερωμένο.



Σχήμα 5.5: Αλλαγή στοιχείων χρήστη

5.4.2 Διαγραφή Λογαριασμού

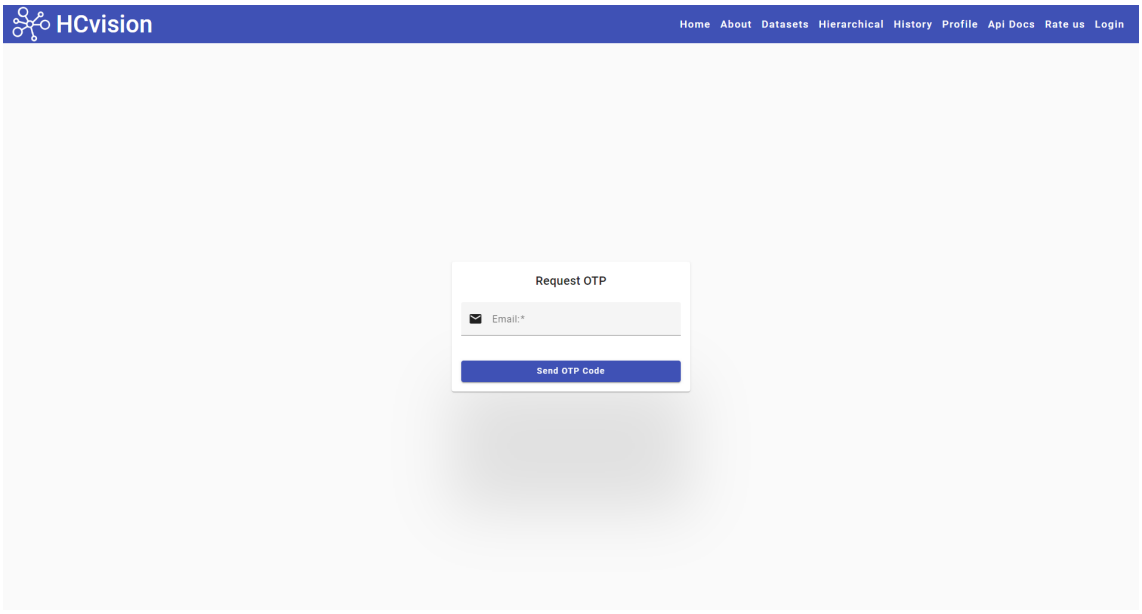
Στο σχήμα 5.6 φαίνεται η περίπτωση που ο χρήστης επιθυμεί να διαγράψει τον λογαριασμό του. Όλα τα στοιχεία του θα διαγραφούν μαζί με τον λογαριασμό του στην εφαρμογή.



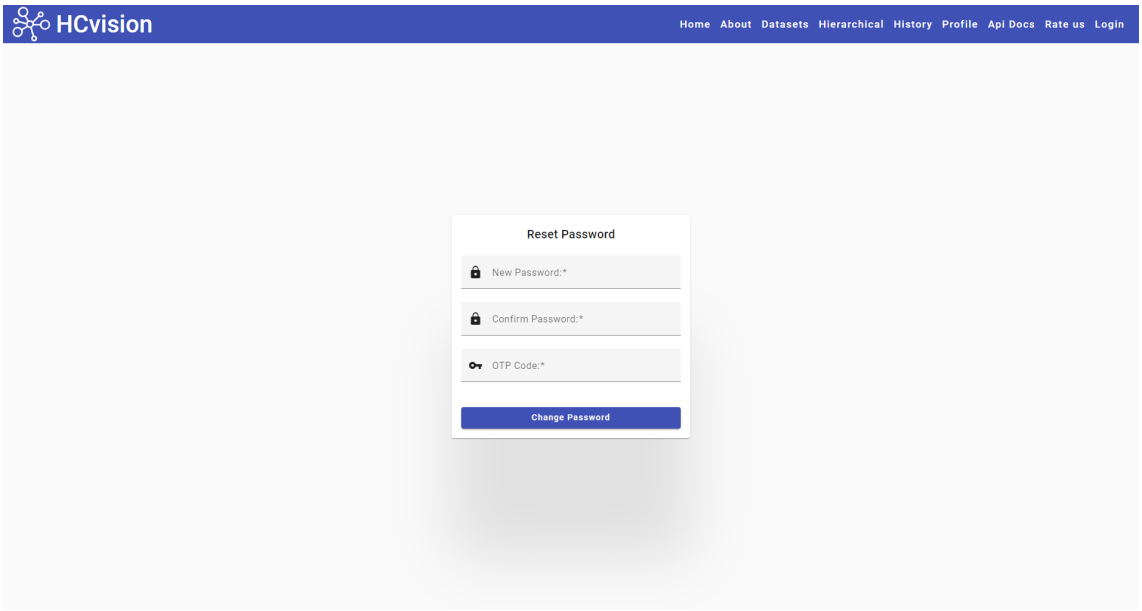
Σχήμα 5.6: Διαγραφή Λογαριασμού

5.5 Ανάκτηση κωδικού πρόσβασης

Στο σχήμα 5.7 παρουσιάζεται η πρώτη φόρμα που πρέπει να συμπληρώσει ένας χρήστης σε περίπτωση που ξεχάσει τον κωδικό του, η οποία είναι υπεύθυνη για να στείλει ένα μοναδικό κωδικό μιας χρήσης στο email του χρήστη. Στη συνέχεια, μια νέα φόρμα (βλ. Σχήμα 5.8) εμφανίζεται για την αλλαγή του κωδικού, στην οποία όμως πρέπει να υποβληθεί και ο μοναδικός κωδικός που απεστάλη στο email του χρήστη στο προηγούμενο βήμα.



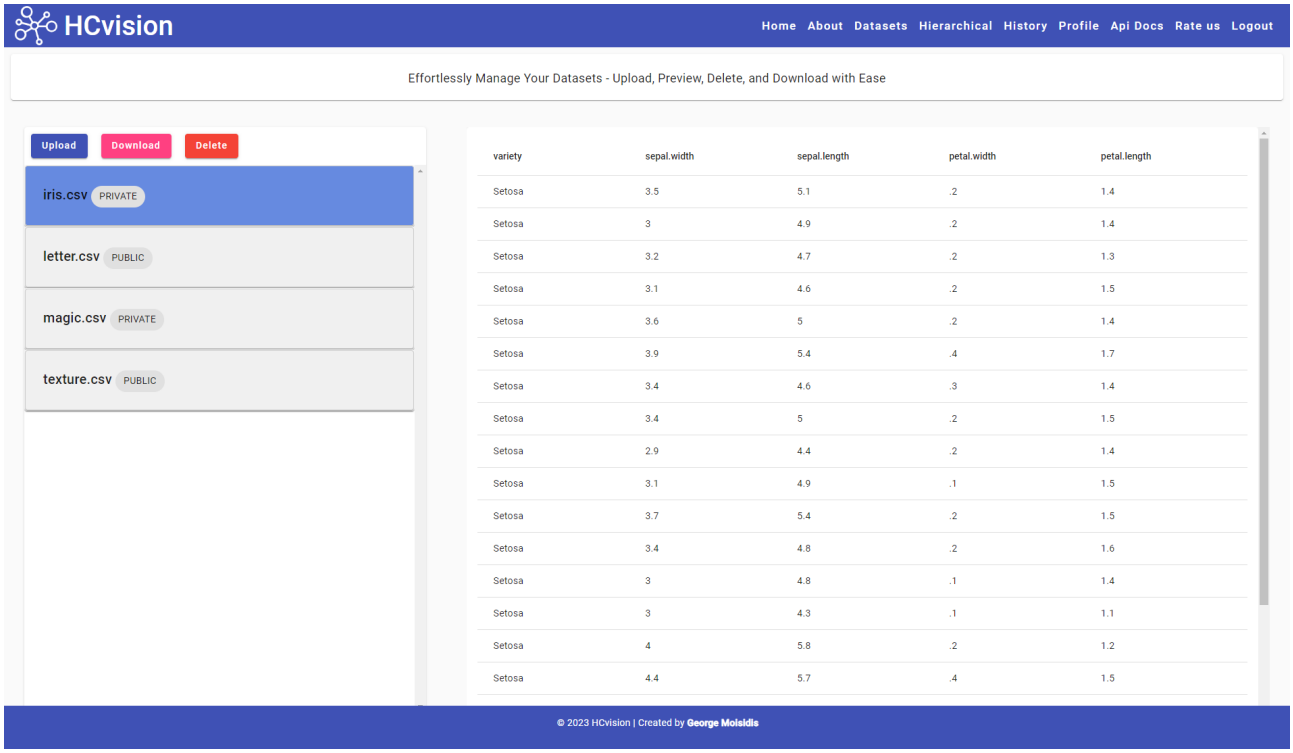
Σχήμα 5.7: Ανάκτηση κωδικού πρόσβασης



Σχήμα 5.8: Ανάκτηση κωδικού πρόσβασης

5.6 Σελίδα Datasets

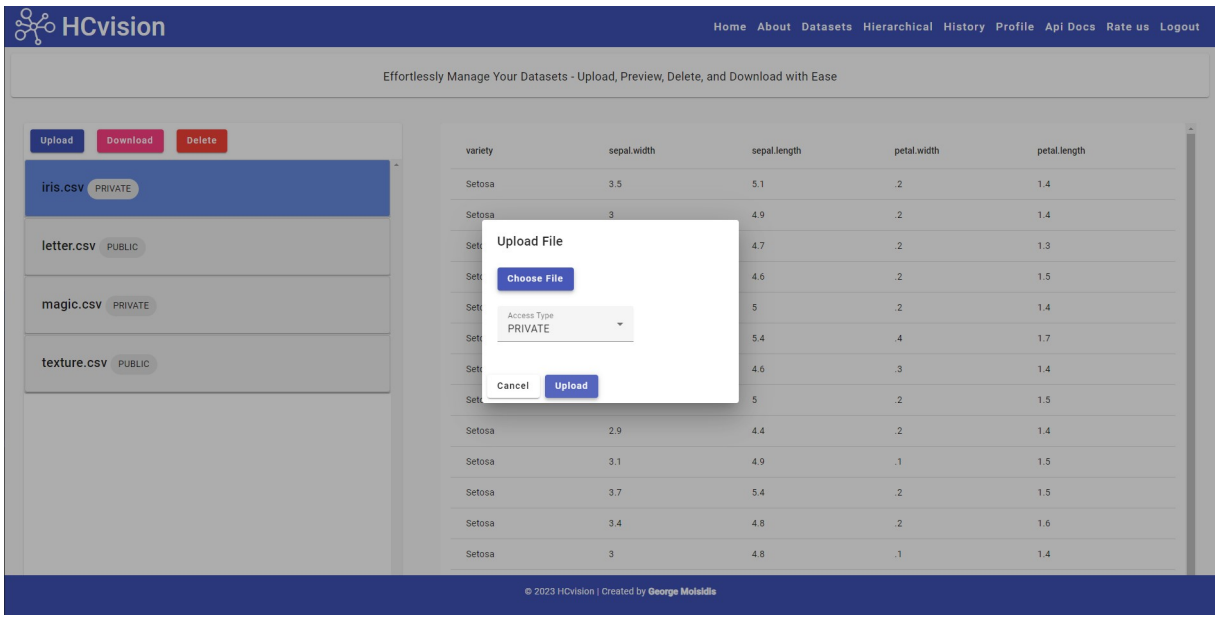
Στο σχήμα 5.9 φαίνεται η σελίδα Datasets. Εδώ ο χρήστης μπορεί να επεξεργαστεί τα σύνολα δεδομένων του, να ανεβάσει νέα, να τα διαγράψει και να δει το περιεχόμενό τους. Συγκεκριμένα, στο αριστερό κομμάτι της σελίδας εμφανίζεται μια λίστα με όλα τα διαθέσιμα σύνολα δεδομένων, ενώ στο δεξί κομμάτι εμφανίζεται η προεπισκόπηση των δεδομένων και υπάρχουν κουμπιά για ενέργειες επεξεργασίας, μεταφόρτωσης και διαγραφής. Οι ενέργειες αυτές παρουσιάζονται παρακάτω.



Σχήμα 5.9: Σελίδα Datasets

5.6.1 Ανέβασμα αρχείου

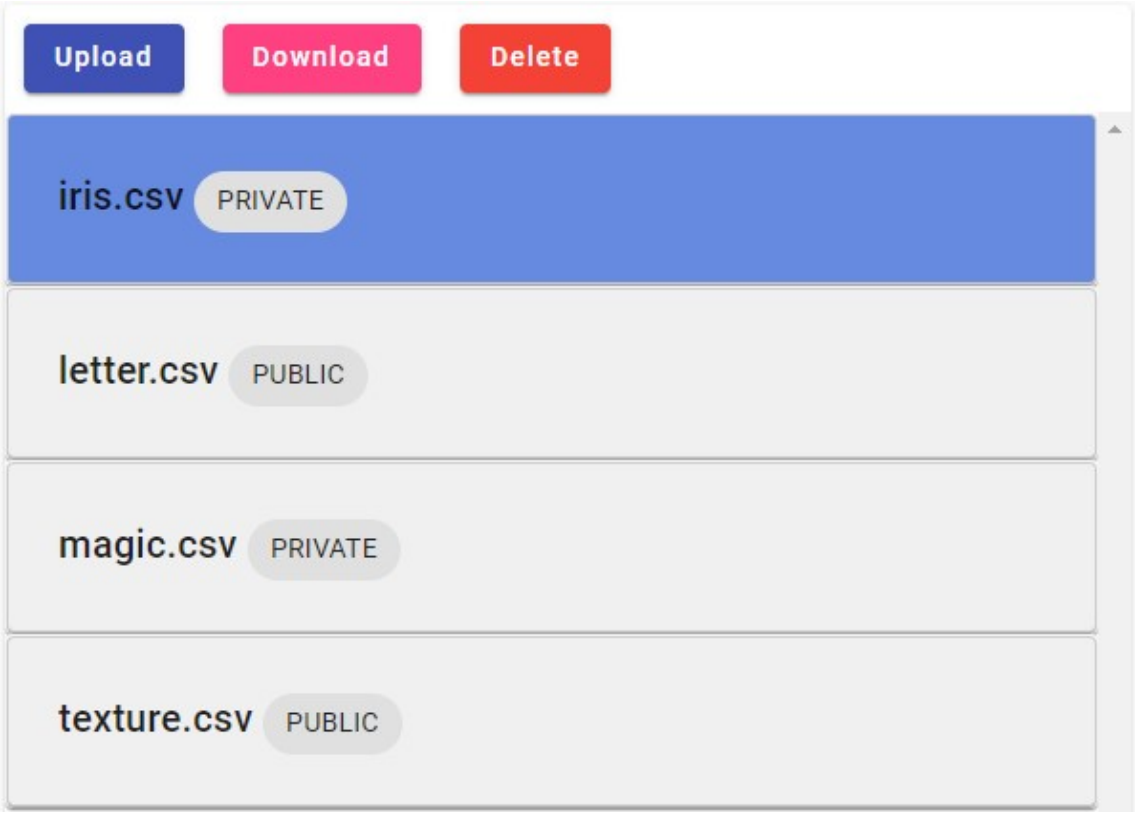
Ο χρήστης μπορεί να ανεβάσει αρχεία δεδομένων που επιθυμεί να αναλύσει μέσω της εφαρμογής. Στο σχήμα 5.10 φαίνεται η φόρμα που βλέπει ένας χρήστης. Πρέπει να επιλέξει το αρχείο που θέλει να ανεβάσει (CSV, XLSX) και στη συνέχεια να επιλέξει τον τύπο πρόσβασης (δημόσιο ή ιδιωτικό). Στη συνέχεια, το αρχείο ανεβαίνει στον διακομιστή και είναι διαθέσιμο για εφαρμογή των λειτουργιών HCvision.



Σχήμα 5.10: Ανέβασμα αρχείου συνόλου δεδομένων

5.6.2 Διαγραφή αρχείου

Εάν ένα αρχείο δεν είναι πλέον απαραίτητο, ο χρήστης μπορεί να το διαγράψει από το σύστημα επιλέγοντας το σύνολο δεδομένων και στην συνέχεια το κουμπί ‘delete‘ όπως φαίνεται στο σχήμα 5.11.



Σχήμα 5.11: Διαγραφή συνόλου δεδομένων

5.6.3 Ανάγνωση αρχείου

Στο σχήμα 5.12 φαίνεται η προεπισκόπηση ενός συνόλου δεδομένων. Εδώ ο χρήστης μπορεί να δει τα περιεχόμενα του συνόλου ώστε να επιλέξει στο επόμενο βήμα σωστά τις στήλες που περιέχουν μόνο αριθμούς.

variety	sepal.width	sepal.length	petal.width	petal.length
Setosa	3.5	5.1	.2	1.4
Setosa	3	4.9	.2	1.4
Setosa	3.2	4.7	.2	1.3
Setosa	3.1	4.6	.2	1.5
Setosa	3.6	5	.2	1.4
Setosa	3.9	5.4	.4	1.7
Setosa	3.4	4.6	.3	1.4
Setosa	3.4	5	.2	1.5
Setosa	2.9	4.4	.2	1.4
Setosa	3.1	4.9	.1	1.5
Setosa	3.7	5.4	.2	1.5
Setosa	3.4	4.8	.2	1.6
Setosa	3	4.8	.1	1.4
Setosa	3	4.3	.1	1.1
Setosa	4	5.8	.2	1.2
Setosa	4.4	5.7	.4	1.5
Setosa	3.9	5.4	.4	1.3
Setosa	3.5	5.1	.3	1.4
Setosa	3.8	5.7	.3	1.7
Setosa	3.8	5.1	.3	1.5

Items per page: 20 1 – 20 of 150

Σχήμα 5.12: Ανάγνωση συνόλου δεδομένων

5.7 Σελίδα Hierarchical

5.7.1 Optimal Parameters

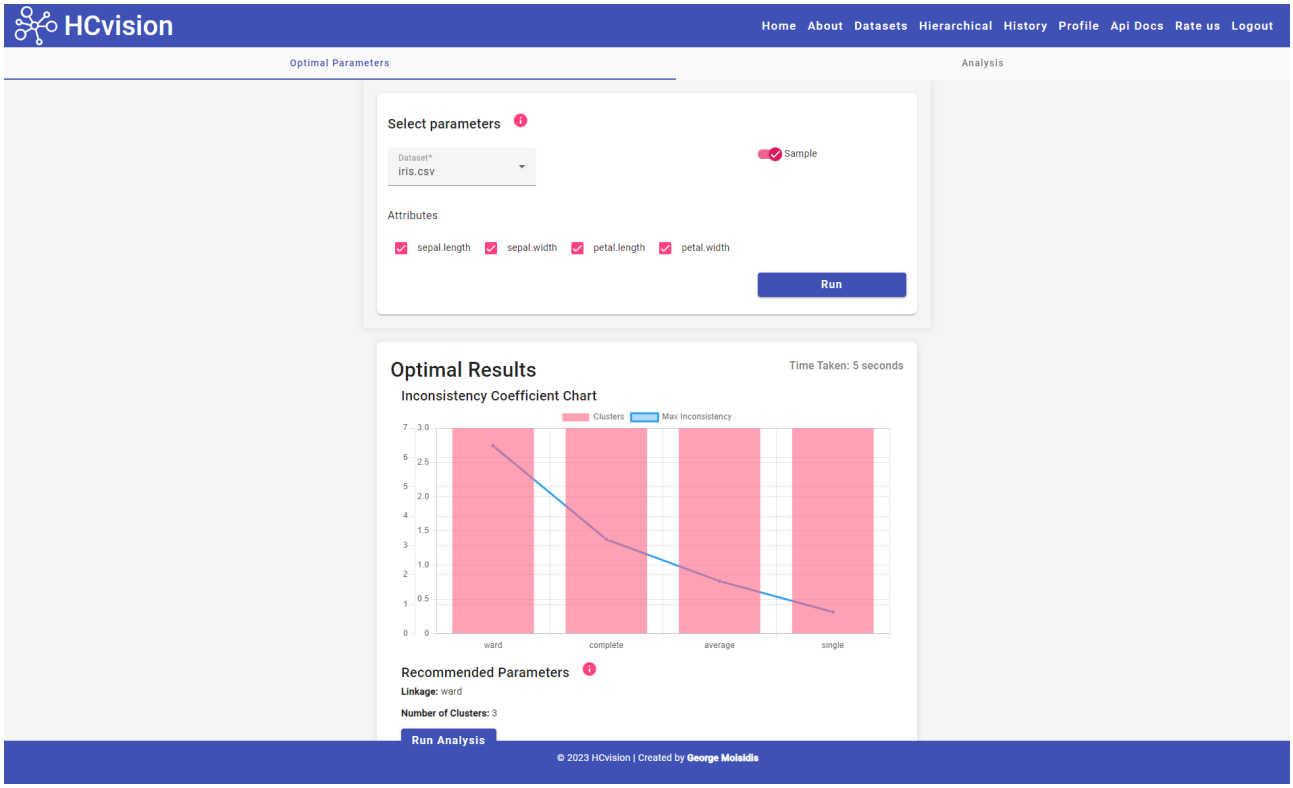
Στην Εικόνα 5.13, βλέπουμε τη φόρμα που πρέπει να συμπληρώσει ο χρήστης για να χρησιμοποιήσει τη μέθοδο για τον προσδιορισμό της μεθόδου σύνδεσης και του αριθμού των συστάδων. Παρατηρούμε ότι υπάρχει μια λίστα με τα χαρακτηριστικά του συνόλου δεδομένων, με προεπιλεγμένα από προεπιλογή μόνο εκείνα που περιέχουν αριθμητικά δεδομένα. Ο χρήστης μπορεί να επιλέξει ποιες στήλες θα συμμετέχουν στη διαδικασία. Μετά την εκτέλεση της διαδικασίας, εμφανίζεται ένα γράφημα που παρουσιάζει τα αποτελέσματα, τα οποία όμως παρατίθενται και παρακάτω. Τέλος, ο χρήστης έχει τη δυνατότητα να προχωρήσει στη σελίδα ανάλυσης πατώντας ένα κουμπί, όπου συμπληρώνονται αυτόματα όσα είχε επιλέξει, μαζί με τις προτεινόμενες τιμές, και εκτελείται η ιεραρχική ανάλυση.

Ερμηνεία του Γραφήματος

Το Python script υπολογίζει, για κάθε μέθοδο σύνδεσης, το ύψος του δενδρογράμματος όπου

παρατηρείται η μέγιστη ασυνέπεια, με τη βοήθεια του συντελεστή συνέπειας. Αυτή η διαδικασία καθορίζει επίσης τον αριθμό των συστάδων. Ήταν απαραίτητο να βρεθεί ένας τρόπος να παρουσιαστούν αυτά τα μετρήσιμα σε ένα γράφημα που θα μπορούσε να ερμηνευτεί εύκολα από έναν απλό χρήστη. Ειδικότερα, επιλέξαμε να απεικονίσουμε τον αριθμό των παραγόμενων συστάδων ανά μέθοδο σύνδεσης χρησιμοποιώντας ένα διάγραμμα ράβδων, όπως φαίνεται από τις τέσσερις ράβδους στο Σχήμα 5.13.

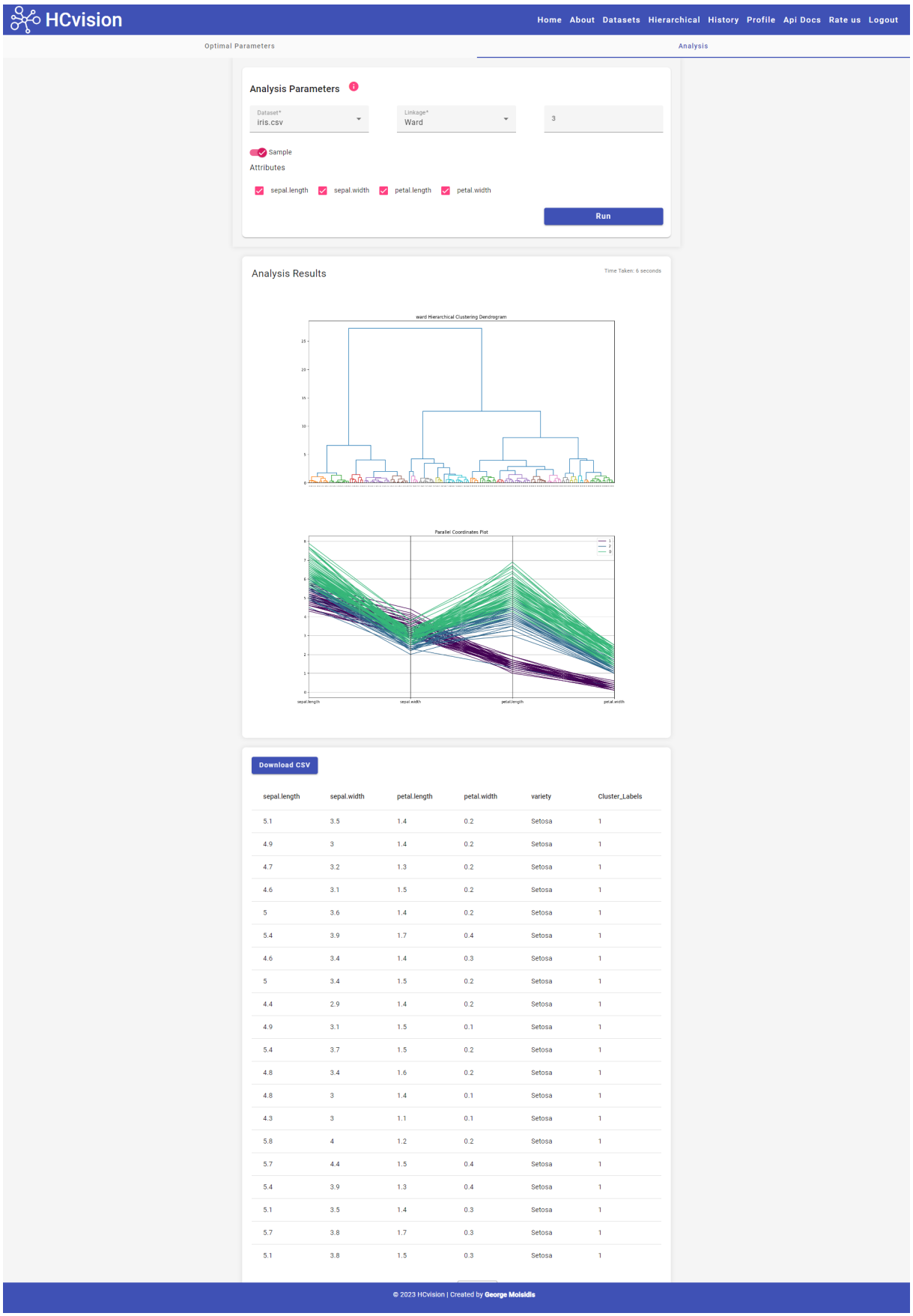
Για τον συντελεστή ασυνέπειας, επιλέξαμε μια γραφική αναπαράσταση με γραμμική σχέση, όπου ο άξονας x αναπαριστά τη μέθοδο σύνδεσης και ο άξονας y την τιμή του συντελεστή ασυνέπειας. Το γράφημα ερμηνεύεται εντοπίζοντας τη μεγαλύτερη μη μηδενική τιμή του συντελεστή ασυνέπειας που παράγει το μεγαλύτερο αριθμό συστάδων. Στο Σχήμα 5.13 παρατηρούμε ότι, όταν ο συντελεστής ασυνέπειας είναι 7, παράγει έναν αριθμό συστάδων ίσο με 3, και η μέθοδος που χρησιμοποιείται είναι η $Ward$.
Αυτή η παρουσίαση ενισχύει την κατανόηση της ανάλυσης, καθιστώντας την προσιτή σε ένα ευρύτερο κοινό.



Σχήμα 5.13: Σελίδα προσδιορισμού προτεινόμενων τιμών

5.7.2 Analysis

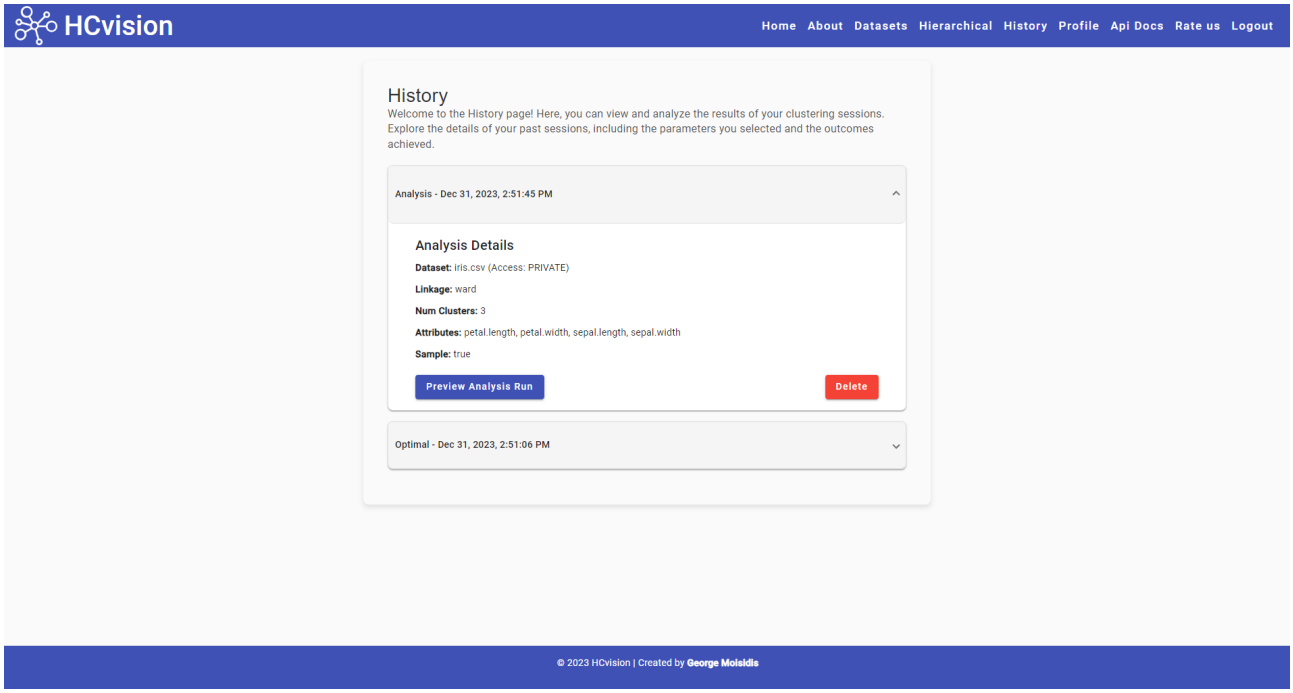
Εδώ, ο χρήστης μπορεί να εισάγει τις παραμέτρους που επιθυμεί και να εφαρμόσει τον αλγόριθμο ιεραρχικής συσσωρευτικής συρματοποίησης για τα δεδομένα του. Στο σχήμα 5.14 βλέπουμε την προεπισκόπηση ενος συρματοποιημένου συνόλου για τις τιμές $linkage = Ward$ και $clusters = 3$.



Σχήμα 5.14: Σελίδα ανάλυσης ιεραρχικής συρματοποίησης

5.8 Σελίδα History

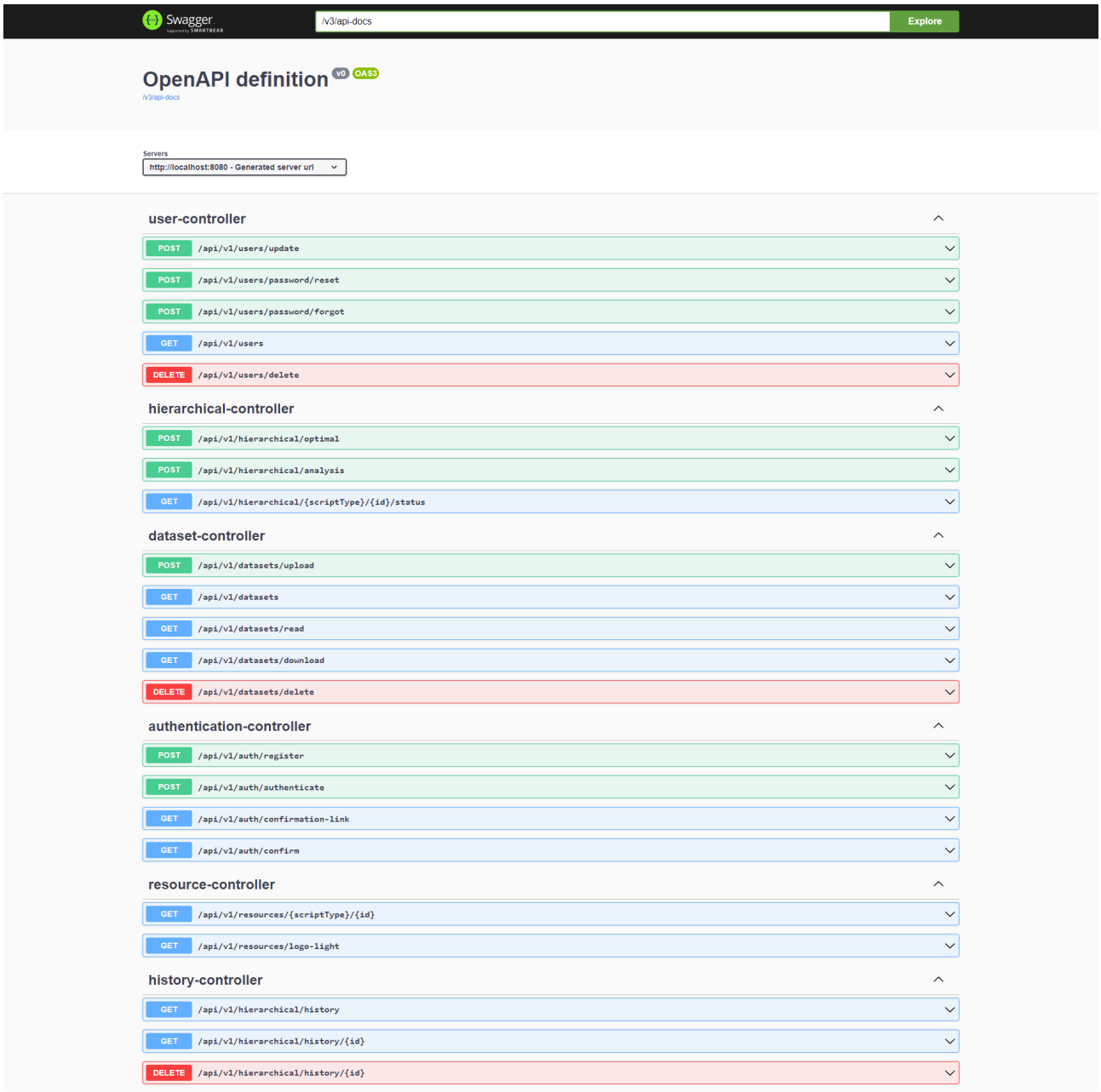
Εδώ, ο χρήστης μπορεί να επανεξετάσει αναλύσεις που έτρεξε στο παρελθόν. Στο σχήμα 5.15 βλέπουμε τη λίστα με προηγούμενες εκτελέσεις των δύο κύριων λειτουργιών του HCvision. Κάθε εκτέλεση καταγράφεται και εμφανίζεται στον χρήστη. Στη συνέχεια, ο χρήστης είναι σε θέση να δει λεπτομέρειες της λειτουργίας που εκτέλεσε, όπως τις παραμέτρους που επέλεξε. Επιπλέον, έχει τη δυνατότητα, με το πάτημα ενός κουμπιού, να μεταφερθεί στην αντίστοιχη σελίδα και να ξαναδεί τα αποτελέσματα. Υποστηρίζεται επίσης και η διαγραφή της ιστορικής καταγραφής.



Σχήμα 5.15: Σελίδα ιστορικού

5.9 Σελίδα API Docs

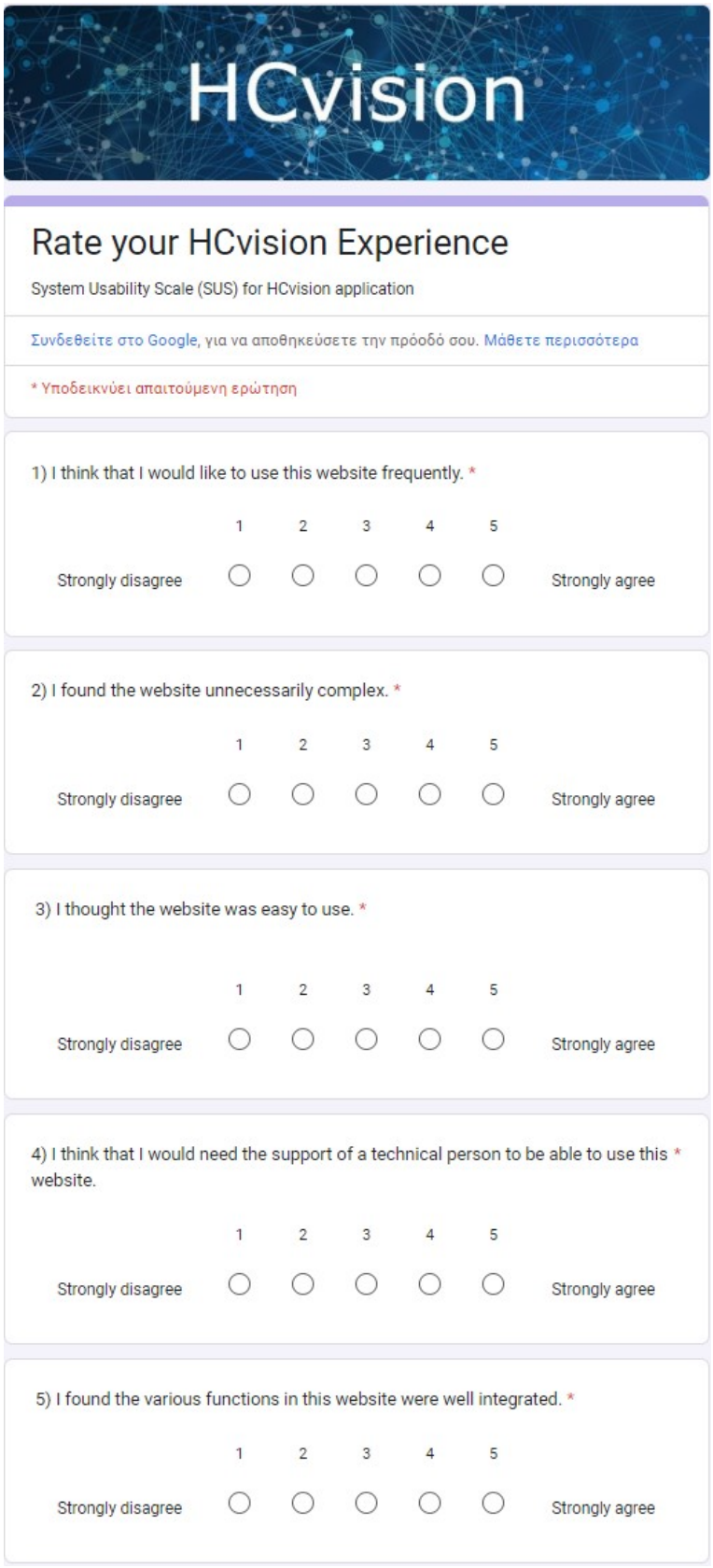
Στο σχήμα 5.16 φαίνεται η σελίδα με τη λίστα των διαθέσιμων endpoints. Επιλέγοντας ένα endpoint, επεκτείνονται οι πληροφορίες και μπορούμε να δούμε τις απαιτήσεις/προϋποθέσεις που απαιτούνται για να λειτουργήσει. Αυτή η σελίδα παράγεται με τη χρήση του Swagger UI, το οποίο αποτελεί ένα από τα πιο διαδεδομένα frameworks για την καταγραφή και δοκιμή των APIs. Αυτή η σελίδα είναι σημαντική για τους προγραμματιστές που θέλουν να χρησιμοποιήσουν το δημόσιο API του HCvision.



Σχήμα 5.16: Σελίδα προβολής API

5.10 Αξιολόγηση Εμπειρίας Χρήσης

Πατώντας το ”Rate Us” από τη γραμμή πλοήγησης της σελίδας, ο χρήστης μεταφέρεται σε ένα Google Form το οποίο περιέχει 10 ερωτήσεις σύμφωνα με το System Usability Scale (SUS) για την αξιολόγηση εμπειρίας χρήστη. Στο Σχήμα 5.17 παρουσιάζεται το ερωτηματολόγιο.



Σχήμα 5.17: Ερωτηματολόγιο Εμπειρίας Χρήστη

Κεφάλαιο 6

Αξιολόγηση του HCvision

6.1 Αξιολόγηση Απόδοσης Συστήματος

Στη φάση αξιολόγησης του συστήματος αυτής της εργασίας, πραγματοποιήσαμε μια ολοκληρωμένη ανάλυση των δύο Python scipts που χρησιμοποιούνται από την εφαρμογή HCvision. Για την αξιολόγηση της απόδοσης των σεναρίων χρησιμοποιήθηκαν εννέα διαφορετικά σύνολα δεδομένων, τα οποία διέφεραν ως προς το μέγεθος και τα χαρακτηριστικά τους. Το περιβάλλον εκτέλεσης που χρησιμοποιήθηκε για αυτά τα πειράματα περιλάμβανε ένα μηχάνημα εξοπλισμένο με επεξεργαστή 6 πυρήνων που λειτουργεί στα 3,6 GHz και 16 GB μνήμης RAM. Τα σύνολα δεδομένων υποβλήθηκαν σε δύο διαδοχικές διαδικασίες: τον προσδιορισμό των συνιστώμενων παραμέτρων για την ιεραρχική συσταδοποίηση και την επακόλουθη εφαρμογή του αλγορίθμου συσταδοποίησης με τη χρήση αυτών των συνιστώμενων παραμέτρων.

Τα αποτελέσματα της αξιολόγησης παρουσιάζονται στον πίνακα που ακολουθεί, παρουσιάζοντας βασικές μετρικές όπως ο χρόνος εκτέλεσης, η χρήση της CPU και της μνήμης, οι προτεινόμενες τιμές για το πλήθος των συστάδων, η προτεινόμενη μέθοδος σύνδεσης και οι αντίστοιχες μετρικές για τη φάση της συσταδοποίησης. Αξίζει να σημειωθεί ότι τα σύνολα δεδομένων παρουσίαζαν διαφορετικά χαρακτηριστικά, οδηγώντας σε διαφορετικές βέλτιστες παραμέτρους.

Συμπερασματικά, η αξιολόγηση των σεναρίων ιεραρχικής ομαδοποίησης παρέχει πολύτιμες πληροφορίες σχετικά με την απόδοσή τους σε ένα ευρύ σύνολο δεδομένων. Ειδικότερα, η παρατηρούμενη αύξηση του χρόνου εκτέλεσης καθώς αυξάνεται το μέγεθος του συνόλου δεδομένων αναδεικνύει την εγγενή πολυπλοκότητα της ιεραρχικής συσταδοποίησης, τονίζοντας την ευαισθησία του αλγορίθμου σε μεγαλύτερα σύνολα δεδομένων. Αυτή η συσχέτιση μεταξύ του μεγέθους του συνόλου δεδομένων και του χρόνου εκτέλεσης υπογραμμίζει τη σημασία της εξέτασης της επεκτασιμότητας των μεθόδων ιεραρχικής ομαδοποίησης, ιδίως όταν πρόκειται για εκτεταμένα σύνολα δεδομένων.

Επιπλέον, η επιρροή του αριθμού των χαρακτηριστικών στη διαδικασία ομαδοποίησης είναι εμφανής στα ευρήματά μας. Η διαφοροποίηση των προτεινόμενων παραμέτρων και των αντίστοιχων μετρικών εκτέλεσης σε σύνολα δεδομένων με διαφορετικά χαρακτηριστικά υπογραμμίζει την επίδραση των διαστάσεων των χαρακτηριστικών στο αποτέλεσμα της ιεραρχικής συσταδοποίησης. Η παρατήρηση αυτή υπογραμμίζει την ανάγκη για μια διαφοροποιημένη προσέγγιση στην επιλογή και βελτιστοποίηση των παραμέτρων με βάση τόσο το μέγεθος του συνόλου δεδομένων όσο και τα χαρακτηριστικά.

Παρά τις προκλήσεις που θέτει η εγγενής πολυπλοκότητα της ιεραρχικής ομαδοποίησης, είναι αξιοσημείωτο ότι καθ' όλη τη διάρκεια του παρόντος έργου, οι χρόνοι εκτέλεσης και οι μετρικές χρήσης των πόρων παρέμειναν σταθερά εντός αποδεκτών ορίων. Τα scipts επέδειξαν ένα αξιοσημείωτο επίπεδο αποτελεσματικότητας και αξιοπιστίας, διαχειριζόμενα με επιτυχία σύνολα δεδομένων διαφορετικών μεγεθών και χαρακτηριστικών. Παρά την παρατηρούμενη αύξηση του χρόνου εκτέλεσης με μεγαλύτερα σύνολα δεδομένων, οι αλγόριθμοι λειτούργησαν ομαλά, ικανοποιώντας τις προσδοκίες απόδοσης του έργου. Αυτή η σταθερότητα είναι ζωτικής σημασίας για τις πρακτικές εφαρμογές, διασφαλίζοντας ότι το HCvision μπορεί να αναπτυχθεί αποτελεσματικά σε σενάρια όπου η ιεραρχική συσταδοποίηση αποτελεί βιώσιμη λύση.

Η συνολική απόδοση του HCvision, όπως αποδεικνύεται από τις αξιολογήσεις που πραγματοποιήθηκαν, υπογραμμίζει την πρακτικότητα και την ευρωστία του για ένα εύρος εργασιών ομαδοποίησης, συμβάλλοντας στις δυνατότητές του για πραγματικές εφαρμογές. Στον πίνακα 6.1 παρουσιάζονται τα αποτελέσματα.

Dataset	Size (kB)	Rows	Attributes	CPU Time$_{(Optimal)}$	Clusters	Linkage	CPU Time$_{(Analysis)}$
iris	3.88	151	4	0.01	3	Ward	0.85
telco_2023	45.5	1001	19	0.06	4	Complete	3.28
train	119	2001	5	0.29	3	Ward	6.32
penbased	537	10993	16	13.51	10	Complete	42.25
letter	715	20001	16	43.60	4	Complete	84.03
magic	1420	19021	10	35.93	5	Single	56.90
texture	1460	5501	40	2.37	3	Ward	18.39
air_traffic	5380	34735	2	144.32	4	Ward	135.85

Πίνακας 6.1: Μετρήσεις απόδοσης συστήματος

6.2 Αξιολόγηση της Εμπειρίας Χρήστη

Η κλίμακα ευχρηστίας συστήματος (SUS) είναι ένα αποτελεσματικό εργαλείο για την αξιολόγηση της ευχρηστίας ενός συστήματος, προϊόντος ή υπηρεσίας. Παρουσιάστηκε το 1986 από τον John Brooke και έκτοτε έχει γίνει ένα από τα πιο δημοφιλή και ευρέως αποδεκτά εργαλεία για την αξιολόγηση της ευχρηστίας[52].

Αποτελούμενο από 10 ερωτήσεις με πέντε επιλογές απάντησης, το SUS καλύπτει διάφορες πτυχές της ευχρηστίας, συμπεριλαμβανομένης της ικανοποίησης του χρήστη, της ευκολίας χρήσης και της λειτουργικότητας του συστήματος. Οι χρήστες αξιολογούν την εμπειρία τους με το σύστημα σε μια κλίμακα από το 1 (Διαφωνώ απόλυτα) έως το 5 (Συμφωνώ απόλυτα).

Ακολουθούν οι ερωτήσεις του SUS ερωτηματολογίου:

1. Νομίζω ότι θα ήθελα να χρησιμοποιώ συχνά αυτό το σύστημα.
2. Βρήκα το σύστημα περιττά πολύπλοκο.
3. Νομίζω ότι το σύστημα ήταν εύκολο στη χρήση.

4. Νομίζω ότι θα χρειαζόμουν την υποστήριξη ενός τεχνικού για να μπορέσω να χρησιμοποιήσω αυτό το σύστημα.
5. Βρήκα ότι οι διάφορες λειτουργίες αυτού του συστήματος ήταν καλά ενσωματωμένες.
6. Νομίζω ότι υπήρχε μεγάλη ασυνέπεια σε αυτό το σύστημα.
7. Φαντάζομαι ότι οι περισσότεροι άνθρωποι θα μάθαιναν να χρησιμοποιούν αυτό το σύστημα πολύ γρήγορα.
8. Βρήκα το σύστημα πολύ δυσκίνητο στη χρήση.
9. Αισθάνθηκα πολύ σίγουρος για τη χρήση του συστήματος.
10. Χρειάστηκε να μάθω πολλά πράγματα για να μπορέσω να ξεκινήσω με αυτό το σύστημα.

Ο υπολογισμός της βαθμολογίας SUS υπολογίζεται ως εξής, για τις ερωτήσεις με μονό αριθμό (1, 3, 5, 7, 9), το 1 αφαιρείται από την απάντηση του χρήστη, ενώ για τις ερωτήσεις με ζυγό αριθμό (2, 4, 6, 8, 10), η απάντηση του χρήστη αφαιρείται από το 5. Το άθροισμα των αποτελεσμάτων πολλαπλασιάζεται με το 2,5 για να προκύψει μια βαθμολογία που κυμαίνεται από 0 έως 100. Στο πίνακα 6.2 βλέπουμε τα αποτελέσματα ενός δείγματος 20 ερωτηματολογίων για την εφαρμογή HCvision.

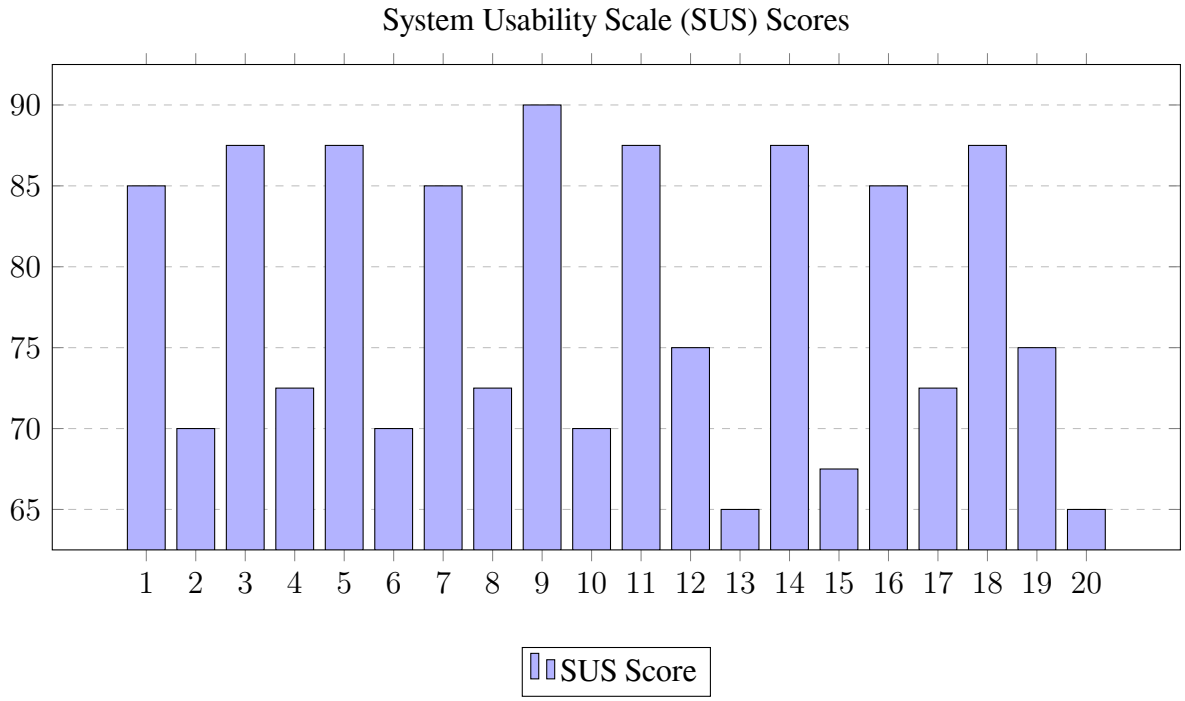
Χρονική σήμανση	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	SUM
19/01/2023	5	4	5	3	5	2	5	3	5	2	85.0
20/01/2023	4	3	5	3	4	2	4	3	5	2	70.0
20/01/2023	5	3	4	2	5	3	5	3	4	3	87.5
21/01/2023	4	4	5	2	4	3	5	2	5	2	72.5
21/01/2023	5	3	4	3	5	2	4	3	4	3	87.5
22/01/2023	4	3	5	2	4	3	4	3	5	2	70.0
22/01/2023	5	4	4	3	5	2	5	3	4	2	85.0
23/01/2023	4	4	5	2	4	3	5	2	5	2	72.5
23/01/2023	5	3	5	2	5	3	4	3	5	3	90.0
24/01/2023	4	3	4	3	4	2	5	3	4	2	70.0
24/01/2023	5	4	5	2	5	2	4	3	5	2	87.5
25/01/2023	4	3	5	3	4	3	4	3	5	3	75.0
25/01/2023	4	4	4	2	4	2	4	2	4	2	65.0
26/01/2023	5	3	5	2	5	3	4	3	5	3	87.5
26/01/2023	4	4	4	2	4	2	5	2	4	2	67.5
27/01/2023	5	4	5	2	5	2	5	3	5	2	85.0
27/01/2023	4	3	4	3	4	3	4	3	4	3	72.5
28/01/2023	5	4	5	2	5	2	5	3	5	2	87.5
28/01/2023	4	3	5	3	4	3	4	3	5	3	75.0
29/01/2023	4	4	4	2	4	2	4	2	4	2	65.0

Πίνακας 6.2: Αποτελέσματα ερωτηματολογίου SUS

Αν και η βαθμολογία κυμαίνεται από 0 έως 100, δεν πρέπει να ερμηνεύεται ως ποσοστό. Γενικά, μια βαθμολογία SUS άνω του 68 θεωρείται άνω του μέσου όρου, υποδηλώνοντας καλή χρηστικότητα, ενώ μια βαθμολογία κάτω του 68 υποδηλώνει περιθώρια βελτίωσης. Το SUS

παρέχει μια γρήγορη και αποτελεσματική μέθοδο για την κατανόηση της ευχρηστίας του συστήματος και τον εντοπισμό περιοχών για βελτίωση.

Βλέπουμε ότι κατά μέσο όρο η εφαρμογή αξιολογείται με περίπου 85.8 για την εμπειρία χρήσης. Επισης μια γραφική αναπαράσταση του Πίνακα 6.2 φαίνεται στο Γράφημα 6.1.



Σχήμα 6.1: Αποτελέσματα των ερωτηματολογίων SUS

Κεφάλαιο 7

Συμπεράσματα και Μελλοντικές Επεκτάσεις

7.1 Συμπεράσματα

Συμπερασματικά, η εργασία αυτή αντιμετωπίζει ένα σημαντικό κενό στο τρέχον τοπίο των εργαλείων αυτοματοποιημένης μηχανικής μάθησης, επικεντρώνοντας στον τομέα της ιεραρχικής ανάλυσης συστάδων. Ενώ τα εργαλεία αυτοματοποιημένης μηχανικής μάθησης έχουν κερδίσει ευρεία δημοτικότητα, ιδιαίτερα στο πλαίσιο των αλγορίθμων ταξινόμησης, υπάρχει ένα εμφανές κενό σε εργαλεία που επικεντρώνονται στην αυτοματοποίηση των πολύπλοκων διαδικασιών που σχετίζονται με τη συσταδική ομαδοποίηση. Η απουσία τέτοιων εργαλείων είναι ιδιαίτερα εμφανής στον τομέα της ιεραρχικής ομαδοποίησης, όπου η καθοριστική επιλογή της βέλτιστης σύνδεσης και του αριθμού των ομάδων παραμένει δύσκολη λόγω της ενσωματωμένης πολυπλοκότητας του αλγορίθμου.

Η έρευνά μας έχει αναδείξει αυτό το κρίσιμο κενό, επισημαίνοντας την ανάγκη για αυτοματοποιημένες λύσεις που σχεδιάζονται ειδικά για την ιεραρχική ομαδοποίηση. Ανταποκρινόμενοι σε αυτήν την ανεπάρκεια, αναπτύξαμε μια καινοτόμο εφαρμογή AutoML που όχι μόνο αυτοματοποιεί την επιλογή της πιο κατάλληλης σύνδεσης για ένα συγκεκριμένο dataset, αλλά βοηθά επίσης τους χρήστες να καθορίσουν τον βέλτιστο αριθμό των ομάδων. Επίσης το HCvision απευθύνεται σε χρήστες που έχουν δεδομένα και επιθυμούν να τα αναλύσουν, χωρίς να απαιτείται η κατοχή των απαραίτητων ειδικών γνώσεων στον τομέα της μηχανικής μάθησης και της εξόρυξης δεδομένων. Συνήθως, οι χρήστες αυτοί δεν διαθέτουν τις απαραίτητες δεξιότητες προγραμματισμού ή την εμπειρία στη χρήση βιβλιοθηκών και ειδικού λογισμικού που απαιτεί πολύπλοκη εγκατάσταση και παραμετροποίηση, και πολλές φορές αυτό το λογισμικό δεν είναι διαθέσιμο δωρεάν. Αυτό το εργαλείο διευκολύνει μια ομαλή και χρηστική εμπειρία, επιτρέποντας στους ερευνητές και τους αναλυτές δεδομένων να μεταφορτώνουν τα datasets τους εύκολα και να αποκτούν πολύτιμες πληροφορίες από τα αποτελέσματα της ιεραρχικής ομαδοποίησης.

Παρέχοντας μια ολοκληρωμένη και αυτοματοποιημένη λύση, το εργαλείο AutoML μας γεφυρώνει το χάσμα μεταξύ των χρηστών και των πολύπλοκων διαδικασιών της ανάλυσης ιεραρχικής ομαδοποίησης. Πιστεύουμε ότι η συμβολή μας θα εξουσιάσει τους ερευνητές, τους επιστήμονες δεδομένων και τους επαγγελματίες να εξερευνήσουν περαιτέρω τα δεδομένα τους χωρίς το βάρος των πολυπλοκοτήτων της αλγοριθμικής λήψης αποφάσεων. Η φιλική προς τον χρήστη διεπαφή εξασφαλίζει την προσβασιμότητα, καθιστώντας προσιτές για ένα ευρύτερο

κοινό τεχνικές μηχανικής μάθησης προηγμένες.

Ουσιαστικά, το εργαλείο AutoML μας όχι μόνο καλύπτει ένα κενό στο τρέχον τοπίο των εργαλείων ομαδοποίησης, αλλά αντιπροσωπεύει ένα βήμα προς τη δημοκρατοποίηση της ανάλυσης δεδομένων και της μηχανικής μάθησης. Καθώς προχωρούμε σε μια εποχή αυξανόμενης εξάρτησης από τη λήψη αποφάσεων με βάση τα δεδομένα, η εφαρμογή μας αντιπροσωπεύει μια απόδειξη της σημασίας της ανάπτυξης εξειδικευμένων εργαλείων που ανταποκρίνονται στις μοναδικές προκλήσεις που παρουσιάζει η ιεραρχική ομαδοποίηση. Μέσω αυτής της διατριβής, ελπίζουμε να ενθαρρύνουμε περαιτέρω προόδους στα εργαλεία AutoML, προωθώντας μια πιο περιεκτική και αποτελεσματική προσέγγιση για την ανάκτηση σημαντικών εισηγμένων αποτελεσμάτων από πολύπλοκα σύνολα δεδομένων.

7.2 Μελλοντικές Επεκτάσεις

Πρόταση για επιλογή σχετικών χαρακτηριστικών
Εισαγωγή χαρακτηριστικού που προτείνει τα πιο σχετικά χαρακτηριστικά για ανάλυση ομαδοποίησης. Παρόλο που η ομαδοποίηση είναι μη εποπτική, η χρήση τεχνικών μείωσης διαστάσεων όπως ο Ανάλυση Κύριων Συνιστωσών (PCA) μπορεί να παράσχει εισαγωγές σχετικά με τα χαρακτηριστικά που συνεισφέρουν σημαντικά στη διακύμανση των δεδομένων.

Υποστήριξη για διαιρετική ιεραρχική συσταδοποίηση
Προτείνεται η επέκταση της εφαρμογής AutoML για να υποστηρίζει τη διαιρετική ιεραρχική ομαδοποίηση, προσφέροντας στους χρήστες τη δυνατότητα να επιλέγουν ανάμεσα σε συσσωρευτική και διαιρετική προσέγγιση. Η διαιρετική ομαδοποίηση εμπλέκει τον διαχωρισμό ομάδων σε μικρότερες υποομάδες, παρέχοντας μια εναλλακτική οπτική γωνία στις ιεραρχικές δομές εντός των δεδομένων.

Διάγραμμα βηματικής οπτικοποίησης του δενδρογράμματος
Επιδίωξη να ενισχυθεί η κατανόηση των χρηστών για την ιεραρχική ομαδοποίηση μέσω της ενσωμάτωσης ενός κινούμενου διαγράμματος βηματικής οπτικοποίησης του δενδρογράμματος. Αυτή η λειτουργία θα επιδεικνύει δυναμικά πώς συγχωνεύονται ομάδες σε κάθε επανάληψη, προσφέροντας έτσι μια πιο ευαίσθητη αντίληψη της προόδου του αλγορίθμου.

Ανάπτυξή διεπαφής για κινητές συσκευές
Μια μελλοντική επέκταση που θα μπορούσε να προστεθεί είναι η ανάπτυξη μιας εφαρμογής για κινητές συσκευές που θα προσφέρει τις λειτουργίες του HCvision. Αυτή η επέκταση θα επιτρέψει στους χρήστες να αξιοποιούν τις δυνατότητες της εφαρμογής ανεξάρτητα από τη συσκευή τους και θα επιτρέπει ευέλικτη πρόσβαση σε δεδομένα και αναλύσεις. Επιπλέον, αναφέρεται ότι μπορεί να χρησιμοποιηθεί το Flutter για την ανάπτυξη της εφαρμογής, προσφέροντας έτσι υποστήριξη και συμβατότητα με διάφορες συσκευές και λειτουργικά συστήματα.

Βιβλιογραφία

[1] L. Rokach and O. Maimon, *Clustering Methods*, pp. 321–352. Boston, MA: Springer US, 2005.

[2] A. Dharmarajan and T. Velmurugan, "Applications of partition based clustering algorithms: A survey," in *2013 IEEE International Conference on Computational Intelligence and Computing Research*, pp. 1–5, 2013.

[3] J.-C. Chouinard, "What is kmeans clustering algorithm (with example) – python." `https://www.jcchouinard.com/kmeans/`. Accessed: 2024-01-21.

[4] D. Xu and Y. Tian, "A comprehensive survey of clustering algorithms," *Annals of Data Science*, vol. 2, pp. 165–193, 06 2015.

[5] P. Pai, "Hierarchical clustering explained," 05 2021.

[6] V. Verykios, V. Kagklis, and E. Stavropoulos, "Data science through the r language," *Kallipos.gr*, 2021.

[7] "Fundamental parameter estimation i - merlin bartel." `https://merlinbartel.com/2023/04/25/fundamental-parameter-estimation-i/`, 04 2023. Accessed: 2024-01-21.

[8] G. Lee, D.-I. Kim, S. Kim, and y.-j. Shin, "Multiscale pmu data compression via density-based wams clustering analysis," *Energies*, vol. 12, p. 617, 02 2019.

[9] N. Gürsakal, H. O. Çobanoğlu, B. Batmaz, S. Cagliyor, and F. Yilmaz, "Modularity in football passing networks," *Turkish Journal of Sport and Exercise*, vol. 22, pp. 296–304, 09 2020.

[10] waikato, "Weka 3 - data mining with open source machine learning software in java," 2019. Accessed: 2024-01-21.

[11] "Data analytics and ai platform | altair rapidminer." Accessed: 2024-01-21.

[12] scikit learn, "scikit-learn: machine learning in python," 2019. Accessed: 2024-01-21.

[13] "Mllib | apache spark," 2019. Accessed: 2024-01-21.

[14] A. Truong, A. Walters, J. Goodsitt, K. Hines, C. B. Bruss, and R. Farivar, "Towards automated machine learning: Evaluation and comparison of automl approaches and tools," in *2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI)*, pp. 1471–1479, 2019.

[15] S. Engineering, "Transmogrifai." Accessed: 2024-01-21.

[16] T. Nagarajah and G. Poravi, "A review on automated machine learning (automl) systems," in *2019 IEEE 5th International Conference for Convergence in Technology (I2CT)*, pp. 1–6, 2019.

[17] F. Hutter, L. Kotthoff, and J. Vanschoren, eds., *Automated Machine Learning*. Springer International Publishing, 2019.

[18] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: a review," *ACM Computing Surveys*, vol. 31, pp. 264–323, 09 1999.

[19] "Sneath, p.h.a. and sokal, r.r. (1973) numerical taxonomy the principles and practice of numerical classification. wf freeman co., san francisco, 573 p. - references - scientific research publishing." Accessed: 2024-01-04.

[20] S. C. Johnson, "Hierarchical clustering schemes," *Psychometrika*, vol. 32, pp. 241–254, 09 1967.

[21] F. Murtagh and P. Legendre, "Ward's hierarchical agglomerative clustering method: Which algorithms implement ward's criterion?," *Journal of Classification*, vol. 31, pp. 274–295, 10 2014.

[22] R. Sibson, "SLINK: An optimally efficient algorithm for the single-link cluster method," *The Computer Journal*, vol. 16, pp. 30–34, 01 1973.

[23] G. N. Lance and W. T. Williams, "A General Theory of Classificatory Sorting Strategies: 1. Hierarchical Systems," *The Computer Journal*, vol. 9, pp. 373–380, 02 1967.

[24] W. H. E. Day and H. Edelsbrunner, "Efficient algorithms for agglomerative hierarchical clustering methods," *Journal of Classification*, vol. 1, pp. 7–24, 12 1984.

[25] J. H. W. Jr., "Hierarchical grouping to optimize an objective function," *Journal of the American Statistical Association*, vol. 58, no. 301, pp. 236–244, 1963.

[26] J. Irani, N. Pise, and M. Phatak, "Clustering techniques and the similarity measures used in clustering: A survey," *International Journal of Computer Applications*, vol. 134, pp. 9–14, 01 2016.

[27] B. Everitt, S. Landau, and M. Leese, *Cluster analysis.* Arnold ; New York, 2001.

[28] G. W. Milligan and M. C. Cooper, "An examination of procedures for determining the number of clusters in a data set," *Psychometrika*, vol. 50, pp. 159–179, 06 1985.

[29] A. K. Jain and R. C. Dubes, *Algorithms for Clustering Data*. 1988.

[30] J. Han, M. Kamber, and J. Pei, *Data mining : concepts and techniques*. Elsevier, 2022.

[31] M. Eisen, P. Spellman, P. Brown, and D. Botstein, "Cluster analysis and display of genome-wide expression patterns," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 95, pp. 14863–14868, Dec. 1998.

[32] J. C. Bezdek, R. Ehrlich, and W. Full, "Fcm: The fuzzy c-means clustering algorithm," *Computers Geosciences*, vol. 10, no. 2, pp. 191–203, 1984.

[33] M. Girvan and M. E. J. Newman, "Community structure in social and biological networks," *Proceedings of the National Academy of Sciences*, vol. 99, no. 12, pp. 7821–7826, 2002.

[34] "Why spring." `https://spring.io/why-spring/`. Accessed: 2024-01-23.

[35] Kotasaisrikanth, "Spring mvc framework," 12 2020. Accessed: 2024-01-23.

[36] C. Walls, *Spring Boot in Action*. Simon and Schuster, 12 2015.

[37] K. Guntupally, R. Devarakonda, and K. Kehoe, "Spring boot based rest api to improve data quality report generation for big scientific data: Arm data center example," in *2018 IEEE International Conference on Big Data (Big Data)*, pp. 5328–5329, 2018.

[38] "Top companies using spring boot." `https://www.scaler.com/topics/stories/top-companies-using-spring-boot/`, 05 2023. Accessed: 2024-01-23.

[39] Hirushirathnayake, "Json web tokens," 11 2023. Accessed: 2024-01-23.

[40] S. Ahmed and Q. Mahmood, "An authentication based scheme for applications using json web token," in *2019 22nd International Multitopic Conference (INMIC)*, pp. 1–6, 2019.

[41] Oracle, "What is mysql?." `https://www.oracle.com/mysql/what-is-mysql/`, 2021. Accessed: 2024-01-23.

[42] P. DuBois, *MySQL*. Addison-Wesley, 03 2013.

[43] P. S. Foundation, "What is python? executive summary." `https://www.python.org/doc/essays/blurb/`, 2023. Accessed:2024-01-23.

[44] G. V. Rossum and F. L. Drake, *An introduction to Python : for Python version 3.2*. Network Theory Ltd, 2011.

[45] S. Raschka, *Python Machine Learning*. Packt Publishing, 2015.

[46] "Angular." `https://angular.io/docs`, 2019. Accessed: 2024-01-23.

[47] J. Wilken, *Angular in Action*. Simon and Schuster, 03 2018.

[48] J. Cincović, S. Delev, and D. Draković, "Architecture of web applications based on angular framework: A case study," 2019.

[49] V. K. Kotaru, *Angular for Material Design*. Apress Berkeley, CA, 01 2020.

[50] A. Moiseev and Y. Fain, *Angular Development with TypeScript*. Simon and Schuster, 12 2018.

[51] "Postman api platform." `https://www.postman.com/product/what-is-postman/`. Accessed: 2024-01-23.

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