ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

“Εφαρμογή ιστού για την εκτέλεση συσταδοποίησης κ-μέσων και τον βέλτιστο προσδιορισμό της παραμέτρου κ με τη μέθοδο του αγκώνα ”

Του φοιτητή
Γράτσου Κωνσταντίνου
Αρ. Μητρώου: 185174

Επιβλέπων
Ουγιάρογλου Στέφανος
Επ. Καθηγητής

31 Ιανουαρίου 2023

Τίτλος Π.Ε.: Εφαρμογή ιστού για την εκτέλεση συσταδοποίησης κ-μέσων και τον βέλτιστο προσδιορισμό της παραμέτρου κ με τη μέθοδο του αγκώνα
Κωδικός Π.Ε. 22266
Ονοματεπώνυμο φοιτητή: Γράτσος Κωνσταντίνος
Ονοματεπώνυμο εισηγητή: Ουγιάρογλου Στέφανος
Ημερομηνία ανάληψης Δ.Ε.: 17-10-2022
Ημερομηνία περάτωσης Δ.Ε.: 31-01-2023

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 πτυχιακή εργασία, στο Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του ΔΙ.ΠΑ.Ε.

Πρόλογος

Αρκετοί είναι οι λόγοι που επέλεξα να ασχοληθώ με την συγκεκριμένη εργασία. Ένας λόγος είναι η εμπειρία που έχω με την ανάπτυξης διαδικτυακών εφαρμογών καθώς και η θέληση μου να εμβαθύνω τις γνώσεις μου στον συγκεκριμένο τομέα καθώς θέλω να τον ακολουθήσω επαγγελματικά στο μέλλον. Άλλος ένας λόγος είναι το γεγονός πως ποτέ δεν μου είχε δοθεί η ευκαιρία να ασχοληθώ πραγματικά με τον τομέα της Μηχανικής Μάθησης και της εξόρυξης δεδομένων. Επιλέγοντας αυτήν την εργασία μπόρεσα να διευρύνω τους ορίζοντες μου και να αποκτήσω μια εμπειρία πάνω στον τομέα αυτό. Επιπλέον, λόγο του ενδεχόμενου να χρησιμοποιηθεί η παρούσα εργασία στο μάθημα της σχολής "Οργάνωση Δεδομένων και Εξόρυξη Πληροφορίας" μου έδωσε ένα κίνητρο να ασχοληθώ με περισσότερη όρεξη και επαγγελματισμό.

Περίληψη

Με τον όρο συσταδοποίηση (clustering) περιγράφουμε τον διαχωρισμό των δεδομένων σε ομάδες παρόμοιων χαρακτηριστικών. Μπορεί να επιτευχθεί με διάφορους αλγόριθμους που διαφέρουν σημαντικά ως προς την κατανόησή τους. Μετά από ερεύνα εντοπίσαμε ότι δεν υπάρχουν πολλές διαδικτυακές εφαρμογές για συσταδοποίηση δεδομένων που να είναι ανοιχτές στην παγκόσμια επιστημονική κοινότητα και οι υπάρχουσες έχουν αρκετά περιορισμένες δυνατότητες. Βέβαια υπάρχουν και κάποιες λύσεις που δεν είναι εφαρμογές ιστού αλλά desktop εφαρμογές, όπως είναι το WEKA και το MATLAB. Στην συγκεκριμένη πτυχιακή εργασία θα χρησιμοποιήσουμε τον αλγόριθμο συσταδοποίησης κ-μέσων (k-means). Βασικό μειονέκτημα είναι ότι ο χρήστης πρέπει να καθορίσει το πλήθος k των συστάδων που θα ανακαλύψει ο αλγόριθμος. Συχνά όμως, ο χρήστης δεν μπορεί να προκαταβάλει τον συγκεκριμένο αριθμό και αν ο αλγόριθμος εκτελεστεί με διαφορετική τιμή k, θα δημιουργηθούν εντελώς διαφορετικές συστάδες. Ένας τρόπος για τον προσδιορισμό της παραμέτρου είναι η μέθοδος του αγκώνα (Elbow). Από τις παραπάνω εφαρμογές και προγράμματα που αναφέρθηκαν δεν υπάρχει κάποια εφαρμογή/πρόγραμμα οπού να τρέχει τη μέθοδο του αγκώνα, να προτείνει τιμή κ στον χρήστη καθώς και να κάνει την συσταδοποίηση. Συνεπώς, ο χρήστης θα πρέπει να τρέξει συνδυασμό εφαρμογών ή προγραμμάτων. Στόχος της πτυχιακής εργασίας είναι η ανάπτυξη μιας web εφαρμογής όπου ο κάθε χρήστης θα μπορεί να ανεβάζει σύνολα δεδομένων και η εφαρμογή θα κατασκευάζει το γράφημα όπου θα παρουσιάζεται ο αγκώνας και θα προτείνει στον χρήστη πιθανή τιμή για την παράμετρο κ. Στη συνέχεια, ο χρήστης θα μπορεί να κάνει την συσταδοποίηση με το προτεινόμενο κ, καθώς και να κατεβάσει το γράφημα ή το σύνολο δεδομένων όπου θα αναφέρετε η συστάδα που ανατέθηκε σε κάθε αντικείμενο του συνόλου δεδομένων.

«Web application for k-means clustering and the k parameter value determination using Elbow method»

«Konstantinos Gratsos»

Abstract

With the term clustering we describe the separation of data into groups of similar characteristics. It can be achieved with various algorithms that differ significantly in terms of their understanding. After research we found that there are not many web applications for data clustering that are open to global scientific community, and the existing ones have quite limited capabilities. Of course there are some solutions that are not web applications but desktop applications, such as WEKA and MATLAB. In this particular thesis we will use the k-means clustering algorithm .The main disadvantage is that the user must specify the number k of clusters that the algorithm will discover. However, the user cannot advance the specific number, and if the algorithm is run with a different value of k, completely different clusters will be generated. One way to determine the parameter is the elbow method. Of the above mentioned applications and programs, there is no application/program that runs the elbow method, suggests a value of k to the user as well as does the clustering , the user should run a combination of applications or programs.The goal of the thesis is the development of a web application where each user will be able to upload data sets and the application will construct the graph where the elbow will be presented and will suggest to the user a possible value for the parameter k, then the user will be able to run the clustering algorithm with the suggested k, as well as download the graph or the dataset where the clusters are assigned to each object.

Ευχαριστίες

Ξεκινώντας, θα ήθελα να ευχαριστήσω τον κ. Στέφανο Ουγιάρογλου, για την εμπιστοσύνη που μου έδειξε για την ανάληψη και ολοκλήρωση αυτής της πτυχιακής, τον ευχαριστώ για τον χρόνο τον οποίο διέθεσε, η καθοδήγηση, η προθυμία του και η συμπαράστασή του κατά τη συγγραφή της εργασίας, ήταν καθοριστική. Έπειτα, θα ήθελα να ευχαριστήσω την οικογένειά μου που με υποστήριξε σε όλη την διάρκεια των σπουδών μου.

Περιεχόμενα

Πρόλογος . . . ii
Περίληψη . . . iii
Abstract . . . iv
Ευχαριστίες . . . v
Περιεχόμενα . . . vi
Κατάλογος Σχημάτων . . . viii
Κατάλογος Πινάκων . . . ix

1 Εισαγωγή **1**
1.1 Συσταδοποίηση δεδομένων . . . 1
1.2 Τύποι συστάδων . . . 2
1.3 Τύποι αλγορίθμων συσταδοποίησης . . . 5
1.4 Συσταδοποίηση:Περιπτώσεις χρήσεις . . . 11
1.5 Κίνητρο και Συνεισφορά . . . 12
1.6 Οργάνωση της εργασίας . . . 14

2 Συσταδοποίηση k-means **15**
2.1 Ο αλγόριθμος κ-μέσων . . . 15
2.2 Πλεονεκτήματα και Μειονεκτήματα . . . 19
2.3 Προσδιορισμός της παραμέτρου κ . . . 21
2.3.1 Η μέθοδος του αγκώνα (elbow method) . . . 22
2.3.2 Η μέθοδος silhouette score . . . 23
2.3.3 Η μέθοδος στατιστική χάσματος(gap statistic) . . . 23
2.3.4 Η μέθοδοοι Bayesian Information Criteria (BIC) και Akaike Information Criterion (AIC) . . . 24
2.4 Παραλλαγές του αλγορίθμου κ-μέσων . . . 25

3 Τεχνολογίες που χρησιμοποιήθηκαν **27**
3.1 Back-end . . . 27
3.1.1 PHP . . . 27
3.1.2 Composer . . . 28
3.1.3 REST API . . . 28
3.1.4 Python . . . 30
3.1.5 MySQL . . . 31
3.1.6 XAMPP . . . 32
3.1.7 Postman . . . 33
3.2 Front-end . . . 33
3.2.1 HTML . . . 33
3.2.2 CSS . . . 34
3.2.3 Bootstrap . . . 35
3.2.4 JavaScript . . . 35
3.2.5 jQuery . . . 36

4 Σχεδίαση και Υλοποίηση του web k-means **38**
4.1 Λειτουργικές Απαιτήσεις . . . 38
4.2 Αρχιτεκτονική . . . 40
4.3 Δημόσια σύνολα δεδομένων και χρήστες . . . 43
4.4 Back-end . . . 43
4.4.1 Βάση δεδομένων . . . 43
4.4.2 Υλοποίηση του REST API . . . 46
4.4.3 Υλοποίηση του elbow module . . . 50
4.4.4 Υλοποίηση του K-Means module . . . 51
4.4.5 Απόδοση του k-means . . . 52
4.5 Υλοποίηση του Front-end . . . 53
4.6 Επίλογος . . . 56

5 Παρουσίαση του REST API **57**
5.1 Μέθοδοι GET 57
5.2 Μέθοδοι POST 59
5.3 Μέθοδοι DELETE 62

6 Παρουσίαση του web k-means **63**
6.1 Αρχική σελίδα 63
6.2 Φόρμες της εφαρμογής 64
6.3 Σελίδα k-means 67
6.4 Σελίδα API Docs 71
6.5 Επίλογος 72

7 Συμπεράσματα και μελλοντικές επεκτάσεις **73**
7.1 Συμπεράσματα 73
7.2 Μελλοντικές επεκτάσεις 73

ΒΙΒΛΙΟΓΡΑΦΙΑ **75**

Κατάλογος Σχημάτων

1.1 Καλά διαχωρισμένες συστάδες . . . 2
1.2 Συστάδες που βασίζονται στο κέντρο . . . 3
1.3 Συστάδες που βασίζονται σε γειτνίαση . . . 3
1.4 Συστάδες που βασίζονται στην πυκνότητα . . . 4
1.5 Εννοιολογικές συστάδες . . . 4
1.6 Παράδειγμα DBSCAN . . . 6
1.7 Συσταδοποίηση με την μέθοδο DBSCAN . . . 7
1.8 Συσταδοποίηση με την μέθοδο OPTICS . . . 7
1.9 Κατάτμηση δεδομένων . . . 9
1.10 Δενδρόγραμμα . . . 9
1.11 Παράδειγμα υπερπροσαρμογής (overfitting) . . . 11
1.12 Παράδειγμα αποτυχίας του k-means . . . 11
2.1 Χρησιμοποιώντας τον αλγόριθμο K-means για την εύρεση τριών συστάδων . . . 16
2.2 Εκτέλεση του k-means σε σύνολο δεδομένων με διαφορετικά μεγέθη . . . 20
2.3 Εκτέλεση του k-means σε σύνολο δεδομένων με διαφορετική πυκνότητα . . . 20
2.4 Εκτέλεση του k-means σε σύνολο δεδομένων μη-σφαιρικό . . . 21
2.5 Το σημείο του αγκώνα στο elbow method . . . 23
3.1 Λογότυπο PHP . . . 27
3.2 Λογότυπο Composer . . . 28
3.3 Λογότυπο Python . . . 30
3.4 Λογότυπο MySQL . . . 31
3.5 Λογότυπο XAMPP . . . 32
3.6 Λογότυπο Postman . . . 33
3.7 Λογότυπο HTML . . . 34
3.8 Λογότυπο CSS . . . 34
3.9 Παράδειγμα CSS . . . 35
3.10 Λογότυπο Bootstrap . . . 35
3.11 Λογότυπο JavaScript . . . 36
3.12 Λογότυπο jQuery . . . 36
4.1 Διάγραμμα ροής της εφαρμογής . . . 41
4.2 Διάγραμμα αρχιτεκτονικής της εφαρμογής . . . 42
4.3 Πίνακας users . . . 44
4.4 Πίνακας verification_tokens . . . 45
4.5 Διάγραμμα ER της εφαρμογής . . . 45
4.6 Παράδειγμα αποθηκευμένης διαδικασίας . . . 46
4.7 Κώδικας για έλεγχο σωστών παραμέτρων . . . 47
4.8 Κώδικας για τον έλεγχο του τύπου του dataset . . . 48
4.9 Κώδικας για τον έλεγχο του API Key . . . 48
4.10 Κώδικας για τον έλεγχο αριθμητικών δεδομένων . . . 48
4.11 Κώδικας για την σύνδεση με την βάση δεδομένων . . . 49
4.12 Κώδικας για αποστολή email επιβεβαίωσης . . . 49
4.13 Κώδικας για την κλήση ενός Python script μέσα από την PHP . . . 50
4.14 Κώδικας για ανέβασμα δημόσιου συνόλου δεδομένων . . . 50
4.15 Κώδικας για διάβασμα αρχείου . . . 51
4.16 Κώδικας για κανονικοποίηση των δεδομένων . . . 51
4.17 Κώδικας για τον υπολογισμό του αθροίσματος των τετραγωνικών αποστάσεων . . . 51
4.18 Κώδικας για τον υπολογισμό της προτεινόμενης τιμής κ . . . 52
4.19 Κώδικας για την συσταδοιποίηση k-means . . . 52
4.20 HTML αρχεία . . . 53
4.21 CSS αρχεία . . . 54
4.22 Κώδικας για κλήση πρός το API . . . 55
4.23 Κώδικας στην περίπτωση success . . . 55
4.24 Κώδικας στην περίπτωση error . . . 55
4.25 Κώδικας για χρήση του sessionStorage . . . 56
6.1 Πάνω μέρος της αρχικής σελίδας . . . 63
6.2 Κάτω μέρος της αρχικής σελίδας . . . 64

6.3 Φόρμα register . . . 65
6.4 Προειδοποίηση για email επαλήθευσης . . . 65
6.5 Φόρμα login . . . 66
6.6 Φόρμα forgot password . . . 66
6.7 Φόρμα για ορισμό νέου κωδικού πρόσβασης . . . 67
6.8 Φόρμα για ορισμό νέου κωδικού πρόσβασης . . . 67
6.9 Τμήμα επιλογής dataset . . . 68
6.10 Φόρμα για ανέβασμα dataset . . . 68
6.11 Πίνακας με τα δεδομένα του dataset . . . 69
6.12 Αριθμητικές στήλες του dataset . . . 69
6.13 Τμήμα για την εκτέλεση της μεθόδου του αγκώνα . . . 69
6.14 Γράφημα για την μέθοδο του αγκώνα . . . 70
6.15 Τμήμα για την εκτέλεση συσταδοποίησης κ-μέσων . . . 70
6.16 Πίνακας με τα δεδομένα συστοδοποίσης . . . 70
6.17 Σελίδα API Docs . . . 71
6.18 Οδηγίες προς κάποιο endpoint . . . 71

Κατάλογος Πινάκων

2.1 Υπολογισμός των αποστάσεων . . . 18
2.2 Επανυπολογισμός των αποστάσεων . . . 19
4.1 Τα scripts του REST API . . . 46
4.2 Πειραματικές μετρήσεις . . . 53
5.1 Τα endpoints του REST API . . . 57

Κεφάλαιο 1ο: Εισαγωγή

1.1 Συσταδοποίηση δεδομένων

Με τον όρο συσταδοποίηση (clustering) περιγράφουμε τον διαχωρισμό των δεδομένων σε ομάδες παρόμοιων χαρακτηριστικών. Κάθε ομάδα ονομάζεται συστάδα και αποτελείται από αντικείμενα που είναι παρόμοια μεταξύ τους και ανόμοια σε σύγκριση με άλλες ομάδες. Η συσταδοποίηση δημιουργήθηκε στη ψυχολογία από τον Zubin (1938) και Tryon (1939) και στην ανθρωπολογία των Driver και Kroeber (1932). Οι υπολογιστικές δυσκολίες καθυστέρησαν την ανάπτυξή του μέχρι τα τέλη της δεκαετίας του 1950, όταν η μηχανογράφηση είχε ως αποτέλεσμα τον πολλαπλασιασμό των τεχνικών συσταδοποίησης. Η ανάπτυξη συνεχίστηκε στην ψυχολογία, ιδίως σε πολλές εργασίες του McQu [1].

Είναι σημαντικό να αναγνωρίσουμε την διαφορά ανάμεσα στην συσταδοποίση και την κατηγοριοποίηση (Classification). Στην μηχανική μάθηση η κατηγοριοποίηση εντάσσεται στους αλγορίθμους με εποπτευόμενη μάθηση. Η προσέγγιση εποπτευόμενης μάθησης χρησιμοποιεί σύνολα δεδομένων με ετικέτα που εκπαιδεύουν αλγόριθμους για την ταξινόμηση δεδομένων ή την ακριβή πρόβλεψη αποτελεσμάτων. Το μοντέλο χρησιμοποιεί τα δεδομένα με ετικέτα για να μετρήσει τη συνάφεια διαφορετικών χαρακτηριστικών για να βελτιώσει σταδιακά την προσαρμογή του μοντέλου στο γνωστό αποτέλεσμα. Η συσταδοποίηση εντάσσεται στους αλγορίθμους της μάθησης χωρίς επίβλεψη. Στην μάθηση χωρίς επίβλεψη, οι αλγόριθμοι χρησιμοποιούνται για την εξέταση και την ομαδοποίηση συνόλων δεδομένων χωρίς ετικέτα. Τέτοιοι αλγόριθμοι μπορούν να αποκαλύψουν άγνωστα μοτίβα σε δεδομένα χωρίς ανθρώπινη επίβλεψη. Η διάκριση μεταξύ επισημασμένων και μη επισημασμένων συνόλων δεδομένων είναι η βασική διαφορά μεταξύ των δύο προσεγγίσεων. Η εποπτευόμενη μάθηση χρησιμοποιεί σύνολα δεδομένων με ετικέτα για την εκπαίδευση αλγορίθμων κατηγοριοποίησης. Τα δεδομένα με την ένδειξη «εκπαίδευση» τροφοδοτούνται και το μοντέλο προσαρμόζει επαναληπτικά τον τρόπο με τον οποίο ζυγίζει τα διαφορετικά χαρακτηριστικά των δεδομένων έως ότου το μοντέλο προσαρμοστεί κατάλληλα στο επιθυμητό αποτέλεσμα. Τα μοντέλα εποπτευόμενης μάθησης είναι πολύ πιο ακριβή από την αντίστοιχη προσέγγιση. Ωστόσο, απαιτούν την συμμετοχή των ανθρώπων στη διαδικασία της επεξεργασίας των δεδομένων για να διασφαλιστεί ότι οι ετικέτες στις πληροφορίες είναι κατάλληλες.

Ένα παράδειγμα είναι ότι ένα εποπτευόμενο μοντέλο εκμάθησης μπορεί να προβλέψει τους χρόνους πτήσης με βάση τις ώρες αιχμής σε ένα αεροδρόμιο, την κυκλοφοριακή συμφόρηση στον αέρα και τις καιρικές συνθήκες (εκτός από άλλες πιθανές παραμέτρους). Αλλά οι άνθρωποι πρέπει να επέμβουν για να επισημάνουν τα σύνολα δεδομένων για να εκπαιδεύσουν το μοντέλο σχετικά με το πώς αυτοί οι παράγοντες μπορούν να επηρεάσουν τους χρόνους πτήσης. Ένα εποπτευόμενο μοντέλο εξαρτάται από τη γνώση του αποτελέσματος για να καταλήξει στο συμπέρασμα ότι το η καταιγίδα είναι ένας παράγοντας για την καθυστέρηση των πτήσεων.

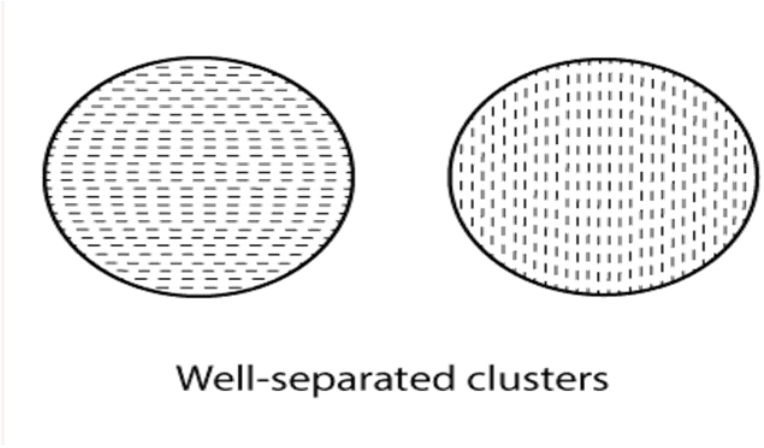
Αντίθετα, τα μοντέλα μάθησης χωρίς επίβλεψη λειτουργούν χωρίς ανθρώπινη παρέμβαση. Βρίσκουν και καταλήγουν σε μια δομή του είδους χρησιμοποιώντας δεδομένα χωρίς ετικέτα. Η μόνη ανθρώπινη βοήθεια που χρειάζεται εδώ είναι για την επικύρωση των μεταβλητών εξόδου. Για παράδειγμα, όταν κάποιος αγοράζει ένα νέο φορητό υπολογιστή στο διαδίκτυο, ένα μοντέλο εκμάθησης χωρίς επίβλεψη θα καταλάβει ότι το άτομο ανήκει σε μια ομάδα αγοραστών που αγοράζουν μαζί ένα σύνολο σχετικών προϊόντων. Ωστόσο, είναι δουλειά ενός αναλυτή δεδομένων να επικυρώσει ότι ένας σύστημα συστάσεων

προσφέρει επιλογές για τσάντα του υπολογιστή, προστατευτικό οθόνης και φορτιστή αυτοκινήτου. [2]

1.2 Τύποι συστάδων

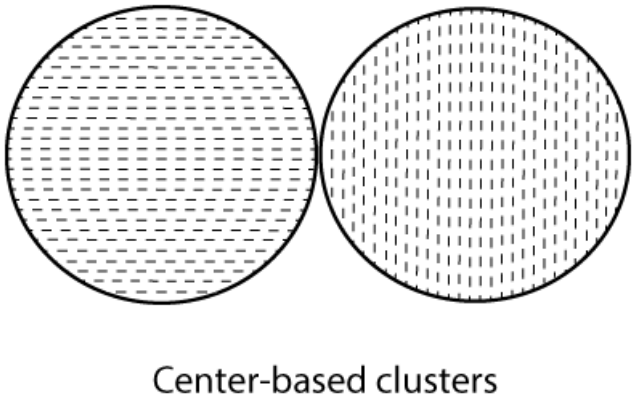
Η συσταδοποίηση στοχεύει στην εύρεση ομάδων αντικειμένων (συστάδες), η χρησιμότητα της ορίζεται από τους στόχους της ανάλυσης δεδομένων.Υπάρχουν αρκετές διαφορετικές έννοιες μιας συστάδας που αποδεικνύονται χρήσιμες στην πράξη.

Καλά Διαχωρισμένες(Well-Separated).
Μία συστάδα είναι ένα σύνολο αντικειμένων στα οποία κάθε αντικείμενο είναι πιο κοντά (ή πιο όμοιο) με κάθε άλλο αντικείμενο της συστάδας παρά με οποιοδήποτε αντικείμενο που δεν βρίσκεται στη συστάδα.Μερικές φορές χρησιμοποιείται ένα κατώφλι για να προσδιορίσει ότι όλα τα αντικείμενα σε μία συστάδα πρέπει να είναι αρκετά κοντά ή παρόμοια το ένα με το άλλο.Αυτός ο ορισμός μιας συστάδας ικανοποιείται μόνο όταν τα δεδομένα περιέχουν φυσικές συστάδες που είναι αρκετά μακριά η μία από την άλλη. Η απόσταση μεταξύ οποιωνδήποτε δύο σημείων σε διαφορετικές ομάδες είναι μεγαλύτερη από την απόσταση μεταξύ οποιωνδήποτε δύο σημείων σε μια ομάδα.Οι Καλά διαχωρισμένες συστάδες δεν χρειάζεται να είναι σφαιρικές, αλλά μπορεί να έχουν οποιοδήποτε σχήμα (Σχήμα 1.1).



Σχήμα 1.1: Καλά διαχωρισμένες συστάδες

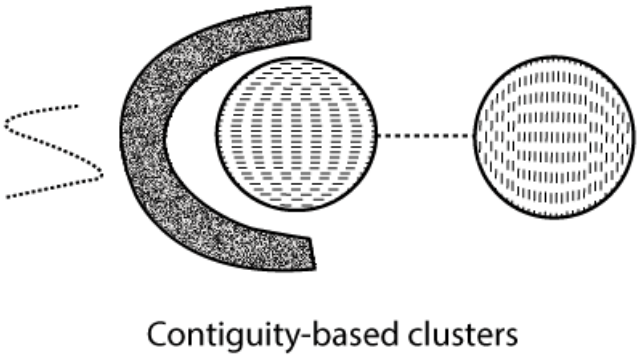
Βασισμένο σε πρότυπο(Prototype-Based).
Μία συστάδα είναι ένα σύνολο αντικειμένων στα οποία κάθε αντικείμενο είναι πιο κοντά (πιο παρόμοιο) στο πρότυπο που ορίζει η συστάδα παρά στο πρότυπο οποιασδήποτε άλλης συστάδας. Για δεδομένα με συνεχή χαρακτηριστικά, το πρότυπο μίας συστάδας είναι συχνά ένα κέντρο, δηλαδή ο μέσος όρος (μέσος όρος) όλων των σημείων της συστάδας.Για πολλούς τύπους δεδομένων, το πρότυπο μπορεί να θεωρηθεί ως το πιο κεντρικό σημείο, και σε τέτοιες περιπτώσεις, συνήθως αναφερόμαστε σε συστάδες που βασίζονται σε πρότυπα ως συστάδες που βασίζονται στο κέντρο(Σχήμα 1.2). Δεν αποτελεί έκπληξη το γεγονός ότι τέτοιες συστάδες τείνουν να είναι σφαιρικές [3].



Σχήμα 1.2: Συστάδες που βασίζονται στο κέντρο

Βάσει γραφήματος(Graph-Based).

Εάν τα δεδομένα αντιπροσωπεύονται ως γράφημα, όπου οι κόμβοι είναι αντικείμενα και οι σύνδεσμοι αντιπροσωπεύουν συνδέσεις μεταξύ αντικειμένων, τότε μια συστάδα μπορεί να οριστεί ως συνδεδεμένο στοιχείο, δηλαδή μια ομάδα αντικειμένων που συνδέονται μεταξύ τους, αλλά δεν έχουν καμία σύνδεση με αντικείμενα εκτός της ομάδας. Ένα σημαντικό παράδειγμα συστάδων που βασίζονται σε γραφήματα είναι οι συστάδες που βασίζονται σε γειτνίαση, όπου δύο αντικείμενα συνδέονται μόνο εάν βρίσκονται σε καθορισμένη απόσταση το ένα από το άλλο. Αυτός ο ορισμός μιας συστάδας είναι χρήσιμος όταν οι συστάδες είναι ακανόνιστες ή αλληλένδετες, αλλά μπορεί να έχουν πρόβλημα όταν υπάρχει θόρυβος, καθώς, όπως φαίνεται από τις δύο σφαιρικές συστάδες του Σχήματος 1.3, μια μικρή γέφυρα σημείων μπορεί να συγχωνεύσει δύο ξεχωριστές συστάδες [3].

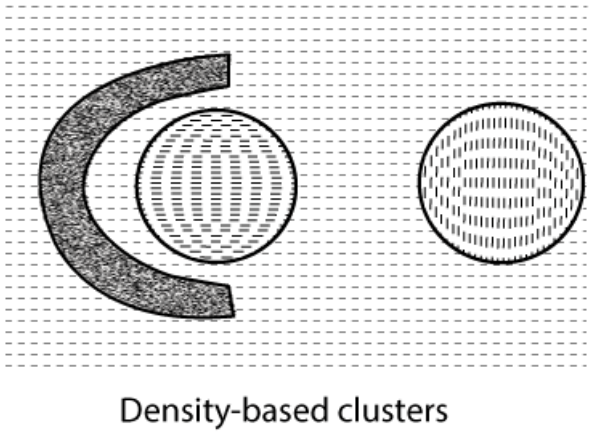


Σχήμα 1.3: Συστάδες που βασίζονται σε γειτνίαση

Με βάση την πυκνότητα(Density-Based).

Μία συστάδα είναι μια πυκνή περιοχή αντικειμένων που περιβάλλεται από μια περιοχή χαμηλής πυκνότητας. Το Σχήμα 1.4 δείχνει ορισμένες συστάδες που βασίζονται στην πυκνότητα για δεδομένα που δημιουργούνται με την προσθήκη θορύβου στα δεδομένα του Σχήματος 1.3. Οι δύο κυκλικές συστάδες δεν συγχωνεύονται, όπως στο Σχήμα 1.3, επειδή η γέφυρα μεταξύ τους εξασθενεί στον θόρυβο. Ένας

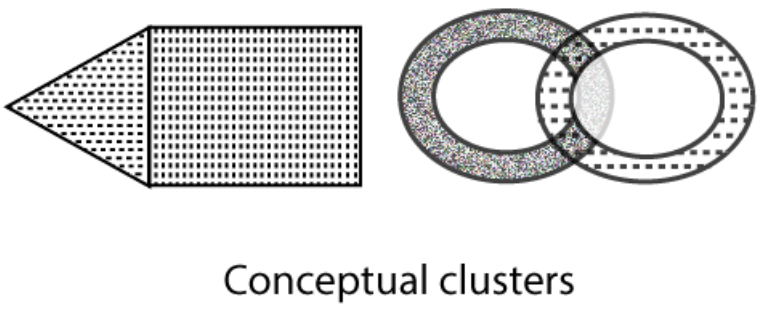
ορισμός με βάση την πυκνότητα μίας συστάδας χρησιμοποιείται συχνά όταν οι συστάδες είναι ακανόνιστες ή Αντίθετα, ένας ορισμός μίας συστάδας που βασίζεται στη γειτνίαση δεν θα λειτουργούσε καλά για τα δεδομένα του Σχήματος 1.4, καθώς ο θόρυβος θα έτεινε να σχηματίζει γέφυρες μεταξύ των συστάδων όταν υπάρχει θόρυβος και ακραίες τιμές. Αντίθετα, ένας ορισμός συστάδας που βασίζεται στη γειτνίαση δεν θα λειτουργούσε καλά για τα δεδομένα του Σχήματος 1.4, καθώς ο θόρυβος θα έτεινε να σχηματίζει γέφυρες μεταξύ των συστάδων [3].



Σχήμα 1.4: Συστάδες που βασίζονται στην πυκνότητα

Εννοιολογική συστάδα(Conceptual Clusters).

Γενικότερα, μπορούμε να ορίσουμε μία συστάδα ως ένα σύνολο αντικειμένων που μοιράζονται κάποια ιδιότητα. Αυτός ο ορισμός περιλαμβάνει όλους τους προηγούμενους ορισμούς μιας συστάδας π.χ. τα αντικείμενα σε μία συστάδα που βασίζεται στο κέντρο μοιράζονται την ιδιότητα ότι είναι όλα πιο κοντά στο ίδιο κέντρο. Ωστόσο, η προσέγγιση της κοινής ιδιότητας περιλαμβάνει επίσης νέους τύπους συστάδων. Για παράδειγμα, στο σχήμα 1.5, μία τριγωνική περιοχή (σμήνος) γειτνιάζει με μια ορθογώνια και υπάρχουν δύο πλεγμένοι κύκλοι (συστάδες). Και στις δύο περιπτώσεις, ένας αλγόριθμος συσταδοποίησης θα χρειαζόταν μια πολύ συγκεκριμένη ιδέα μιας συστάδας για να ανιχνεύσει με επιτυχία αυτές τις συστάδες. Η διαδικασία εύρεσης τέτοιων συστάδων ονομάζεται εννοιολογική ομαδοποίηση(Conceptual Clusters) [3].



Σχήμα 1.5: Εννοιολογικές συστάδες

1.3 Τύποι αλγορίθμων συσταδοποίησης

Σε αυτήν την ενότητα θα κάνουμε μια ανασκόπηση των πιο σημαντικών και δημοφιλέστερων αλγορίθμων συσταδοποίησης. Δεν θα αναφερθούμε σε όλους καθώς υπάρχουν πάνω από 100 δημοσιευμένοι αλγόριθμοι. Δεν υπάρχει αντικειμενικά «σωστός» αλγόριθμος συσταδοποίησης, αλλά όπως αναφέρεται, «η συσταδοποίηση είναι στο μάτι του θεατή». Ο πιο κατάλληλος αλγόριθμος ομαδοποίησης για ένα συγκεκριμένο πρόβλημα συχνά χρειάζεται να επιλέγεται πειραματικά, εκτός εάν υπάρχει μαθηματικός λόγος για να προτιμάτε ένα μοντέλο συσταδοποίησης έναντι ενός άλλου. Ένας αλγόριθμος που έχει σχεδιαστεί για ένα είδος μοντέλου γενικά θα αποτύχει σε ένα σύνολο δεδομένων που περιέχει ένα ριζικά διαφορετικό είδος μοντέλου [4].

Density-based Clustering.

Η συσταδοποίηση με τον αλγόριθμο Density-based λειτουργεί ανιχνεύοντας περιοχές όπου είναι συγκεντρωμένα τα αντικείμενα(points) και όπου χωρίζονται από περιοχές που είναι κενές ή αραιές. Τα αντικείμενα που δεν αποτελούν μέρος μιας συστάδας χαρακτηρίζονται ως θόρυβος(noise). Η πιο δημοφιλής μέθοδος είναι η DBSCAN. Σε αντίθεση με πολλές νεότερες μεθόδους, διαθέτει ένα καλά καθορισμένο μοντέλο συσταδοποίησης που ονομάζεται "πυκνότητα-προσβασιμότητα". Βασίζεται στην σύνδεση των αντικείμενα εντός ορισμένων ορίων απόστασης. Ωστόσο, συνδέει μόνο αντικείμενα που ικανοποιούν ένα κριτήριο πυκνότητας(Density), στην αρχική παραλλαγή που ορίζεται ως ένας ελάχιστος αριθμός άλλων αντικειμένων εντός αυτής της ακτίνας. Μια συστάδα αποτελείται από όλα τα αντικείμενα που συνδέονται με την πυκνότητα (τα οποία μπορούν να σχηματίσουν μια συστάδα αυθαίρετου σχήματος, σε αντίθεση με πολλές άλλες μεθόδους) συν όλα τα αντικείμενα που βρίσκονται εντός της εμβέλειας αυτών των αντικειμένων. Ακόμα μια ενδιαφέρουσα ιδιότητα του DBSCAN είναι ότι η πολυπλοκότητά του είναι αρκετά χαμηλή, απαιτεί γραμμικό αριθμό ερωτημάτων εύρους στη βάση δεδομένων και ότι ανακαλύπτει ουσιαστικά τα ίδια αποτελέσματα σε κάθε εκτέλεση, επομένως δεν χρειάζεται να εκτελεστεί πολλές φορές [5]. Η κύρια ιδέα του αλγορίθμου DBSCAN είναι να εντοπίζει περιοχές υψηλής πυκνότητας που χωρίζονται μεταξύ τους από περιοχές χαμηλής πυκνότητας. Η μέθοδος με την οποία μετράμε την πυκνότητα έχει ως εξής: Πυκνότητα σε σημείο P: Αριθμός αντικειμένων εντός κύκλου ακτίνας Eps (ε) από το σημείο P. Πυκνή περιοχή: Για κάθε αντικείμενο της συστάδας, ο κύκλος με ακτίνα □ περιέχει τουλάχιστον ελάχιστο αριθμό αντικειμένων(MinPts). Η γειτονιά Epsilon ενός σημείου P στη βάση δεδομένων D ορίζεται ως (ακολουθώντας τον ορισμό από τους Ester et.al.)

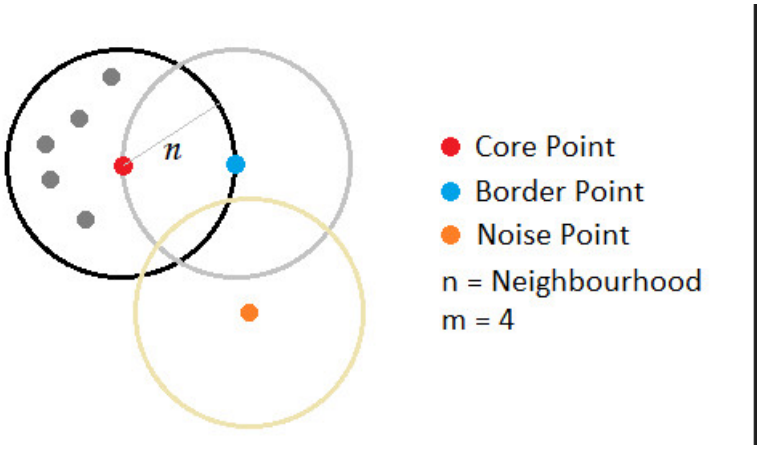
$$N(p) = \{q \varepsilon D | dist(p,q) \leq \varepsilon\} \tag{1.1}$$

Η βασική εργασία του αλγορίθμου είναι να χωρίσει τα αντικείμενα σε 3 κατηγορίες οι οποίες είναι οι εξής: Κέντρο(core): Ακολουθώντας τον ορισμό της πυκνής περιοχής, ένα σημείο μπορεί να ταξινομηθεί ως Κεντρικό Σημείο(core) εάν |N (p)|≥ MinPts. Τα κεντρικά σημεία, όπως υποδηλώνει το όνομα, βρίσκονται συνήθως στο εσωτερικό μίας συστάδας. Όριο(border): Ένα Οριακό σημείο έχει λιγότερα MinPts εντός της ε-γειτονιάς του (N), αλλά βρίσκεται στη γειτονιά ενός άλλου κεντρικού σημείου. Θόρυβος(noise): Θόρυβος είναι οποιοδήποτε σημείο δεδομένων που δεν είναι ούτε κεντρικό ούτε οριακό σημείο [6]. Δείτε το σχήμα 1.6 για καλύτερη κατανόηση.
Ο αλγοριθμός του DBSCAN εχει ως εξής:

Algorithm 1 DBSCAN

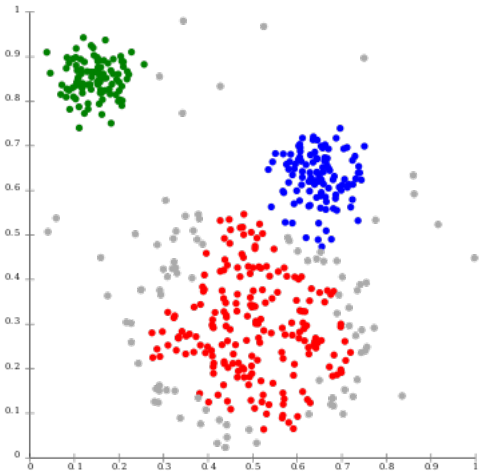
```
1: for each o ε D do
2:     if o is not yet classified then
3:         if o is a core-object then
4:             collect all objects density-reachable from o and assign them to a new cluster
5:         end if
6:     else
7:         assign o to NOISE
8:     end if
9: end for
```

Ο αλγόριθμος εκτελείτε για κάθε αντικείμενο στο σύνολο δεδομένων, στην δεύτερη γραμμή ελέγχει αν το αντικείμενο έχει οριστεί, στην 3 γραμμή ελέγχει αν το αντικείμενο είναι core και τότε συλλέγει όλα τα αντικείμενα που είναι προσβάσιμα βάση της πυκνότητας του συγκεκριμένου ο και τα αναθέτει σε καινούργια συστάδα.

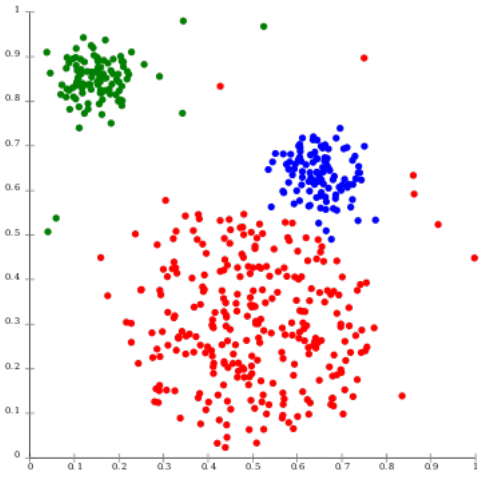


Σχήμα 1.6: Παράδειγμα DBSCAN

Η μέθοδος OPTICS είναι μια γενίκευση του DBSCAN που καταργεί την ανάγκη επιλογής κατάλληλης τιμής για την παράμετρο εύρους. Στα σχήματα 1.7 και 1.8 φαίνεται η διαφορά του DBSCAN και του OPTICS. Στο σχήμα 1.7 το DBSCAN υποθέτει συστάδες παρόμοιας πυκνότητας και μπορεί να έχει προβλήματα με τον διαχωρισμό κοντινών συστάδων, στο σχήμα 1.8 Το OPTICS βελτιώνει τον χειρισμό των συστάδων διαφορετικών πυκνοτήτων. Το βασικό μειονέκτημα του DBSCAN και του OPTICS είναι ότι αναμένουν την πτώση της τιμής της πυκνότητας για την ανίχνευση ορίων σε μια συστάδα. [5]



Σχήμα 1.7: Συσταδοποίηση με την μέθοδο DBSCAN



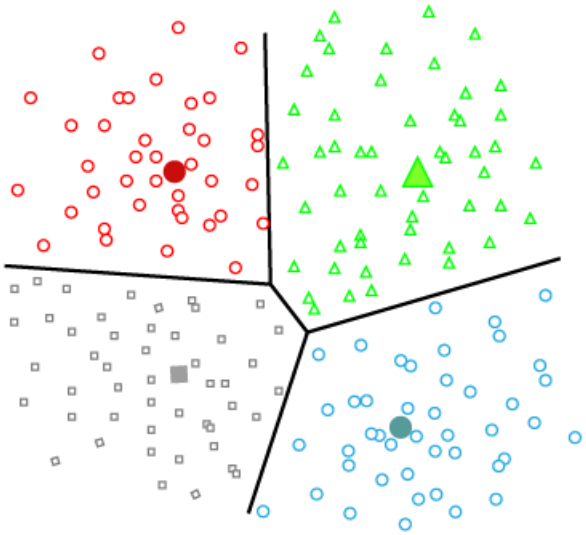
Σχήμα 1.8: Συσταδοποίηση με την μέθοδο OPTICS

Partitioning-based Clustering.

Είναι ίσως η πιο δημοφιλής κατηγορία αλγορίθμων συσταδοποίησης. Είναι αλγόριθμοι συνδυαστικής βελτιστοποίησης γνωστοί και ως αλγόριθμοι επαναληπτικής μετεγκατάστασης. Η τμηματική συσταδοποίηση εκχωρεί ένα σύνολο αντικειμένων σε k-cluster χρησιμοποιώντας επαναληπτικές διεργασίες. Σε αυτές τις διαδικασίες n δεδομένα ταξινομούνται σε k συστάδες. Κάθε συστάδα αντιπροσωπεύεται από ένα κεντρικό διάνυσμα(centroid), το οποίο δεν είναι απαραίτητα μέλος του συνόλου δεδομένων [7]. Υπάρχουν πολλοί αλγόριθμοι που υπάγονται στη μέθοδο κατάτμησης ορισμένοι από τους δημοφιλείς είναι οι K-Means, είναι ένας από τους πιο συχνά χρησιμοποιούμενους αλγόριθμους για την κατάτμηση ενός συνόλου δεδομένων σε ένα σύνολο k συστάδων. Ταξινομεί αντικείμενα σε πολλαπλές συστάδες έτσι ώστε τα αντικείμενα εντός της ίδιας συστάδας να είναι όσο το δυνατόν παρόμοια, ενώ τα αντικείμενα από διαφορετικές συστάδες είναι όσο το δυνατόν ανόμοια. Στην συσταδοποίηση k-means, κάθε συστάδα αντιπροσωπεύεται από το κέντρο της που αντιστοιχεί στον μέσο όρο των αντικειμένων που έχουν εκχωρηθεί σε αυτή. Η βασική ιδέα πίσω από την συσταδοποίηση k-means αποτελείται από τον καθορισμό συστάδων έτσι ώστε η συνολική διακύμανση εντός του συστάδας (γνωστή ως συνολική διακύμανση εντός της συστάδας) να ελαχιστοποιείται,ο αλγόριθμος kmeans++ εξασφαλίζει μια πιο έξυπνη προετοιμασία των κεντροειδών(centroids) και βελτιώνει την ποιότητα της συσταδοποίησης, ο αλγόριθμος k-modes όπου είναι για κατηγορικά τύπου δεδομένα, επίσης υπάρχουν και οι K-Medoids όπου σε αντίθεση με τον αλγόριθμο k-means επιλέγει πραγματικά σημεία δεδομένων ως κέντρα,ο αλγόριθμος CLARA (Clustering Large Applications) επεκτείνει την προσέγγιση k-medoids για μεγάλα σύνολα δεδομένων [8]. Οι περισσότεροι αλγόριθμοι τύπου k-means απαιτούν να καθοριστεί εκ των προτέρων ο αριθμός των συστάδων k, κάτι που θεωρείται ένα από τα μεγαλύτερα μειονεκτήματα αυτών των αλγορίθμων. Επιπλέον, οι αλγόριθμοι προτιμούν συστάδες περίπου παρόμοιου μεγέθους, καθώς θα εκχωρούν πάντα ένα αντικείμενο στο πλησιέστερο κέντρο. Ο αριθμός των διαφορετικών συνδυασμών υπολογίζεται ως εξής:

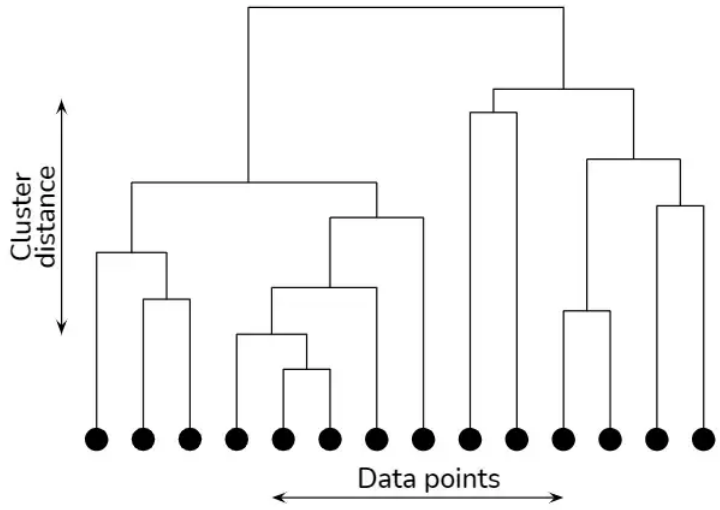
$$\sum_{i=1}^{i=Kmax} S_n^{(i)} \tag{1.2}$$

όπου Kmax είναι ο μέγιστος αριθμός συστάδων και είναι προφανές ότι Kmax <=n [7]. Αυτό συχνά οδηγεί σε εσφαλμένα αποτελέσματα αφού ο χρήστης δεν μπορεί να προκαταβάλει τον συγκεκριμένο αριθμό και αν ο αλγόριθμος εκτελεστεί με διαφορετική τιμή k, θα δημιουργηθούν εντελώς διαφορετικές συστάδες. Στο δεύτερο κεφάλαιο αναλύεται με περισσότερη λεπτομέρεια ο αλγόριθμος k-means και οι παραλλαγές του διότι η παρούσα εργασία βασίζεται σε αυτόν τον αλγόριθμο.



Σχήμα 1.9: Κατάτμηση δεδομένων

Hierarchical-based Clustering.
Στην ιεραρχική συσταδοποίηση βασική ιδέα είναι ότι τα αντικείμενα σχετίζονται περισσότερο με κοντινά αντικείμενα παρά με αντικείμενα πιο μακριά. Αυτοί οι αλγόριθμοι συνδέουν αντικείμενα για να σχηματίσουν συστάδες με βάση την απόστασή τους. Μια συστάδα μπορεί να περιγραφεί σε μεγάλο βαθμό από τη μέγιστη απόσταση που απαιτείται για τη σύνδεση τμημάτων της συστάδας. Σε διαφορετικές αποστάσεις, θα σχηματιστούν διαφορετικές συστάδες, οι οποίες μπορούν να αναπαρασταθούν χρησιμοποιώντας ένα δενδρόγραμμα, το οποίο εξηγεί από πού προέρχεται το κοινό όνομα ”ιεραρχική συσταδοποίηση” [3]. Σε ένα δενδρόγραμμα, ο άξονας y σηματοδοτεί την απόσταση στην οποία συγχωνεύονται οι συστάδες, ενώ τα αντικείμενα τοποθετούνται κατά μήκος του άξονα x έτσι ώστε οι συστάδες να μην αναμειγνύονται(Σχήμα 1.10).

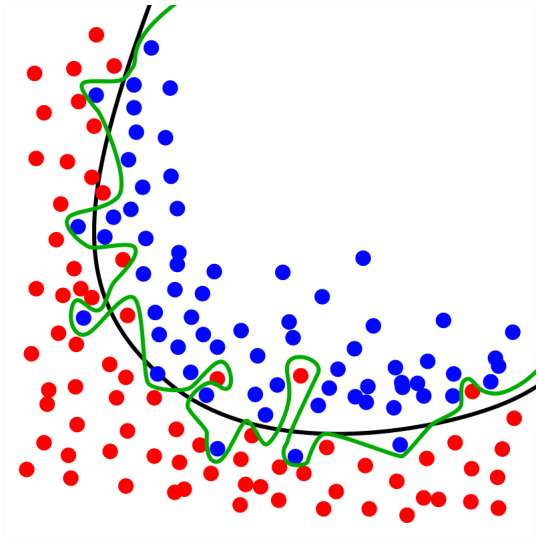


Σχήμα 1.10: Δενδρόγραμμα

Στην ιεραρχική συσταδοποίηση, κατά την κατασκευή του δενδρογράμματος, δεν κρατάμε καμία υπόθεση για τον αριθμό των συστάδων. Μόλις κατασκευαστεί το δενδρόγραμμα, κόβουμε αυτή τη δομή οριζόντια. Όλοι οι προκύπτοντες θυγατρικοί κλάδοι που σχηματίζονται κάτω από την οριζόντια τομή αντιπροσωπεύουν μία μεμονωμένη συστάδα στο υψηλότερο επίπεδο στο σύστημά και ορίζει την συσχετισμένη συστάδα για κάθε σύνολο δεδομένων. Τεχνικές ιεραρχικής συσταδοποίησης μπορούν να υποδιαιρεθούν σε συσσωματωτικές(agglomerative) μεθόδους, οι οποίες προχωρούν με μια σειρά από διαδοχικές συγχωνεύσεις των n αντικειμένων σε συστάδες. Οι συσσωματωτικές μεθόδοι είναι ίσως οι πιο ευρέως χρησιμοποιούμενες από τις ιεραρχικές μεθόδους. Στις συσσωματωτικές μεθόδους, σε κάθε επανάληψη, οι δύο πιο κοντινές συστάδες συγχωνεύονται. Η απόσταση μεταξύ των συστάδων υπολογίζεται από τις μετρικές single-linkage και complete linkage. Αυτές οι δυο είναι οι πιο δημδημοφιλείς αλλα υπάρχουν και άλλες. Η single-linkage είναι η μικρότερη απόσταση μεταξύ ενός ζεύγους αντικειμένων σε δύο συστάδες. Μερικές φορές μπορεί να παράγει συστάδες όπου τα αντικείμενα σε διαφορετικές συστάδες είναι πιο κοντά μεταξύ τους παρά σε αντικείμενα εντός των δικών τους συστάδων. Αυτές οι συστάδες μπορεί να εμφανίζονται απλωμένες. Ο complete-linkage είναι όπου η απόσταση μεταξύ των συστάδων μετράται από την απόσταση μεταξύ του πιο απομακρυσμένου ζεύγους αντικειμένων στις δύο συστάδες. Αυτή η μέθοδος παράγει συνήθως πιο στενές συστάδες από της single-linkage, αλλά αυτές οι στενές συστάδες μπορεί να καταλήξουν πολύ κοντά μεταξύ τους. Άλλη μια κατηγορία ιεραρχικής συσταδοποίησης είναι οι διχαστικές(Divisive) μέθοδοι οι οποίες διαχωρίζουν τα n αντικείμενα διαδοχικά σε λεπτότερες συστάδες. Και οι δύο τύποι της ιεραρχικής συσταδοποίησης μπορούν να θεωρηθούν ως προσπάθεια εύρεσης του βέλτιστου βήματος [3].

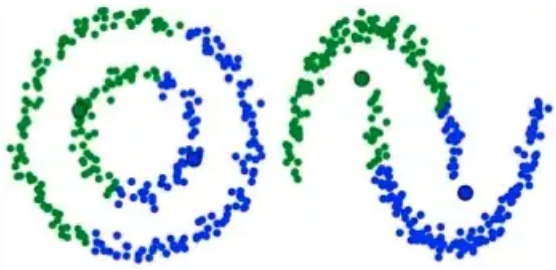
Distribution-based clustering.
Το μοντέλο συσταδοποίησης που σχετίζεται περισσότερο με τις στατιστικές βασίζεται σε μοντέλα διανομής(Distribution). Με μια προσέγγιση συσταδοποίησης που βασίζεται στη διανομή, όλα τα αντικείμενα θεωρούνται μέρη μιας συστάδας με βάση την πιθανότητα ότι ανήκουν σε μια δεδομένη συστάδα. Αν και η θεωρητική βάση αυτών των μεθόδων είναι εξαιρετική, υποφέρουν από ένα βασικό πρόβλημα που είναι γνωστό ως υπερπροσαρμογή(overfitting)(Σχήμα 1.11), εκτός εάν τεθούν περιορισμοί στην πολυπλοκότητα του μοντέλου. Ένα πιο σύνθετο μοντέλο θα είναι συνήθως σε θέση να εξηγήσει καλύτερα τα δεδομένα, γεγονός που καθιστά την επιλογή της κατάλληλης πολυπλοκότητας του μοντέλου εγγενώς δύσκολη.

Η πιο γνωστή μέθοδος σε αυτόν το τύπο αλγορίθμου είναι η Gaussian mixture [9]. Ένα από τα προβλήματα με το k-means είναι ότι τα δεδομένα πρέπει να ακολουθούν μια κυκλική μορφή. Ο τρόπος με τον οποίο ο k-means υπολογίζει την απόσταση μεταξύ των αντικειμένων έχει να κάνει με μια κυκλική διαδρομή, επομένως τα μη κυκλικά δεδομένα δεν ομαδοποιούνται σωστά ένα παράδειγμα φαίνεται στο σχήμα 1.12 [10]. Αυτό είναι ένα ζήτημα που επιδιορθώνουν το μοντέλο Gaussian mix. Δεν χρειαζόμαστε δεδομένα κυκλικού σχήματος για να λειτουργήσει καλά.Το μοντέλο μίξης Gauss χρησιμοποιεί πολλαπλές κατανομές Gauss για να χωρέσει δεδομένα αυθαίρετου σχήματος. Εδώ, το σύνολο δεδομένων μοντελοποιείται συνήθως με έναν σταθερό (για να αποφευχθεί η υπερβολική προσαρμογή) αριθμό κατανομών Gauss που αρχικοποιούνται τυχαία και των οποίων οι παράμετροι βελτιστοποιούνται επαναληπτικά για να ταιριάζουν καλύτερα στο σύνολο δεδομένων. Αυτό θα συγκλίνει σε ένα τοπικό βέλτιστο, επομένως πολλαπλές εκτελέσεις μπορεί να παράγουν διαφορετικά αποτελέσματα. Προκειμένου να επιτευχθεί μια σκληρή (hard) ομαδοποίηση, τα αντικείμενα συχνά αντιστοιχίζονται στην κατανομή Gauss στην οποία



Σχήμα 1.11: Παράδειγμα υπερπροσαρμογής (overfitting)

πιθανότατα ανήκουν, για soft ομαδοποιήσεις, αυτό δεν είναι απαραίτητο. Υπάρχουν πολλά μεμονωμένα μοντέλα Gaussian που λειτουργούν ως κρυφά στρώματα σε αυτό το υβριδικό μοντέλο. Έτσι το μοντέλο υπολογίζει την πιθανότητα ένα σημείο δεδομένων να ανήκει σε μια συγκεκριμένη κατανομή Gauss και αυτή είναι η συστάδα κάτω από την οποία θα εμπίπτει [9].



Σχήμα 1.12: Παράδειγμα αποτυχίας του k-means

1.4 Συσταδοποίηση:Περιπτώσεις χρήσεις

Το γενικό πρόβλημα που αντιμετωπίζει η συσταδοποίηση εμφανίζεται σε πολλούς κλάδους, σε αυτήν την ενότητα αναλύουμε μερικές από τις περιπτώσεις χρήσεις της.

Έρευνα αγοράς(Market research). Η διαίρεση των πελατών σε ομοιογενείς ομάδες είναι μία από τις βασικές στρατηγικές του μάρκετινγκ. Ένας ερευνητής αγοράς μπορεί, για παράδειγμα, να ρωτήσει πώς να ομαδοποιήσει τους καταναλωτές που αναζητούν παρόμοια οφέλη από ένα προϊόν, ώστε να μπορεί να επικοινωνεί μαζί τους καλύτερα.

Στις μηχανές αναζήτησης. Ο Παγκόσμιος Ιστός αποτελείται από δισεκατομμύρια ιστοσελίδες και τα αποτελέσματα ενός ερωτήματος σε μια μηχανή αναζήτησης μπορούν να επιστρέψουν χιλιάδες σελίδες. Η συσταδοποίηση μπορεί να χρησιμοποιηθεί για την ομαδοποίηση αυτών των αποτελεσμάτων αναζήτησης σε ένα μικρό αριθμό συστάδων, καθένα από τα οποία καταγράφει μια συγκεκριμένη πτυχή του ερωτήματος. Για παράδειγμα, ένα ερώτημα “ταινία” μπορεί να επιστρέψει ιστοσελίδες ομαδοποιημένες σε

κατηγορίες όπως κριτικές, τρέιλερ, αστέρια και θέατρα. Κάθε κατηγορία (cluster) μπορεί να χωριστεί σε υποκατηγορίες (subclusters), δημιουργώντας μια ιεραρχική δομή που βοηθά περαιτέρω την εξερεύνηση των αποτελεσμάτων του ερωτήματος από τον χρήστη.

Συστήματα συστάσεων (Recommender systems). Τα συστήματα συστάσεων έχουν σχεδιαστεί για να προτείνουν νέα προϊόντα με βάση τα γούστα ενός χρήστη. Μερικές φορές χρησιμοποιούν αλγόριθμους συσταδοποίησης για να προβλέψουν τις προτιμήσεις ενός χρήστη με βάση τις προτιμήσεις άλλων χρηστών στην συστάδα του χρήστη.

Ψυχολογία και Ιατρική. Μια ασθένεια ή κατάσταση έχει συχνά μια σειρά από παραλλαγές και συσταδοποίηση μπορεί να χρησιμοποιηθεί για τον προσδιορισμό αυτών των διαφορετικών υποκατηγοριών. Για παράδειγμα, η συσταδοποίηση έχει χρησιμοποιηθεί για τον εντοπισμό διαφορετικών τύπων κατάθλιψης.

Οι αλγόριθμοι συσταδοποίησης χρησιμοποιούνται ευρέως για την αναγνώριση καρκινικών κυττάρων. Χωρίζει τα καρκινικά και μη καρκινικά σύνολα δεδομένων σε διαφορετικές ομάδες.

Βιολογία. Στον τομέα της βιολογίας, μπορεί να χρησιμοποιηθεί για την εξαγωγή ταξινομιών φυτών και ζώων, την κατηγοριοποίηση γονιδίων με παρόμοιες λειτουργικότητες και την απόκτηση γνώσεων σχετικά με τις δομές που είναι εγγενείς στους πληθυσμούς. Πιο πρόσφατα, οι βιολόγοι έχουν εφαρμόσει συσταδοποίηση για να αναλύσουν τις μεγάλες ποσότητες γενετικών πληροφοριών που είναι τώρα διαθέσιμες. Για παράδειγμα, η συσταδοποίηση έχει χρησιμοποιηθεί για την εύρεση ομάδων γονιδίων που έχουν παρόμοιες λειτουργίες.

Για το κλίμα της γής. Η κατανόηση του κλίματος της Γης απαιτεί την εύρεση προτύπων στην ατμόσφαιρα και στον ωκεανό. Για το σκοπό αυτό, έγινε ανάλυση συστάδων εφαρμόζεται για την εύρεση μοτίβων στην ατμοσφαιρική πίεση των πολικών περιοχών και περιοχών του ωκεανού που έχουν σημαντικό αντίκτυπο στο χερσαίο κλίμα

Επιστήμη των υπολογιστών (Computer Science). Η συσταδοποίηση είναι χρήσιμη στην εξέλιξη του λογισμικού, καθώς συμβάλλει στη μείωση των ιδιοτήτων παλαιού τύπου στον κώδικα, αναμορφώνοντας τη λειτουργικότητα που έχει καταργηθεί. Είναι μια μορφή αναδιάρθρωσης και ως εκ τούτου είναι ένας τρόπος άμεσης προληπτικής συντήρησης.

Η συσταδοποίηση μπορεί να χρησιμοποιηθεί για τον εντοπισμό διαφορετικών θέσεων εντός του πληθυσμού ενός εξελικτικού αλγορίθμου, έτσι ώστε οι ευκαιρίες αναπαραγωγής να μπορούν να κατανεμηθούν πιο ομοιόμορφα μεταξύ των εξελισσόμενων ειδών ή υποειδών. [3]

1.5 Κίνητρο και Συνεισφορά

Από τις παραπάνω ενότητες καταλαβαίνουμε οτι η συσταδοποίηση είναι αρκετά χρήσιμη και σημαντική στον τομέα της αναλυτικής των δεδομένων και της μηχανικής μάθησης. Πολλές ερευνητικές ομάδες εργάζονται στον τομέα Cluster analysis. Οι παραλλαγές του ή άλλοι αλγόριθμοι και η βιβλιογραφία για αυτό το θέμα είναι πλούσια. Μετά από ερεύνα εντοπίσαμε οτι δεν υπάρχουν πολλές διαδικτυακές εφαρμογές που είναι ανοιχτές στην παγκόσμια επιστημονική κοινότητα, και οι υπάρχουσες έχουν αρκετά περιορισμένες δυνατότητες ή ακόμα προτρέπουν τους χρήστες να πληρώσουν συνδρομή ώστε να “ξε-

κλειδώσουν" περισσότερες λειτουργίες. Βέβαια υπάρχουν και κάποιες λύσεις που δεν είναι εφαρμογές ιστού αλλά desktop εφαρμογές, δηλαδή ο χρήστης πρέπει να κατεβάσει τοπικά στον υπολογιστή του την εφαρμογή και να την τρέξει, κάποιες από αυτές είναι ευρέως γνωστές όπως το WEKA και το MATLAB. Το Weka αναπτύχθηκε εσωτερικά στο Πανεπιστήμιο του Waikato για ερευνητικούς σκοπούς. Το Weka με το GUI του παρέχει την ευκολότερη μετάβαση στον κόσμο της επιστήμης των δεδομένων. Όντας γραμμένο σε Java, όσοι έχουν εμπειρία με Java μπορούν επίσης να καλέσουν τη βιβλιοθήκη στον κώδικά τους. Το Weka είναι μια συλλογή αλγορίθμων μηχανικής μάθησης για εργασίες εξόρυξης δεδομένων. Οι αλγόριθμοι μπορούν είτε να εφαρμοστούν απευθείας σε ένα σύνολο δεδομένων είτε να κληθούν από τον δικό μας κώδικα Java. Το Weka περιέχει εργαλεία για προεπεξεργασία δεδομένων, ταξινόμηση, παλινδρόμηση, συσταδοποίηση,κανόνες συσχέτισης και οπτικοποίηση [11], ωστόσο όπως αναφέρετε και παραπάνω το μειονέκτημα είναι οτι ο χρήστης πρέπει να το κατεβάσει τοπικά στο υπολογιστή του. Το MATLAB είναι μια γλώσσα προγραμματισμού που χρησιμοποιεί υπολογισμούς και αλγόριθμους για να αναλύει μεγάλο όγκο δεδομένων και να τα παρουσιάζει σε οπτικά ελκυστικές μορφές. Ορισμένα χαρακτηριστικά του MATLAB περιλαμβάνουν:

- Υπολογισμός αριθμητικών δεδομένων
- Δημιουργία γραφικών για επιστημονική χρήση
- Μοντελοποίηση και προσομοίωση δεδομένων
- Ανάλυση Δεδομένων

Το περιβάλλον του MATLAB παρουσιάζει το πρόγραμμα και τα διαθέσιμα εργαλεία. Το περιβάλλον MATLAB μάς επιτρέπει να εκτελούμε εντολές, να διαχειριζόμαστε τα αρχεία μας, ακόμη και να αναλύουμε δεδομένα. Έχει ένα παράθυρο εντολών όπου μπορούμε να γράψουμε απλές εντολές [12].Ωστόσο και πάλι στα αρνητικά είναι οτι ο χρήστης πρέπει να κατεβάσει το πρόγραμμα τοπικά,επίσης ο χρήστης που θα το χρησιμοποίηση θα πρέπει να έχει βασικές γνώσει προγραμματισμού. Και τέλος, αξίζει να αναφέρουμε τίς προσπάθειες κυρίως μεμονωμένων προγραμματιστών να εφαρμόσουν προγράμματα ή σενάρια σε ποικιλία γλωσσών προγραμματισμού που σχετίζονται με την συσταδοποίηση και να το μοιραστούν με την κοινότητα.Και πάλι στα αρνητικά είναι οτι δεν υπάρχει διεπαφή χρήστη,επίσης οι χρήστες θα πρέπει να χρησιμοποιήσουν συνδυασμό αυτών τών προγραμμάτων/σεναρίων για να πάρουν τα σωστά αποτελέσματα.

Στην συγκεκριμένη πτυχιακή θα χρησιμοποιήσουμε τον αλγόριθμο συσταδοποίησης κ-μέσων (k-means) όπου ανήκει στην κατηγορία των αλγορίθμων κατάτμησης (partitioning). Βασικό μειονέκτημα είναι ότι ο χρήστης πρέπει να καθορίσει το πλήθος k των συστάδων που θα ανακαλύψει ο αλγόριθμος. Συχνά όμως, ο χρήστης δεν μπορεί να προκαταβάλει τον συγκεκριμένο αριθμό και αν ο αλγόριθμος εκτελεστεί με διαφορετική τιμή k, θα δημιουργηθούν εντελώς διαφορετικές συστάδες. Ένας τρόπος για τον προσδιορισμό της παραμέτρου είναι η μέθοδος του αγκώνα (Elbow). Από τις παραπάνω εφαρμογές και προγράμματα που αναφέρθηκαν δεν υπάρχει κάποια εφαρμογή/πρόγραμμα οπού να τρέχει την μέθοδο του αγκώνα, να προτείνει τιμή κ στον χρήστη καθώς και να κάνει την συσταδοποίηση, ο χρήστης θα πρέπει να τρέξει συνδυασμό εφαρμογών ή προγραμμάτων.

Στόχος της πτυχιακής εργασίας είναι η ανάπτυξη μιας web εφαρμογής όπου ο κάθε χρήστης θα μπορεί

να ανεβάζει σύνολα δεδομένων και η εφαρμογή θα κατασκευάζει το γράφημα όπου θα παρουσιάζεται ο αγκώνας και θα προτείνει στον χρήστη πιθανή τιμή για την παράμετρο κ. Στη συνέχεια, ο χρήστης θα μπορεί να κάνει την συσταδοποίηση με το προτεινόμενο κ, καθώς και να κατεβάσει το γράφημα ή το σύνολο δεδομένων όπου θα αναφέρετε η συστάδα που ανατέθηκε σε κάθε αντικείμενο του συνόλου δεδομένων.

1.6 Οργάνωση της εργασίας

Η εργασία, αποτελείται από 7 κεφάλαια. Έως τώρα έχει γίνει μια σύντομη αναφορά στο τι θα ακολουθήσει, καθώς επίσης έχει γίνει μια εισαγωγή σχετικά με την συσταδοποίηση και τους αλγορίθμους της.

Στο δεύτερο κεφάλαιο θα γίνει λεπτομερής περιγραφή του αλγορίθμου κ-μέσων καθώς είναι ο κύριος αλγόριθμος που χρησιμοποιούμε στην εφαρμογή. Θα αναλυθούν τα πλεονεκτήματα και τα μειονεκτήματα του αλγορίθμου, πως μπορούμε να βρούμε την παράμετρο κ,θα αναλυθεί η μέθοδος του αγκώνα, καθώς και παραλλαγές του αλγορίθμου.

Στο τρίτο κεφάλαιο θα αναλυθούν οι τεχνολογίες που χρησιμοποιήθηκαν για την ανάπτυξη της εφαρμογής. Το κεφάλαιο χωρίζεται σε δύο μέρη, το πρώτο αφορά τις τεχνολογίες του back-end και το δεύτερο τις τεχνολογίες του front-end.

Στο τέταρτο κεφάλαιο θα αναλυθεί η σχεδίαση και υλοποίηση της εφαρμογής. Θα παρουσιαστούν οι λειτουργικές απαιτήσεις της εφαρμογής, θα γίνει περιγραφή της αρχιτεκτονικής με διαγράμματα, θα αναλυθούν οι τύποι των χρηστών. Στην συνέχεια θα αναλυθεί το back-end και το front-end παρουσιάζοντας κάποια κομμάτια κώδικα, ώστε να καταλάβει ο αναγνώστης την υλοποίηση της εφαρμογής.

Στο πέμπτο κεφάλαιο θα γίνει παρουσίαση του REST API που αναπτύχθηκε για την εφαρμογή. Θα παρουσιαστούν όλα τα endpoints μαζί με παραδείγματα κλήσης και απόκρισης από τον διακομιστή.

Στο έκτο κεφάλαιο θα γίνει παρουσίαση της ιστοσελίδας της εφαρμογής με οδηγίες χρήσης της.

Στο έβδομο και τελευταίο κεφάλαιο θα αναφερθούν κάποια συμπεράσματα και θα αναλυθούν μελλοντικές επεκτάσεις της εφαρμογής.

Κεφάλαιο 2ο: Συσταδοποίηση k-means

2.1 Ο αλγόριθμος κ-μέσων

Η συσταδοποίηση κ-μεσων (k-means) είναι μια μέθοδος κβαντοποίησης διανυσμάτων, αρχικά από την επεξεργασία σήματος, που στοχεύει να χωρίσει n παρατηρήσεις σε k συστάδες στις οποίες κάθε παρατήρηση ανήκει στη συστάδα με τον πλησιέστερο μέσο όρο (κέντρα συστάδων ή cluster centroid), που χρησιμεύει ως πρωτότυπο στην συστάδα. Ένα κεντρο (centroid) είναι ένα στιγμιότυπο δεδομένων που αντιπροσωπεύει το κέντρο της συστάδας (το μέσο όρο) και μπορεί να μην είναι απαραίτητα μέλος του συνόλου δεδομένων. Με αυτόν τον τρόπο, ο αλγόριθμος λειτουργεί μέσω μιας επαναληπτικής διαδικασίας έως ότου κάθε στιγμιότυπο είναι πιο κοντά στο κέντρο της συστάδας του από ό,τι στο κέντρο των άλλων συστάδων, ελαχιστοποιώντας την απόσταση εντός της συστάδας σε κάθε βήμα.

Ο όρος “k-means” χρησιμοποιήθηκε για πρώτη φορά από τον James MacQueen το 1967 αν και η ιδέα ανάγεται στον Hugo Steinhaus το 1956. Ο τυπικός αλγόριθμος προτάθηκε για πρώτη φορά από τον Stuart Lloyd των Bell Labs το 1957 ως τεχνική για τη διαμόρφωση του κώδικα παλμού, αν και δεν δημοσιεύτηκε ως άρθρο σε περιοδικό μέχρι το 1982. Το 1965, ο Edward W. Forgy δημοσίευσε ουσιαστικά την ίδια μέθοδο, γι’ αυτό και μερικές φορές αναφέρεται ως αλγόριθμος Lloyd–Forgy.

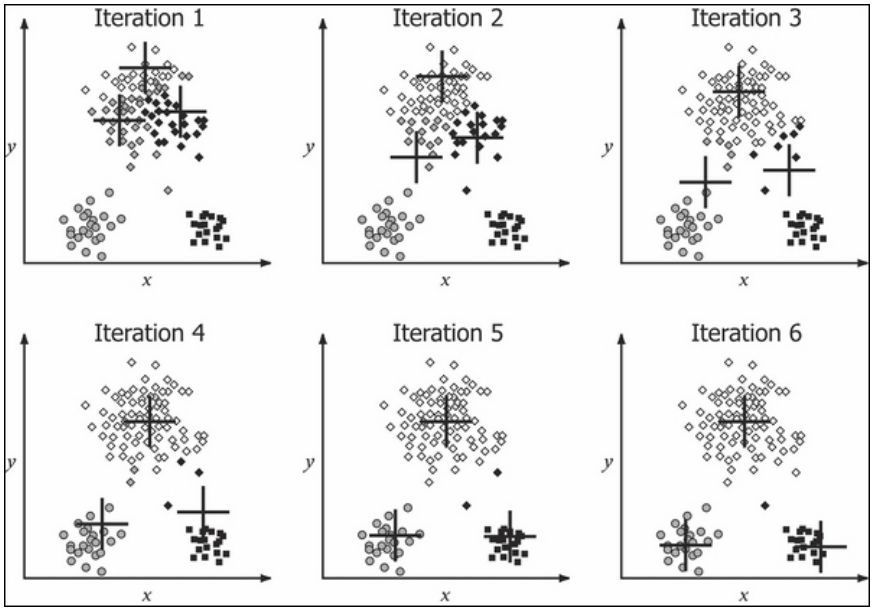
Ο αλγόριθμος κ-μεσων είναι απλός στην υλοποίηση του. Για αρχή επιλέγουμε Κ αρχικά κέντρα, όπου το Κ είναι μια παράμετρος που καθορίζεται από τον χρήστη, δηλαδή ο αριθμός των επιθυμητών συστάδων, για παράδειγμα, ορίζοντας το κ ίσο με 2, το σύνολο δεδομένων θα ομαδοποιηθεί σε δύο συστάδες, ενώ εάν ορίσουμε το κ ίσο με τέσσερα θα ομαδοποιήσουμε τα δεδομένα σε 4 συστάδες.

Στη συνέχεια, κάθε στιγμιότυπο εκχωρείται στο πλησιέστερο κέντρο και κάθε συλλογή στιγμιοτύπων που εκχωρείται σε ένα κέντρο είναι συστάδα. Στη συνέχεια, το κέντρο κάθε συστάδας ενημερώνεται με βάση τα στιγμιότυπα που έχουν εκχωρηθεί στην συστάδα. Επαναλαμβάνουμε τα βήματα ανάθεσης και ενημέρωσης έως ότου κανένα στιγμιότυπο δεν αλλάξει συστάδα, ή ισοδύναμα, έως ότου τα κέντρα παραμένουν τα ίδια.

Algorithm 2 Ο βασικός αλγόριθμος των κ-μέσων

1: Επιλογή του κ για τα αρχικά κέντρα
2: **repeat**
3: Σχηματίστε κ συστάδες αναθέτοντας κάθε στιγμιότυπο στο πλησιέστερο κέντρο του.
4: Υπολογίστε ξανά το κέντρο κάθε συστάδας
5: **until** Τα κέντρα να μην αλλάξουν

Ας δούμε ένα παράδειγμα για να κατανοήσουμε καλύτερα τον αλγόριθμο, στο σχήμα 2.1 τρέχουμε τον αλγόριθμο για την εύρεση τριών συστάδων δηλαδή κ=3, στο πρώτο βήμα τα κέντρα αρχικοποιούνται τυχαία μέσα στο σύνολο δεδομένων. Στο δεύτερο βήμα, τα στιγμιότυπα εκχωρούνται στα ενημερωμένα κέντρα και τα κέντρα ενημερώνονται ξανά. Στα βήματα 2, 3, 4 και 5 δύο από τα κέντρα κινούνται στις δύο μικρές ομάδες στιγμιοτύπων στο κάτω μέρος των σχημάτων. Όταν ο αλγόριθμος κ-μέσων τερματίζεται στο βήμα 6 επειδή δεν συμβαίνουν άλλες αλλαγές, τα κέντρα έχουν αναγνωρίσει τις φυσικές συστάδες τους. Για ορισμένους συνδυασμούς συναρτήσεων εγγύτητας και τύπων κεντροειδών, ο αλγόριθμος κ-μέσων συγκλίνει πάντα σε μια λύση, δηλαδή φτάνει σε μια κατάσταση στην οποία κανένα στιγμιότυπο δεν μετατοπίζεται από το συστάδα σε μία άλλη, και ως εκ τούτου, τα κέντρα δεν αλλάζουν.



Σχήμα 2.1: Χρησιμοποιώντας τον αλγόριθμο K-means για την εύρεση τριών συστάδων

Εκχώρηση σημείων στο πλησιέστερο κέντρο Για να αντιστοιχίσουμε ένα σημείο στο πλησιέστερο κέντρο, χρειαζόμαστε ένα μέτρο εγγύτητας που ποσοτικοποιεί την έννοια του πλησιέστερου για τα συγκεκριμένα δεδομένα που εξετάζουμε. Η Ευκλείδεια απόσταση χρησιμοποιείται συχνά για στιγμιότυπα στον Ευκλείδειο χώρο, ενώ η ομοιότητα συνημιτόνου είναι πιο κατάλληλη για έγγραφα. Ωστόσο, μπορεί να υπάρχουν διάφοροι τύποι μέτρων εγγύτητας που είναι κατάλληλα για έναν συγκεκριμένο τύπο δεδομένων. Για παράδειγμα, η απόσταση Μανχάταν μπορεί να χρησιμοποιηθεί για Ευκλείδεια δεδομένα, ενώ το μέτρο Jaccard χρησιμοποιείται συχνά για έγγραφα. Συνήθως, τα μέτρα ομοιότητας που χρησιμοποιούνται για το αλγόριθμο είναι σχετικά απλά αφού υπολογίζει επανειλημμένα την ομοιότητα κάθε στιγμιοτύπου με κάθε κέντρο. Σε ορισμένες περιπτώσεις, ωστόσο, όπως όταν τα δεδομένα βρίσκονται σε ευκλείδειο χώρο χαμηλής διάστασης, είναι δυνατό να αποφευχθεί ο υπολογισμός πολλών από τις ομοιότητες, επιταχύνοντας έτσι σημαντικά τον αλγόριθμο κ-μέσων.

Για να μετρήσουμε την ποιότητα μίας συσταδοποίησης χρησιμοποιούμε το άθροισμα του τετραγωνικού σφάλματος ή αλλιώς sum of the squared error(SSE), το οποίο είναι επίσης γνωστό ως scatter. Με άλλα λόγια, υπολογίζουμε το σφάλμα κάθε στιγμιοτύπου, δηλαδή την Ευκλείδεια απόστασή του από το πλησιέστερο κέντρο, και στη συνέχεια υπολογίζουμε το συνολικό άθροισμα των τετραγωνικών σφαλμάτων. Λαμβάνοντας υπόψη δύο διαφορετικά σύνολα συστάδων που παράγονται από δύο διαφορετικά τρεξίματα του αλγορίθμου, προτιμάμε αυτό με το μικρότερο τετραγωνικό σφάλμα, καθώς αυτό σημαίνει

ότι τα κέντρα αυτής της συσταδοποίησης αναπαριστούν καλύτερα τα στιγμιότυπα της συστάδας τους. Χρησιμοποιώντας τη σημειογραφία το SSE ορίζεται επίσημα ως εξής:

$$SSE = \sum_{j=1}^{K} \sum_{i=1}^{n} dist(C_i, x)^2 \tag{2.1}$$

Όπου x είναι ένα στιγμιότυπο, το κέντρο(c) είναι το κέντρο της συστάδας στο οποίο έχει εκχωρηθεί το x και το dist(κέντρο(c),x) είναι η Ευκλείδεια απόσταση μεταξύ του x και του κέντρου(c). Ο αλγόριθμος στοχεύει στην ελαχιστοποίηση του αθροίσματος των τετραγωνικών αποστάσεων (SSE) μεταξύ κάθε στιγμιοτύπου και του πλησιέστερου κέντρου, γνωστό και ως άθροισμα τετραγώνων σφαλμάτων εντός συστάδας ή within-cluster sum of squared errors(WCSS). Ο αλγόριθμος κ-μέσων θα συγκλίνει σε ένα τοπικό ελάχιστο του SSE, το οποίο δεν είναι εγγυημένο ότι είναι το συνολικό ελάχιστο. Αυτός είναι ένας από τους λόγους για τους οποίους ο αλγόριθμος είναι ευαίσθητος στην αρχική επιλογή κέντρων και για αυτό είναι χρήσιμο να εκτελείται πολλές φορές με διαφορετικά αρχικά κέντρα για να βρεθεί η καλύτερη λύση. [13]

Ας δούμε κάποια παραδείγματα με τα βήματα και τις αριθμητικές πράξεις που πρέπει να εκτελέσει ο αλγόριθμος ώστε να τον κατανοήσουμε καλύτερα.

Παράδειγμα 1
Έστω ότι μας δίνετε ένα μονοδιάστατο αρχείο δεδομένων με τις αριθμητικές τιμές : **10, 15, 20, 28, 30, 48, 62**

Τα βήματα είναι ως εξής:

1. Αποφασίζουμε να τα χωρίσουμε σε 2 συστάδες (k)
2. ο αλγόριθμος ορίζει τυχαία ως αρχικά κέντρα τις τιμές 15 και 40
3. Ο αλγόριθμος αναθέτει κάθε στιγμιότυπο στο πλησιέστερο κέντρο συστάδας με βάση την απόσταση. Οπότε οι τιμές **10, 15, 20** ανατίθενται στην συστάδα 1 με κέντρο το 15 και οι τιμές **28, 30, 48, 62** στην συστάδα 2 και κέντρο το 40
4. Υπολογίζει ξανά τα κέντρα συστάδας ως τον μέσο όρο των στιγμιοτύπων που έχουν εκχωρηθεί σε κάθε συστάδα. Για την συστάδα 1, μέσος όρος είναι το 15 ενώ για την συστάδα 2 είναι το 42, αυτές οι τιμές είναι τα νέα κέντρα.
5. Επαναλαμβάνει τα βήματα 3 και 4 μέχρι να μην αλλάξουν τα κέντρα των συστάδων. Οπότε για την συστάδα 1 με κέντρο 15 αναθέτει τις τιμές **10, 15, 20, 28** ενώ για την συστάδα 2 με κέντρο 42 τις τιμές **30, 48, 62**.

   Επαναλαμβάνουμε το βήμα 4 και βρίσκουμε καινούργια κέντρα τις τιμές 18.25 για την συστάδα 1 και 46.67 για την συστάδα 2. Οπότε για το βήμα 3 αναθέτει τις τιμές **10, 15, 20, 28, 30** στην συστάδα 1 και τις τιμές **48, 62** στην συστάδα 2.

   Επαναλαμβάνουμε το βήμα 4 και βρίσκουμε καινούργια κέντρα τις τιμές 20.6 για την συστάδα 1 και 55 για την συστάδα 2. Οπότε για το βήμα 3 αναθέτει τις τιμές **10, 15, 20, 28, 30** στην συστάδα 1 και τις τιμές **48, 62** στην συστάδα 2.

Ο K-Means σταματάει επειδή δεν έγινε καμία μετακίνηση από συστάδα σε συστάδα.

Το SSE υπολογίζεται ως εξής:
SSE Συστάδας 1: (10-20.6)^2 + (15-20.6)^2 + (20-20.6)^2 + (28-20.6)^2 + (30-20.6)^2 = 287.2

SSE Συστάδας 2: (48 - 55)^2 + (62 - 55) ^2 = 98

SSE= 287.2 + 98 = 385.2

Παράδειγμα 2
Έστω ότι έχουμε ένα σύνολο δεδομένων δύο διαστάσεων με τις τιμές:
A1: (2,10), A2: (2,5), A3: (8,4),
B1: (5,8), B2: (7,5), B3: (6,4),
C1: (1,2), C2: (4,9)

Θέλουμε να τα χωρίσουμε σε 3 συστάδες με αρχικά κέντρα Σ1 (2, 10), Σ2 (7, 5) και Σ3 (1, 2)

Γενικά ο αλγόριθμος ακολουθεί τα ίδια βήματα που είδαμε στο παράδειγμα 1. Η διαφορά είναι ότι για να υπολογίσουμε την απόσταση μεταξύ του στιγμιοτύπου και του κάθε κέντρου χρησιμοποιούμε την Ευκλείδεια απόσταση, η οποία ορίζεται ως εξής:

$$d(p,q) = \sqrt{\sum_{i=1}^{n}(q_i - p_i)^2} \tag{2.2}$$

Στον πίνακα 2.1 υπολογίζουμε τις αποστάσεις κάθε στιγμιοτύπου με τα κέντρα κάθε συστάδας. Στην συνέχεια αναθέτουμε το στιγμιότυπο στην συστάδα με την μικρότερη απόσταση.

Πίνακας 2.1: Υπολογισμός των αποστάσεων

Object	x	y	Dist Σ1	Dist Σ2	Dist Σ3	Cluster
A1	2	10	0	7.0711	8.0622	Σ1
A2	2	5	5	5	3.1623	Σ3
A3	8	4	8.4853	1.4142	7.2801	Σ2
B1	5	8	3.6056	3.6056	7.2111	Σ1
B2	7	5	7.0711	0	6.7082	Σ2
B3	6	4	7.2111	1.4142	5.3852	Σ2
C1	1	2	8.0622	6.7082	0	Σ3
C2	4	9	2.2361	5	7.6158	Σ1

Υπολογίζουμε τον μέσο ορο με βάση το x και το y κάθε συστάδας και παίρνουμε τις τιμές Σ1 (3.6667, 9), Σ2 (7, 4.3333) και Σ3 (1.5, 3.5).

Υπολογίζουμε ξανά τις αποστάσεις όπως φαίνεται στο πίνακα 2.2 και εφόσον δεν εχει γίνει καμία μετακίνηση από συστάδα σε συστάδα ο αλγόριθμος σταματάει.

Πίνακας 2.2: Επανυπολογισμός των αποστάσεων

Object	x	y	Dist Σ1	Dist Σ2	Dist Σ3	Cluster
A1	2	10	1.9437	7.5572	6.5192	Σ1
A2	2	5	4.3333	5.0442	1.5811	Σ3
A3	8	4	6.6165	1.0541	6.5192	Σ2
B1	5	8	1.6667	4.1767	5.7009	Σ1
B2	7	5	5.2068	0.6667	5.7009	Σ2
B3	6	4	5.5176	1.0541	4.5277	Σ2
C1	1	2	7.4907	6.4377	1.5811	Σ3
C2	4	9	0.3333	5.5478	6.0415	Σ1

2.2 Πλεονεκτήματα και Μειονεκτήματα

Σε αυτήν την ενότητα αξιολογούμε τα πλεονεκτήματα και τα μειονεκτήματα του αλγόριθμου κ-μέσων ώστε να ”ζυγίσουμε” τα οφέλη και τα ελαττώματα από τη χρήση αυτής της τεχνικής συσταδοποίησης.

Πλεονεκτήματα

Είναι εύκολο να κατανοηθεί και να εφαρμοστεί: Ο K-means είναι ένας απλός αλγόριθμος που είναι εύκολο να κατανοηθεί και να εφαρμοστεί ακόμη και για όσους δεν έχουν ισχυρό μαθηματικό υπόβαθρο. Περιλαμβάνει μόνο μερικά βασικά βήματα, όπως αρχικοποίηση κέντρων, επαναληπτική εκχώρηση στιγμιοτύπων σε συστάδες με βάση την εγγύτητά τους με τα κέντρα και ενημέρωση των κέντρων με βάση τις νέες εκχωρήσεις της συστάδας.

Είναι υπολογιστικά αποδοτικό: Ο K-means είναι υπολογιστικά αποδοτικός, ειδικά όταν υλοποιείται χρησιμοποιώντας έναν κλιμακούμενο(scalable) αλγόριθμο όπως ο αλγόριθμος του Lloyd. Η χρονική πολυπλοκότητα του αλγορίθμου είναι O(nkI*d), όπου n είναι ο αριθμός των σημείων δεδομένων, k είναι ο αριθμός των συστάδων, I είναι ο αριθμός των επαναλήψεων και d ο αριθμός των χαρακτηριστικών. Τα δύο πρώτα βήματα του αλγορίθμου, που εκχωρούν κάθε στιγμιότυπο στο πλησιέστερο κέντρο και υπολογίζουν εκ νέου το κέντρο για κάθε συστάδα, και τα δύο έχουν γραμμική χρονική πολυπλοκότητα O(nk) σε σχέση με τον αριθμό των σημείων δεδομένων. Αυτό είναι καλό για μεγάλα σύνολα δεδομένων.

Μπορεί να χρησιμοποιηθεί ως βήμα προεπεξεργασίας για άλλους αλγόριθμους: Ο K-means μπορεί να χρησιμοποιηθεί ως βήμα προεπεξεργασίας για άλλους αλγόριθμους μηχανικής μάθησης, όπως τα νευρωνικά δίκτυα και τα δέντρα αποφάσεων. Ομαδοποιώντας τα δεδομένα, ο αλγόριθμος μπορεί να βοηθήσει στον εντοπισμό υποκείμενων μοτίβων και δομής στα δεδομένα, τα οποία μπορούν να βοηθήσουν στη βελτίωση της απόδοσης του επόμενου αλγορίθμου.

Μπορεί να χρησιμοποιηθεί για συμπίεση δεδομένων: Ο K-means μπορεί να χρησιμοποιηθεί για συμπίεση δεδομένων μειώνοντας τον αριθμό των bit που απαιτούνται για την αναπαράσταση των δεδομένων. Αυτό γίνεται αντικαθιστώντας κάθε σημείο δεδομένων με το αντίστοιχο κέντρο συστάδων.

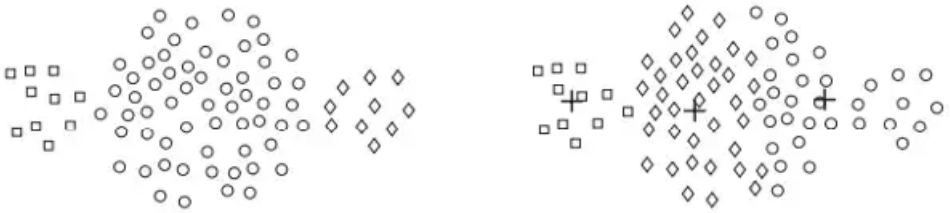
Μπορεί να αναγνωρίσει κρυφά μοτίβα και δομή στα δεδομένα: Ο K-means μπορεί να αναγνωρίσει μοτίβα και δομή στα δεδομένα που μπορεί να μην είναι άμεσα εμφανή. Αυτό μπορεί να είναι χρήσιμο

για εφαρμογές όπως η αναγνώριση εικόνας και ομιλίας, η επεξεργασία φυσικής γλώσσας και η τμηματοποίηση πελατών.

Κατάλληλος για μεγάλα σύνολα δεδομένων: Ο K-means είναι κατάλληλος για μεγάλα σε αριθμό σύνολα δεδομένων και υπολογίζεται πολύ πιο γρήγορα από ένα μικρότερο σύνολο δεδομένων. Μπορεί επίσης να παράγει υψηλότερες σε αριθμό συστάδες. Ένας άλλος λόγος για τον οποίο ο K-means είναι καλός σε μεγάλα σύνολα δεδομένων είναι ότι μπορεί να χειριστεί μεγάλες ποσότητες δεδομένων χωρίς να κάνει overfitting ή να χάσει την ικανότητά του να γενικεύει σε νέα δεδομένα. Αυτό συμβαίνει επειδή ο αλγόριθμος k-means δεν κάνει υποθέσεις σχετικά με την υποκείμενη κατανομή πιθανοτήτων των δεδομένων και δεν επηρεάζεται από την παρουσία θορύβου(noise) ή ακραίων τιμών στα δεδομένα.

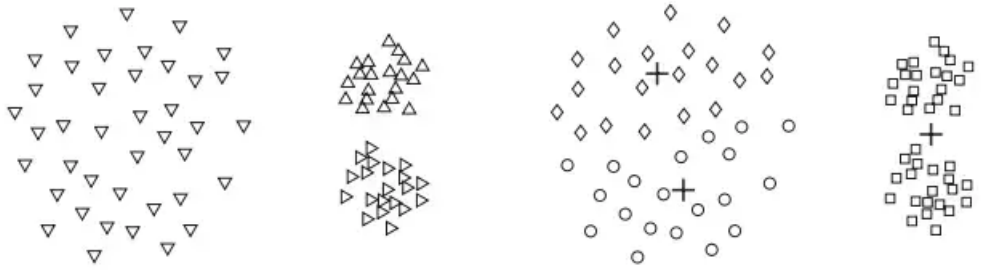
Μειονεκτήματα

Υποθέτει ότι οι συστάδες έχουν παρόμοιο μέγεθος και παρόμοια πυκνότητα: Ο αλγόριθμος k-means υποθέτει ότι οι συστάδες έχουν παρόμοιο μέγεθος και παρόμοια πυκνότητα. Εάν οι συστάδες έχουν πολύ διαφορετικά μεγέθη(Σχήμα 2.2) ή πυκνότητες(Σχήμα 2.3), ο k-means μπορεί να μην είναι σε θέση να αναγνωρίσει την πραγματική δομή των δεδομένων ή μπορεί να οδηγήσει σε κακή κατάτμηση.



(a) Original points. (b) Three K-means clusters.

Σχήμα 2.2: Εκτέλεση του k-means σε σύνολο δεδομένων με διαφορετικά μεγέθη



(a) Original points. (b) Three K-means clusters.

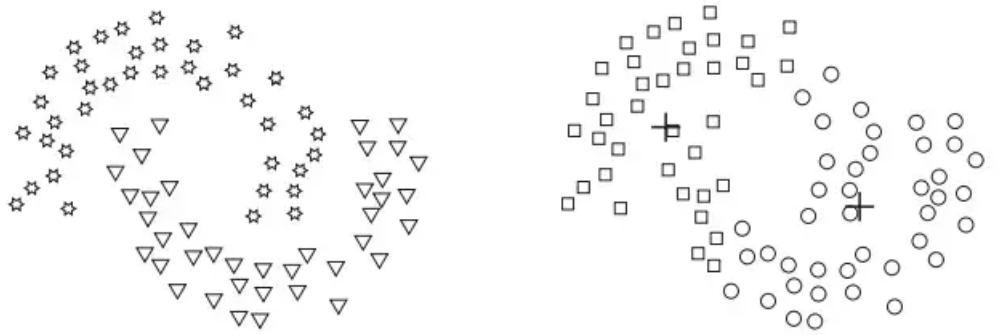
Σχήμα 2.3: Εκτέλεση του k-means σε σύνολο δεδομένων με διαφορετική πυκνότητα

Είναι ευαίσθητο στα αρχικά κέντρα: Ο αλγόριθμος k-means είναι ευαίσθητος στα αρχικά κέντρα, επομένως μια καλή πρακτική είναι να τον εκτελούμαι πολλές φορές με διαφορετικά αρχικά κέντρα και στην συνέχεια να επιλέξουμε την καλύτερη λύση. Ωστόσο, αυτό μπορεί να είναι υπολογιστικά ακριβό, ειδικά για μεγάλα σύνολα δεδομένων.

Δεν είναι κατάλληλος για κατηγορικά δεδομένα: Ο K-means είναι ένας αλγόριθμος που βασίζεται στην απόσταση και δεν είναι κατάλληλος για κατηγορικά δεδομένα αλλά για αριθμητικά. Για κατηγο-

ρικά δεδομένα, θα πρέπει να χρησιμοποιούνται άλλοι αλγόριθμοι συσταδοποίησης, όπως η ιεραρχική συσταδοποίηση ή η συσταδοποίηση με βάση την πυκνότητα.

Δεν είναι κατάλληλο για ομαδοποίηση μη σφαιρικών συστάδων: Ο αλγόριθμος K-means δεν είναι κατάλληλος για τη ομαδοποίηση μη σφαιρικών συστάδων, δηλαδή συστάδων που έχουν διαφορετικό σχήμα από μια σφαίρα, όπως επιμήκεις συστάδες ή συστάδες με πολλαπλά κέντρα πυκνότητας. Άλλοι αλγόριθμοι όπως η ομαδοποίηση με βάση την πυκνότητα είναι πιο κατάλληλοι σε αυτήν την περίπτωση.



(a) Original points. (b) Two K-means clusters.

Σχήμα 2.4: Εκτέλεση του k-means σε σύνολο δεδομένων μη-σφαιρικό

Ο αριθμός των συστάδων δεν είναι γνωστός εκ των προτέρων: Ο αλγόριθμος k-means υποθέτει ότι ο αριθμός των συστάδων είναι γνωστός εκ των προτέρων, κάτι που δεν συμβαίνει πάντα. Στην πράξη, ο αριθμός των συστάδων συχνά δεν είναι γνωστός και πρέπει να προσδιοριστεί χρησιμοποιώντας άλλες μεθόδους, όπως η μέθοδος του αγκώνα,το silhouette score ή η στατιστική του χάσματος (gap statistic).

Μη ανθεκτικός στον θόρυβο και σε ακραίες τιμές: Ο αλγόριθμος k-means δεν είναι ανθεκτικός στον θόρυβο και σε ακραίες τιμές στα δεδομένα, μπορεί να επηρεαστεί από έναν μικρό αριθμό στιγμιοτύπων που είναι πολύ μακριά από τα υπόλοιπα δεδομένα. Τα ακραία σημεία και τα σημεία θορύβου μπορούν να επηρεάσουν τη θέση των κέντρων και να αναγκάσουν τον αλγόριθμο να παράγει κακά αποτελέσματα.

Είναι ευαίσθητος στην κλίμακα των δεδομένων: Ο αλγόριθμος k-means είναι ευαίσθητος στην κλίμακα των δεδομένων, επομένως οποιαδήποτε μεταβλητή με υψηλότερη κλίμακα θα επηρεάσει περισσότερο την απόσταση και τα αποτελέσματα της συσταδοποίησης. Αυτό μπορεί να αντιμετωπιστεί κανονικοποιώντας τα δεδομένα πριν από την εκτέλεση του αλγόριθμου. [14, 15]

2.3 Προσδιορισμός της παραμέτρου κ

Η παράμετρος κ στον αλγόριθμο κ-μέσων αναφέρεται στον αριθμό των συστάδων. Στην πράξη, ο αριθμός των συστάδων συχνά δεν είναι γνωστός και πρέπει να προσδιοριστεί χρησιμοποιώντας άλλες μεθόδους. Στις επόμενες υποενότητες θα περιγράψουμε μερικές δημοφιλείς μεθόδους για τον προσδιορισμό της τιμής κ.

2.3.1 Η μέθοδος του αγκώνα (elbow method)

Η μέθοδος του αγκώνα (elbow) είναι μια δημοφιλής μέθοδος για τον προσδιορισμό του βέλτιστου αριθμού συστάδων στον αλγόριθμο k-means. Στην εφαρμογή που αναπτύξαμε αυτή είναι η βασική μέθοδος που χρησιμοποιούμε για τον προσδιορισμό της παραμέτρου κ. Βασίζεται στην ιδέα ότι καθώς αυξάνεται ο αριθμός των συστάδων, το άθροισμα τετραγωνικού σφάλματος (SSE) θα πρέπει να μειώνεται. Το SSE είναι ένα μέτρο της απόστασης κάθε στιγμιότυπου από το αντίστοιχο κέντρο της συστάδας του. Το SSE υπολογίζεται ως το άθροισμα των τετραγωνικών αποστάσεων μεταξύ κάθε σημείου δεδομένων και του κέντρου της συστάδας του.

Η μέθοδος elbow λειτουργεί σχεδιάζοντας το SSE έναντι του αριθμού των συστάδων και επιλέγοντας την τιμή του k όπου το SSE αρχίζει να μειώνεται με πιο αργό ρυθμό. Το σημείο όπου συμβαίνει αυτό ονομάζεται σημείο του αγκώνα και χρησιμοποιείται για τον προσδιορισμό του βέλτιστου αριθμού συστάδων.

Τα βήματα της μεθόδου του αγκώνα είναι τα εξής:

1. Αρχικοποιεί τα k κέντρα, όπου k είναι ο αριθμός των συστάδων. Αυτό μπορεί να γίνει τυχαία ή χρησιμοποιώντας κάποια άλλη μέθοδο όπως το k-means++.

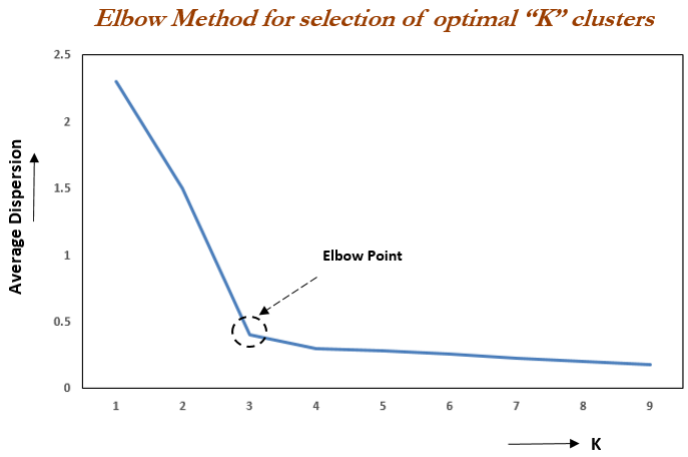
2. Αντιστοιχεί κάθε στιγμιότυπο στο πλησιέστερο κέντρο.

3. Υπολογίζει ξανά το SSE για κάθε συστάδα λαμβάνοντας το άθροισμα των τετραγωνικών αποστάσεων μεταξύ κάθε στιγμιοτύπου και του κέντρου της συστάδας του.

4. Επαναλαμβάνει τα βήματα 2 και 3 για διαφορετικές τιμές του k και σχεδιάζει το SSE σε σχέση με τον αριθμό των συστάδων.

5. Βρίσκουμε το σημείο του αγκώνα στο γράφημα, όπου το SSE αρχίζει να μειώνεται με πιο αργό ρυθμό. Ο αριθμός των συστάδων που αντιστοιχεί σε αυτό το σημείο είναι ο βέλτιστος αριθμός συστάδων(Σχήμα 2.5).

   Είναι σημαντικό να σημειωθεί ότι η μέθοδος αγκώνα δεν είναι πάντα μια αξιόπιστη μέθοδος για τον προσδιορισμό του βέλτιστου αριθμού συστάδων και το σημείο του αγκώνα μπορεί να μην είναι καθαρά ορατό, ειδικά όταν τα δεδομένα είναι θορυβώδη ή όταν οι συστάδες έχουν διαφορετικές πυκνότητες. Σε αυτήν την περίπτωση, συνιστάται η χρήση άλλων μεθόδων. [13]



Σχήμα 2.5: Το σημείο του αγκώνα στο elbow method

2.3.2 Η μέθοδος silhouette score

Η μέθοδος silhouette score είναι ένα μέτρο της ομοιότητας κάθε στιγμιοτύπου με την δικιά της συστάδα σε σύγκριση με άλλες συστάδες. Είναι ένας τρόπος αξιολόγησης της ποιότητας ενός αλγορίθμου συσταδοποίησης. Η silhouette score κυμαίνεται μεταξύ -1 και 1 και μια υψηλότερη βαθμολογία υποδηλώνει ότι το στιγμιότυπο ταιριάζει καλά με την δικιά του συστάδα. Ο βέλτιστος αριθμός συστάδων είναι αυτός που έχει ως αποτέλεσμα την υψηλότερη μέση βαθμολογία silhouette σε όλα τα στιγμιότυπα.

Η μέθοδος silhouette score υπολογίζεται για κάθε στιγμιότυπο ως εξής:

1. Για κάθε στιγμιότυπο, υπολογίζει τη μέση απόσταση από τα άλλα στιγμιότυπα στην δικιά της συστάδα (a(i))
2. Για κάθε στιγμιότυπο, υπολογίζει την ελάχιστη μέση απόσταση από τα άλλα στιγμιότυπα σε άλλες συστάδες (b(i))
3. Υπολογίζουμε τη βαθμολογία silhouette για κάθε στιγμιότυπο ως: s(i) = (b(i) - a(i)) / maxa(i), b(i)
4. Η βαθμολογία για μια συσταδοποίηση είναι η μέση βαθμολογία silhouette σε όλα τα στιγμιότυπα.

Η silhouette score κυμαίνεται μεταξύ -1 και 1, όπου η βαθμολογία 1 υποδηλώνει ότι το στιγμιότυπο ταιριάζει καλά με την δικιά του συστάδα, η βαθμολογία 0 υποδηλώνει ότι το στιγμιότυπο είναι εξίσου παρόμοιο τόσο με την δικιά του συστάδα όσο και με άλλες συστάδες, και η βαθμολογία -1 υποδηλώνει ότι το στιγμιότυπο δεν ταιριάζει καλά με την δικιά του συστάδα. [16]

2.3.3 Η μέθοδος στατιστική χάσματος(gap statistic)

Η στατιστική χάσματος είναι μια μέθοδος για τον προσδιορισμό του βέλτιστου αριθμού συστάδων στον αλγόριθμο k-means. Εισήχθη από τους Tibshirani, Walther και Hastie το 2001. Η μέθοδος συγκρίνει το αρχείο καταγραφής του αθροίσματος τετραγωνικού σφάλματος (SSE) για τον επιλεγμένο αριθμό συστάδων με το αναμενόμενο αρχείο του SSE για μια κατανομή αναφοράς. Ο βέλτιστος αριθμός συστάδων

είναι αυτός που μεγιστοποιεί το χάσμα μεταξύ των δύο τιμών. Η στατιστική του χάσματος βασίζεται στην ιδέα ότι το SSE θα πρέπει να μειώνεται καθώς αυξάνεται ο αριθμός των συστάδων. Ωστόσο, σε κάποιο σημείο η προσθήκη περισσότερων συστάδων δεν θα βελτιώσει σημαντικά το SSE. Η στατιστική του χάσματος στοχεύει να βρει αυτό το σημείο συγκρίνοντας το SSE των πραγματικών δεδομένων με αυτό μιας κατανομής αναφοράς των δεδομένων.

Τα βήματα της μεθόδου της στατιστικής χάσματος είναι τα εξής:

1. Δημιουργεί σύνολα δεδομένων αναφοράς B δειγματίζοντας τα σημεία δεδομένων ανεξάρτητα και ομοιόμορφα τυχαία.

2. Εφαρμόζει συσταδοποίηση k-means σε κάθε σύνολο δεδομένων αναφοράς και υπολογίζει το SSE.

3. Υπολογίζει τον μέσο όρο και την τυπική απόκλιση του καταγραφικού SSE για κάθε τιμή του k από τα σύνολα δεδομένων αναφοράς B.

4. Εφαρμόζει ομαδοποίηση k-means στο πραγματικό σύνολο δεδομένων και υπολογίζει το SSE για κάθε τιμή του k.

5. Υπολογίζει τη στατιστική του χάσματος για κάθε τιμή του k ως τη διαφορά μεταξύ του αρχείου καταγραφής SSE για το πραγματικό σύνολο δεδομένων και του αναμενόμενου αρχείου καταγραφής SSE για τα σύνολα δεδομένων αναφοράς.

6. Ο βέλτιστος αριθμός συστάδων είναι η τιμή του k που μεγιστοποιεί τη στατιστική του χάσματος.

Είναι σημαντικό να σημειωθεί ότι η στατιστική του χάσματος είναι ευαίσθητη στον αριθμό των συνόλων δεδομένων αναφοράς B. Ένας μεγάλος αριθμός συνόλων δεδομένων αναφοράς μπορεί να οδηγήσει σε μια καλή εκτίμηση του αναμενόμενου SSE, ωστόσο, οδηγεί επίσης σε υψηλό υπολογιστικό κόστος. Συνιστάται η χρήση ενός αριθμού συνόλων δεδομένων αναφοράς μεταξύ 10 και 50, ανάλογα με το μέγεθος του συνόλου δεδομένων και τους διαθέσιμους υπολογιστικούς πόρους. [17]

2.3.4 Η μέθοδοοι Bayesian Information Criteria (BIC) και Akaike Information Criterion (AIC)

Τα Bayesian Information Criteria (BIC) και το Akaike Information Criterion (AIC) είναι κριτήρια επιλογής μοντέλων που μπορούν να χρησιμοποιηθούν για τον προσδιορισμό του βέλτιστου αριθμού συστάδων στον αλγόριθμο k-means. Αυτές οι μέθοδοι βασίζονται στην ιδέα ότι το βέλτιστο μοντέλο είναι αυτό που εξισορροπεί την πιθανότητα των δεδομένων δεδομένου την πολυπλοκότητα του μοντέλου.Το BIC και το AIC χρησιμοποιούνται και τα δύο για τη σύγκριση διαφορετικών μοντέλων με βάση την πιθανότητα και την πολυπλοκότητά τους. Οι χαμηλότερες βαθμολογίες BIC και AIC υποδεικνύουν καλύτερο μοντέλο.

AIC: Το Akaike Information Criterion (AIC) είναι ένα μέτρο της σχετικής ποιότητας ενός στατιστικού μοντέλου. Βασίζεται στη συνάρτηση πιθανότητας, η οποία ποσοτικοποιεί την καλή προσαρμογή του μοντέλου στα δεδομένα. Ορίζεται ως AIC = 2k - 2ln(L), όπου k είναι ο αριθμός των παραμέτρων στο μοντέλο και L είναι η πιθανότητα των δεδομένων.

BIC: Τα Bayesian Information Criteria (BIC) είναι ένα μέτρο της σχετικής ποιότητας ενός στατιστικού μοντέλου. Βασίζεται στη συνάρτηση πιθανότητας και λαμβάνει επίσης υπόψη τον αριθμό των παρατηρήσεων. Ορίζεται ως BIC = kln(n) - 2ln(L), όπου k είναι ο αριθμός των παραμέτρων στο μοντέλο, n είναι ο αριθμός των παρατηρήσεων και L είναι η πιθανότητα των δεδομένων.

Για να χρησιμοποιήσετε αυτές τις μεθόδους για τον προσδιορισμό του βέλτιστου αριθμού συστάδων, τα βήματα είναι τα εξής:

1. Εκτελεί τον k-means για διαφορετικές τιμές του k και υπολογίζει τη βαθμολογία BIC ή AIC για κάθε τιμή του k.
2. Ο βέλτιστος αριθμός συστάδων είναι η τιμή του k που οδηγεί στη χαμηλότερη βαθμολογία BIC ή AIC. [18]

2.4 Παραλλαγές του αλγορίθμου κ-μέσων

Υπάρχουν διάφορες παραλλαγές του αλγορίθμου k-means που έχουν αναπτυχθεί για την αντιμετώπιση ορισμένων περιορισμών ή για τη βελτίωση της απόδοσης του αρχικού αλγορίθμου. Μερικές από τις πιο αξιοσημείωτες παραλλαγές του k-means είναι:

K-means++: Αυτή η παραλλαγή αντιμετωπίζει το πρόβλημα της κακής αρχικοποίησης επιλέγοντας πιο έξυπνα τα αρχικά κέντρα, γεγονός που οδηγεί σε ταχύτερη σύγκλιση και καλύτερα τελικά αποτελέσματα [19].

K-medoids: Αυτή η παραλλαγή είναι παρόμοια με το k-means, αλλά αντί να χρησιμοποιεί τον μέσο όρο των στιγμιοτύπων σε μια συστάδα ως κέντρο, χρησιμοποιεί ένα από τα στιγμιότυπα της συστάδας (ένα ”medoid”) ως κέντρο. Αυτό μπορεί να είναι πιο ανθεκτικό σε ακραίες τιμές και μπορεί να χειριστεί κατηγορηματικά δεδομένα [20].

Fuzzy K-means: Αυτή η παραλλαγή επιτρέπει σε ένα στιγμιότυπο να ανήκει σε περισσότερες από μια συστάδες, με διαφορετικούς βαθμούς συμμετοχής. Αυτό μπορεί να είναι χρήσιμο σε περιπτώσεις όπου τα σημεία δεδομένων δεν ανήκουν σαφώς σε μία μόνο συστάδα [21].

K-modes: Αυτή η παραλλαγή χρησιμοποιείται για κατηγορικά δεδομένα και αντικαθιστά τη μέση τιμή με τη λειτουργία ως πρωτότυπο της συστάδας [22].

K-prototypes: Αυτή η παραλλαγή είναι ένας συνδυασμός K-means και K-mode για το χειρισμό μικτών τύπων δεδομένων (αριθμητικά και κατηγορικά) [23]

Expectation-maximization (EM)-K-means: Αυτή η παραλλαγή χρησιμοποιεί τον αλγόριθμο Expectation-Maximization για να εκτιμήσει τις παραμέτρους της κατανομής Gauss για κάθε συστάδα, η οποία μπορεί να είναι χρήσιμη όταν τα δεδομένα δεν είναι καλά διαχωρισμένα ή όταν οι συστάδες έχουν διαφορετικά σχήματα [24]

Spherical K-means: Αυτή η παραλλαγή είναι μια τροποποιημένη έκδοση του k-means όπου η μετρική απόστασης που χρησιμοποιείται είναι ομοιότητα συνημιτόνου αντί για ευκλείδεια απόσταση, η οποία

είναι χρήσιμη όταν αντιμετωπίζουμε δεδομένα υψηλών διαστάσεων [25].

Mini-batch K-means: Αυτή η παραλλαγή είναι μια παραλλαγή του αλγόριθμου k-means που χρησιμοποιεί τυχαίο υποσύνολο των στιγμιοτύπων αντί για ολόκληρο το σύνολο δεδομένων για την ενημέρωση των κέντρων, το οποίο μπορεί να είναι πιο γρήγορο και πιο αποδοτικό στη μνήμη για μεγάλα σύνολα δεδομένων [26].

Κεφάλαιο 3ο: Τεχνολογίες που χρησιμοποιήθηκαν

Στο κεφάλαιο αυτό θα γίνει μία εισαγωγή στις τεχνολογίες και τα εργαλεία που χρησιμοποιήθηκαν για την ανάπτυξη της εφαρμογής που αφορά την εργασία.

3.1 Back-end

Το back-end μιας διαδικτυακής εφαρμογής αναφέρεται στο τμήμα της εφαρμογής που δεν είναι ορατά στον χρήστη. Αυτό περιλαμβάνει τον διακομιστή, την εφαρμογή και τη βάση δεδομένων.

3.1.1 PHP

Σχήμα 3.1: Λογότυπο PHP

Η PHP (Hypertext Preprocessor) είναι μια scripting γλώσσα προγραμματισμού η οποία τρέχει στην πλευρά του server που χρησιμοποιείται για τη δημιουργία ιστοσελίδων και δυναμικών εφαρμογών Ιστού. Είναι μια γλώσσα ανοιχτού κώδικα, γενικής χρήσης που χρησιμοποιείται ευρέως για την ανάπτυξη εφαρμογών ιστού, ιδιαίτερα για τη δημιουργία δυναμικών και διαδραστικών ιστοσελίδων [27].

Ο κώδικας PHP εκτελείται στον server και όχι στο πρόγραμμα περιήγησης ιστού(broweser) του πελάτη. Αυτό σημαίνει ότι ο server κάνει τη βαριά επεξεργασία του κώδικα PHP και τη δημιουργία του HTML ή άλλης εξόδου που αποστέλλεται στο πρόγραμμα περιήγησης του πελάτη για εμφάνιση. Αυτό καθιστά την PHP μια δημοφιλή επιλογή για τη δημιουργία εφαρμογών ιστού που απαιτούν επεξεργασία από την πλευρά του server, όπως e-commerce, συστήματα διαχείρισης περιεχομένου(CMS) και φόρουμ.

Η PHP χρησιμοποιείται συχνά σε συνδυασμό με άλλες τεχνολογίες, όπως η βάση δεδομένων MySQL και ο διακομιστής ιστού Apache. Ο συνδυασμός PHP, MySQL και Apache είναι γνωστός ως στοίβα "LAMP", που σημαίνει Linux, Apache, MySQL και PHP. Αυτή η στοίβα χρησιμοποιείται συνήθως στην ανάπτυξη ιστού επειδή είναι ανοιχτού κώδικα, είναι εύκολη στη ρύθμιση και μπορεί να εκτελεστεί σε διάφορες πλατφόρμες, συμπεριλαμβανομένων των Windows και του macOS.

Ο κώδικας PHP είναι ενσωματωμένος σε σελίδες HTML και μπορεί να αναμιχθεί με άλλες γλώσσες όπως JavaScript και CSS. Διαθέτει ενσωματωμένη υποστήριξη για εργασία με φόρμες, cookies και περιόδους σύνδεσης, γεγονός που καθιστά εύκολη τη δημιουργία διαδραστικών ιστότοπων. Η PHP υποστηρίζει επίσης διάφορες βιβλιοθήκες, πλαίσια και API για πρόσθετες λειτουργίες.

Η PHP ενημερώνεται και βελτιώνεται συνεχώς από την κυκλοφορία της το 1995, με την τελευταία σταθε-

ρή έκδοση να είναι η PHP 8.0. Έχει μια μεγάλη κοινότητα προγραμματιστών που δημιουργούν συνεχώς νέες βιβλιοθήκες και εργαλεία για να βοηθήσουν τους προγραμματιστές να εργάζονται με την PHP πιο αποτελεσματικά. [28]

3.1.2 Composer

Σχήμα 3.2: Λογότυπο Composer

Το Composer είναι ένας διαχειριστής εξαρτήσεων(dependencies) για την PHP. Είναι ένα εργαλείο γραμμής εντολών που επιτρέπει στους προγραμματιστές να διαχειρίζονται τις εξαρτήσεις των εφαρμογών PHP.

Οι εξαρτήσεις στο πλαίσιο της ανάπτυξης της PHP αναφέρονται στις εξωτερικές βιβλιοθήκες και πακέτα στα οποία βασίζεται μια εφαρμογή για να λειτουργήσει. Η διαχείριση αυτών των εξαρτήσεων μπορεί να είναι μια πολύπλοκη και χρονοβόρα εργασία, ειδικά καθώς μια εφαρμογή μεγαλώνει και εξελίσσεται με την πάροδο του χρόνου. Το Composer αντιμετωπίζει αυτό το πρόβλημα αυτοματοποιώντας τη διαδικασία διαχείρισης εξαρτήσεων και διατηρώντας τις ενημερωμένες.

Με το Composer, οι προγραμματιστές μπορούν να καθορίσουν τις εξαρτήσεις της εφαρμογής τους σε ένα μόνο αρχείο, που ονομάζεται composer.json. Αυτό το αρχείο παραθέτει τα απαιτούμενα πακέτα, τις εκδόσεις τους και τυχόν άλλες εξαρτήσεις που έχουν. Μόλις ρυθμιστεί το αρχείο διαμόρφωσης, ο προγραμματιστής μπορεί να εκτελέσει την εντολή composer για αυτόματη λήψη και εγκατάσταση των καθορισμένων πακέτων, μαζί με τις εξαρτήσεις τους [29].

3.1.3 REST API

Σε αυτήν την υποενότητα θα μιλήσουμε για το τι είναι API και για την αρχιτεκτονική REST.

Τι είναι API;
Ένα API (Application Programming Interface) είναι ένα σύνολο κανόνων και πρωτοκόλλων για τη δημιουργία και την αλληλεπίδραση με εφαρμογές λογισμικού. Χρησιμεύει ως διεπαφή μεταξύ διαφορετικών συστημάτων λογισμικού και τους επιτρέπει να επικοινωνούν μεταξύ τους.

Τα API επιτρέπουν στους προγραμματιστές να έχουν πρόσβαση στη λειτουργικότητα μιας συγκεκριμένης εφαρμογής λογισμικού ή υπηρεσίας. Για παράδειγμα, ένας προγραμματιστής μπορεί να χρησιμοποιήσει ένα API για να αποκτήσει πρόσβαση σε μια βάση δεδομένων, να ανακτήσει δεδομένα και να τα εμφανίσει στον δικό του ιστότοπο ή εφαρμογή. Αυτό επιτρέπει τη δημιουργία νέων εφαρμογών που αξιοποιούν τη λειτουργικότητα των υπαρχόντων συστημάτων.

Τα API μπορούν να ταξινομηθούν σε διαφορετικούς τύπους ανάλογα με τη λειτουργικότητά τους, όπως:

API που βασίζονται στον ιστό: Πρόκειται για API στα οποία είναι δυνατή η πρόσβαση μέσω Διαδικτύου χρησιμοποιώντας το τυπικό πρωτόκολλο HTTP. Συχνά χρησιμοποιούνται για πρόσβαση σε υπηρεσίες που βασίζονται στο διαδίκτυο, όπως πλατφόρμες μέσων κοινωνικής δικτύωσης, υπηρεσίες καιρού και διαδικτυακές βάσεις δεδομένων.

API λειτουργικού συστήματος: Πρόκειται για API που παρέχουν πρόσβαση στην υποκείμενη λειτουργικότητα ενός λειτουργικού συστήματος, όπως τη δυνατότητα δημιουργίας και χειρισμού αρχείων ή πρόσβασης σε συσκευές υλικού.

Βιβλιοθήκη API: Πρόκειται για API που παρέχουν πρόσβαση σε μια προκατασκευασμένη συλλογή κώδικα, όπως μια βιβλιοθήκη μαθηματικών συναρτήσεων ή ρουτίνες επεξεργασίας εικόνας.

API βάσεων δεδομένων: Πρόκειται για API που παρέχουν έναν τρόπο στις εφαρμογές να αλληλεπιδρούν με βάσεις δεδομένων, επιτρέποντάς τους να ανακτούν, να ενημερώνουν και να διαγράφουν δεδομένα. [30]

Αρχιτεκτονική REST

Το REST (Representational State Transfer) είναι μια αρχιτεκτονική για την κατασκευή διαδικτυακών υπηρεσιών. Το RESTful API (Application Programming Interface) είναι ένα API που συμμορφώνεται με το αρχιτεκτονικό στυλ REST. Τα RESTful API χρησιμοποιούν αιτήματα HTTP για POST (δημιουργία), PUT (ενημέρωση), GET (ανάγνωση) και DELETE δεδομένων.

Τα RESTful API είναι χτισμένα πάνω αρχές της αρχιτεκτονικής REST και χρησιμοποιούν τις τυπικές μεθόδους HTTP (GET, POST, PUT, DELETE) για αλληλεπίδραση με πόρους ιστού. Κάθε πόρος αναγνωρίζεται από ένα μοναδικό URI (Uniform Resource Identifier) και μπορεί να προσπελαστεί χρησιμοποιώντας τις τυπικές μεθόδους HTTP. [31]

Ένα από τα βασικά χαρακτηριστικά των RESTful API είναι ότι είναι stateless, πράγμα που σημαίνει ότι ο server δεν αποθηκεύει καμία πληροφορία για τον πελάτη μεταξύ των αιτημάτων. Αυτό επιτρέπει μεγαλύτερη επεκτασιμότητα και ευελιξία, καθώς ο server δεν χρειάζεται να διατηρεί κατάσταση λειτουργίας για κάθε πελάτη. Αντίθετα, ο πελάτης πρέπει να περιλαμβάνει όλες τις απαραίτητες πληροφορίες σε κάθε αίτημα.

Τα RESTful API χρησιμοποιούν ένα τυπικό σύνολο κωδικών κατάστασης HTTP για να υποδείξουν την επιτυχία ή την αποτυχία των αιτημάτων. Για παράδειγμα, ένας κωδικός κατάστασης 200 OK υποδεικνύει ότι ένα αίτημα ήταν επιτυχές, ενώ ένας κωδικός κατάστασης 404 not found υποδηλώνει ότι δεν ήταν δυνατός ο πόρος που ζητήθηκε.

Τα RESTful API μπορούν να επιστρέψουν δεδομένα σε διάφορες μορφές, όπως XML και JSON, και συχνά χρησιμοποιούν την κεφαλίδα HTTP Content-Type για να υποδείξουν τη μορφή των δεδομένων που επιστρέφονται.

Τα RESTful API χρησιμοποιούνται ευρέως στην ανάπτυξη ιστού, ιδιαίτερα για τη δημιουργία διαδικτυακών υπηρεσιών και εφαρμογών για κινητές συσκευές. Πολλές δημοφιλείς υπηρεσίες ιστού, όπως

το Facebook, το Twitter και η Google, παρέχουν RESTful API που επιτρέπουν στους προγραμματιστές να έχουν πρόσβαση στη λειτουργικότητα και τα δεδομένα τους. [32]

3.1.4 Python

Σχήμα 3.3: Λογότυπο Python

Η Python είναι μια γλώσσα προγραμματισμού υψηλού επιπέδου, γενικής χρήσης που χρησιμοποιείται ευρέως για ανάπτυξη εφαρμογών ιστού, επιστημονικούς υπολογισμούς, ανάλυση δεδομένων, τεχνητή νοημοσύνη και πολλές άλλες εφαρμογές. Κυκλοφόρησε για πρώτη φορά το 1991 από τον Guido van Rossum και έκτοτε έχει αυξηθεί σε δημοτικότητα λόγω της απλότητας, της αναγνωσιμότητας του και του τεράστιου αριθμού βιβλιοθηκών που είναι διαθέσιμες για διάφορες εργασίες. [33]

Μερικά από τα βασικά χαρακτηριστικά της Python περιλαμβάνουν:

Εύκολη εκμάθηση και χρήση: Η Python έχει μια απλή και ευανάγνωστη σύνταξη, η οποία την καθιστά εξαιρετική επιλογή για αρχάριους. Έχει επίσης μια μεγάλη και ενεργή κοινότητα που παρέχει πληθώρα πόρων και σεμιναρίων για την εκμάθηση και τη χρήση της γλώσσας.

Αντικειμενοστραφή(object-oriented): Η Python είναι μια αντικειμενοστραφής γλώσσα προγραμματισμού, που σημαίνει ότι χρησιμοποιεί κλάσεις και αντικείμενα για την οργάνωση και τη δομή του κώδικα. Αυτό το καθιστά μια καλή επιλογή για την ανάπτυξη μεγάλων και πολύπλοκων εφαρμογών.

Interpreted: Η Python είναι μια ερμηνευμένη γλώσσα, που σημαίνει ότι ο κώδικας εκτελείται γραμμή προς γραμμή, αντί να μεταγλωττίζεται σε κώδικα μηχανής. Αυτό διευκολύνει την ανάπτυξη και τη δοκιμή κώδικα, καθώς τα σφάλματα αναφέρονται καθώς εμφανίζονται.

Dynamically-typed: Η Python είναι μια dynamically-typed γλώσσα, που σημαίνει ότι ο τύπος μιας μεταβλητής καθορίζεται κατά το χρόνο εκτέλεσης. Αυτό την καθιστά πιο ευέλικτη από τις statically-typed γλώσσες όπως η C ή η Java, αλλά μπορεί επίσης να την κάνει πιο επιρρεπή σε σφάλματα.

Εκτεταμένες βιβλιοθήκες: Ένα από τα κύρια πλεονεκτήματα της Python είναι ο τεράστιος αριθμός βιβλιοθηκών που είναι διαθέσιμες για διάφορες εργασίες. Ορισμένες δημοφιλείς βιβλιοθήκες περιλαμβάνουν τη NumPy για επιστημονικούς υπολογισμούς, τις Pandas για την ανάλυση δεδομένων και την Matplotlib για την οπτικοποίηση δεδομένων.

Cross-platform: Η Python μπορεί να τρέξει σε ένα ευρύ φάσμα πλατφορμών, συμπεριλαμβανομένων των Windows, Mac και Linux, καθιστώντας την εξαιρετική επιλογή για την ανάπτυξη εφαρμογών πολλαπλών πλατφορμών. [34]

Ένα από τα κύρια πλεονεκτήματα της Python για τη μηχανική μάθηση και την εξόρυξη δεδομένων είναι

ο μεγάλος αριθμός βιβλιοθηκών που διατίθενται για αυτές τις εργασίες. Μερικές από τις πιο δημοφιλείς βιβλιοθήκες για μηχανική μάθηση στην Python περιλαμβάνουν:

TensorFlow: Μια βιβλιοθήκη ανοιχτού κώδικα για τη δημιουργία και την ανάπτυξη μοντέλων μηχανικής εκμάθησης, που αναπτύχθηκε από την Google. Μπορεί να χρησιμοποιηθεί για ένα ευρύ φάσμα εργασιών, όπως η αναγνώριση εικόνας και ομιλίας, η επεξεργασία φυσικής γλώσσας και τα νευρωνικά δίκτυα [35].

Scikit-learn: Μια απλή και αποτελεσματική βιβλιοθήκη για μηχανική μάθηση στην Python, η οποία παρέχει ένα ευρύ φάσμα εργαλείων για προεπεξεργασία δεδομένων, επιλογή μοντέλου και αξιολόγηση [36].

Keras: Ένα API νευρωνικών δικτύων υψηλού επιπέδου, γραμμένο σε Python και ικανό να τρέχει πάνω από το TensorFlow. Έχει σχεδιαστεί για να επιτρέπει γρήγορο πειραματισμό με μοντέλα βαθιάς μάθησης [37].

PyTorch: Μια βιβλιοθήκη μηχανικής εκμάθησης ανοιχτού κώδικα για την Python, βασισμένη στο Torch. Παρέχει ένα ισχυρό και ευέλικτο πλαίσιο νευρωνικών δικτύων που επιτρέπει την εύκολη και διαισθητική ανάπτυξη μοντέλων βαθιάς μάθησης [38].

Pandas: Μια βιβλιοθήκη για χειρισμό και ανάλυση δεδομένων, που παρέχει γρήγορες, ευέλικτες και εκφραστικές δομές δεδομένων που έχουν σχεδιαστεί για να κάνουν την εργασία με "σχεσιακά" ή "επισημασμένα" δεδομένα τόσο εύκολη όσο και διαισθητική [39].

Numpy: Μια βιβλιοθήκη για τη γλώσσα προγραμματισμού Python, που προσθέτει υποστήριξη για μεγάλους, πολυδιάστατους πίνακες και άλλα [40].

3.1.5 MySQL

Σχήμα 3.4: Λογότυπο MySQL

Η MySQL είναι ένα ευρέως χρησιμοποιούμενο, ανοιχτού κώδικα σχεσιακό σύστημα διαχείρισης βάσεων δεδομένων (RDBMS) που χρησιμοποιείται συνήθως στην ανάπτυξη εφαρμογών ιστού και σε άλλες εφαρμογές. Αναπτύσσεται, διανέμεται και υποστηρίζεται από την Oracle Corporation. Η MySQL βα-

σίζεται στη δομημένη γλώσσα ερωτημάτων (SQL), η οποία είναι μια τυπική γλώσσα για τη διαχείριση σχεσιακών βάσεων δεδομένων [41].

Η MySQL είναι γνωστή για την αξιοπιστία, τη σταθερότητα και την ευκολία χρήσης της. Μερικά από τα βασικά χαρακτηριστικά της MySQL περιλαμβάνουν:

Ασφάλεια δεδομένων: Η MySQL παρέχει μια ποικιλία λειτουργιών που βοηθούν στη διασφάλιση της ασφάλειας των δεδομένων μας, όπως κρυπτογράφηση, επικύρωση κωδικού πρόσβασης και στοιχεία ελέγχου πρόσβασης.

Επεκτασιμότητα: Η MySQL μπορεί να χειριστεί μεγάλους όγκους δεδομένων και ταυτόχρονους χρήστες, καθιστώντας την εξαιρετική επιλογή για ιστότοπους και εφαρμογές υψηλής επισκεψιμότητας.

Υψηλή απόδοση: Η MySQL χρησιμοποιεί μια ποικιλία τεχνικών για τη βελτιστοποίηση της απόδοσης, όπως η προσωρινή αποθήκευση και η δημιουργία ευρετηρίου, γεγονός που της επιτρέπει να ανακτά γρήγορα και να χειρίζεται δεδομένα.

Υψηλή διαθεσιμότητα: Η MySQL παρέχει μια ποικιλία λειτουργιών για τη διασφάλιση υψηλής διαθεσιμότητας, όπως η αναπαραγωγή, η οποία επιτρέπει την αυτόματη αντιγραφή δεδομένων σε άλλους διακομιστές για δημιουργία αντιγράφων ασφαλείας και ανάκτηση από καταστροφή.

Υποστήριξη πολλαπλών πλατφορμών: Η MySQL μπορεί να εκτελεστεί σε ένα ευρύ φάσμα πλατφορμών, συμπεριλαμβανομένων των Windows, Mac και Linux, καθιστώντας την εξαιρετική επιλογή για την ανάπτυξη εφαρμογών πολλαπλών πλατφορμών.

Μεγάλη κοινότητα: Η MySQL έχει μια μεγάλη και ενεργή κοινότητα που παρέχει πληθώρα πόρων και σεμιναρίων για εκμάθηση και χρήση του λογισμικού.

Η MySQL χρησιμοποιείται ευρέως στην ανάπτυξη ιστού, ιδιαίτερα σε συνδυασμό με τη γλώσσα προγραμματισμού PHP, αλλά μπορεί επίσης να χρησιμοποιηθεί και σε άλλες εφαρμογές όπως η αποθήκευση δεδομένων, το ηλεκτρονικό εμπόριο και οι εφαρμογές καταγραφής. [42, 43]

3.1.6 XAMPP

Σχήμα 3.5: Λογότυπο XAMPP

Το XAMPP είναι ένα δωρεάν πακέτο λογισμικού ανοιχτού κώδικα που περιλαμβάνει τον διακομιστή ιστού Apache, MariaDB, PHP και Perl. Το XAMPP σημαίνει X (που σημαίνει ότι λειτουργεί σε διάφορες πλατφόρμες όπως Windows, Linux και macOS), Apache, MariaDB, PHP και Perl. Το XAMPP έχει

σχεδιαστεί για να διευκολύνει την εγκατάσταση και την εκτέλεση ενός διακομιστή web στον τοπικό μας υπολογιστή, ο οποίος μπορεί να είναι χρήσιμος για την ανάπτυξη εφαρμογών ιστού, τις δοκιμές και τη συντήρηση ενός ιστότοπου.

Το XAMPP περιλαμβάνει επίσης το phpMyAdmin, ένα διαδικτυακό εργαλείο για τη διαχείριση βάσεων δεδομένων MariaDB/MySQL. Αυτό μας επιτρέπει να δημιουργούμε και να διαχειριζόμαστε εύκολα βάσεις δεδομένων για τις δικές μας εφαρμογές ιστού. [44]

3.1.7 Postman

Σχήμα 3.6: Λογότυπο Postman

Το Postman είναι ένα δημοφιλές εργαλείο για εργασία με API (Application Programming Interfaces). Είναι ένα λογισμικό που επιτρέπει στους προγραμματιστές να δοκιμάζουν, να τεκμηριώνουν και να μοιράζονται API.

Με τον Postman, οι προγραμματιστές μπορούν εύκολα να δημιουργήσουν, να δοκιμάσουν και να μοιραστούν αιτήματα HTTP, τα οποία είναι τα δομικά στοιχεία των API. Το εργαλείο επιτρέπει τη δημιουργία σύνθετων αιτημάτων με πολλαπλές κεφαλίδες, παραμέτρους ερωτήματος και περιεχόμενο σώματος. Μας επιτρέπει επίσης να αποθηκεύσουμε και να οργανώσουμε αιτήματα σε συλλογές για εύκολη επαναχρησιμοποίηση και κοινή χρήση.

Το Postman παρέχει επίσης ένα ισχυρό πλαίσιο δοκιμών που επιτρέπει στους προγραμματιστές να γράφουν δοκιμαστικά σενάρια που επικυρώνουν αυτόματα τις απαντήσεις που λαμβάνουν από το API. Αυτό μπορεί να μας βοηθήσει να διασφαλίσουμε ότι το API λειτουργεί σωστά και ότι τυχόν αλλαγές που έγιναν στο API δεν παραβιάζουν την υπάρχουσα λειτουργικότητα. [45]

3.2 Front-end

Το front-end ενός ιστότοπου ή μιας διαδικτυακής εφαρμογής αναφέρεται στο τμήμα της εφαρμογής με το οποίο ο χρήστης αλληλεπιδρά άμεσα.

3.2.1 HTML

Τα αρχικά της HTML είναι Hypertext Markup Language δηλαδή γλώσσα σήμανσης υπερκειμένου. Είναι μία γλώσσα σήμανσης που χρησιμοποιείται για τη δημιουργία ιστοσελίδων. Η HTML χρησιμοποιείται για τη δημιουργία της δομής και της διάταξης μιας ιστοσελίδας χρησιμοποιώντας ένα σύνολο ετικετών

Σχήμα 3.7: Λογότυπο HTML

σήμανσης και χαρακτηριστικών (attributes). Αυτές οι ετικέτες και τα χαρακτηριστικά χρησιμοποιούνται για τον καθορισμό των διαφόρων στοιχείων μιας ιστοσελίδας, όπως επικεφαλίδες, παραγράφους, εικόνες, συνδέσμους και άλλα.

Η HTML αποτελείται από μια σειρά στοιχείων, τα οποία αντιπροσωπεύονται από ετικέτες(tags). Αυτές οι ετικέτες περικλείονται σε γωνιακές αγκύλες <> και καθορίζουν τον τύπο του στοιχείου που δημιουργείται ενώ με την αγκύλη </> δηλώνουμε τον τερματισμό της ετικέτας. Για παράδειγμα, η ετικέτα <p> χρησιμοποιείται για τη δημιουργία μίας παραγράφου και η ετικέτα <img> χρησιμοποιείται για την εισαγωγή μιας εικόνας. Επίσης, η HTML χρησιμοποιεί χαρακτηριστικά για να παρέχει πρόσθετες πληροφορίες σχετικά με ένα στοιχείο(element). Τα χαρακτηριστικά προστίθενται στην ετικέτα ανοίγματος ενός στοιχείου και χρησιμοποιούνται για τον καθορισμό ιδιοτήτων όπως το μέγεθος μιας εικόνας, το χρώμα του κειμένου ή και "κλάσεις" που αναφέρονται σε στοιχεία των css αρχείων (ενότητα 3.2.2). Τα έγγραφα HTML συνήθως τα ανοίγουμε από προγράμματα περιήγησης ιστού και συνήθως αποθηκεύονται με την επέκταση αρχείου .html ή .htm. [46]

3.2.2 CSS

Σχήμα 3.8: Λογότυπο CSS

Το CSS σημαίνει Cascading Style Sheets. Είναι μια γλώσσα που χρησιμοποιείται για να περιγράψει την παρουσίαση ενός εγγράφου γραμμένου σε γλώσσα σήμανσης. Χρησιμοποιείται πιο συχνά σε συνδυασμό με HTML και JavaScript για τη δημιουργία της διάταξης, των χρωμάτων και του συνολικού σχεδιασμού μίας ιστοσελίδας. Το CSS επιτρέπει στους προγραμματιστές να διαχωρίσουν την παρουσίαση μιας ιστοσελίδας από τη δομή και το περιεχόμενό της, το οποίο ορίζεται χρησιμοποιώντας HTML. Χρησιμοποιώντας CSS, οι προγραμματιστές μπορούν να ελέγξουν τη διάταξη, τα χρώματα, τις γραμματοσειρές και άλλα στοιχεία μιας ιστοσελίδας με αποτελεσματικό τρόπο. Το CSS χρησιμοποιεί ένα σύνολο κανόνων, που ονομάζονται selectors, για να στοχεύσει συγκεκριμένα στοιχεία HTML και να εφαρμόσει στυλ σε αυτά. Κάθε κανόνας αποτελείται από έναν selector, ο οποίος στοχεύει ένα στοιχείο

HTML, και ένα σύνολο ιδιοτήτων και τιμών, που ορίζουν τα στυλ που θα εφαρμοστούν σε αυτό το στοιχείο. Επίσης επιτρέπει στους προγραμματιστές να χρησιμοποιούν ”κλάσεις” και ids για να στοχεύουν συγκεκριμένα στοιχεία και να εφαρμόζουν στυλ σε αυτά [47]. Για παράδειγμα στο HTML στοιχείο <p class=”myClass”> το οποίο δημιουργεί μία παράγραφο έχουμε δώσει την κλάση με όνομα ”myClass” και πλέον μέσα στο css αρχείο μπορούμε να αναφερθούμε σε αυτήν την κλάση. Ακολουθεί ένα παράδειγμα όπου η συγκεκριμένη κλάση θα αλλάξει το μέγεθος της γραμματοσειράς και το χρώμα των γραμμάτων της παραγράφου:

```
myClass{
  font-size:10px;
  color:green;
}
```

Σχήμα 3.9: Παράδειγμα CSS

3.2.3 Bootstrap

Σχήμα 3.10: Λογότυπο Bootstrap

Το Bootstrap είναι ένα δωρεάν και ανοιχτού κώδικα framework ανάπτυξης front-end που αναπτύχθηκε από το Twitter. Χρησιμοποιείται για τη δημιουργία ιστοσελίδων και εφαρμογών ιστού με προτεραιότητα για κινητά. Το Bootstrap παρέχει ένα σύνολο προκαθορισμένων στοιχείων CSS και JavaScript, όπως μπάρες πλοήγησης (navigation bar), φόρμες, κουμπιά και άλλα, που μπορούν εύκολα να προσαρμοστούν και να χρησιμοποιηθούν σε εφαρμογές ιστού.

Το Bootstrap είναι χτισμένο πάνω από HTML, CSS και JavaScript και χρησιμοποιεί ένα σύστημα πλέγματος (grid) για να δημιουργήσει μια διάταξη. Το σύστημα πλέγματος επιτρέπει στους προγραμματιστές να δημιουργούν εύκολα διατάξεις πολλαπλών στηλών που προσαρμόζονται στο μέγεθος της οθόνης της συσκευής στην οποία προβάλλονται. Παρέχει επίσης ένα σύνολο προκαθορισμένων κλάσεων που μπορούν να προστεθούν σε στοιχεία HTML για γρήγορη προσθήκη στυλ και λειτουργικότητας. Προσφέρει επίσης μια ευρεία γκάμα προκατασκευασμένων στοιχείων διεπαφής χρήστη, όπως μόνταλ, καρουζέλ και άλλα, τα οποία μπορούν να εξοικονομήσουν πολύ χρόνο στους προγραμματιστές που δεν θέλουν να ξεκινήσουν από το μηδέν. [48]

3.2.4 JavaScript

Η JavaScript είναι μια γλώσσα προγραμματισμού που χρησιμοποιείται συνήθως για τη δημιουργία δυναμικών ιστοσελίδων. Είναι μια γλώσσα προγραμματισμού από την πλευρά του πελάτη, που σημαίνει ότι εκτελείται από το πρόγραμμα περιήγησης Ιστού στον υπολογιστή του πελάτη και όχι σε διακομιστή.

Σχήμα 3.11: Λογότυπο JavaScript

Η JavaScript επιτρέπει στους προγραμματιστές να προσθέτουν δυναμική συμπεριφορά σε ιστοσελίδες, όπως επικύρωση φόρμας, κινούμενα σχέδια και διαδραστικούς χάρτες. Επιτρέπει επίσης στους προγραμματιστές να δημιουργούν εφαρμογές ιστού που μπορούν να ενημερώσουν το περιεχόμενο και τις διεπαφές χωρίς να απαιτείται ανανέωση της σελίδας.

Η JavaScript μπορεί να προστεθεί σε ένα αρχείο HTML ή CSS συμπεριλαμβάνοντας το σε μια ετικέτα script ή μπορεί να αποθηκευτεί σε ένα εξωτερικό αρχείο με επέκταση .js και στη συνέχεια να συνδεθεί με το αρχείο HTML.

Ο κώδικας JavaScript μπορεί να χρησιμοποιηθεί για τον χειρισμό του Document Object Model (DOM), το οποίο είναι η δομή της ιστοσελίδας όπως φαίνεται από το πρόγραμμα περιήγησης. Αυτό επιτρέπει στους προγραμματιστές να αλλάζουν το περιεχόμενο, τη διάταξη και τα στυλ μιας ιστοσελίδας με βάση τις αλληλεπιδράσεις των χρηστών ή άλλα συμβάντα. [49]

Η JavaScript διαθέτει επίσης αρκετές βιβλιοθήκες και frameworks, όπως το jQuery και την React, που παρέχουν πρόσθετη λειτουργικότητα και διευκολύνουν την εργασία με JavaScript.

3.2.5 jQuery

Σχήμα 3.12: Λογότυπο jQuery

Η jQuery είναι μια βιβλιοθήκη JavaScript που απλοποιεί τη διαδικασία εργασίας με έγγραφα HTML, συμβάντα, κινούμενα σχέδια και άλλα σε σχέση με την ανάπτυξη εφαρμογών ιστού. Έχει σχεδιαστεί για να διευκολύνει την πλοήγηση και τον χειρισμό του DOM.

Ένα από τα βασικά χαρακτηριστικά του jQuery είναι η ικανότητά του να επιλέγει στοιχεία σε μια ιστοσελίδα χρησιμοποιώντας επιλογείς τύπου CSS. Αυτό διευκολύνει την επιλογή και τον χειρισμό συγκεκριμένων στοιχείων, όπως κουμπιά, φόρμες και άλλα μέρη της σελίδας.

Η jQuery παρέχει επίσης ένα απλό και συνεπές API για το χειρισμό συμβάντων, όπως κλικ, τοποθέτηση του δείκτη του ποντικιού και πατήματα πλήκτρων, το οποίο διευκολύνει την απόκριση στις αλληλεπιδράσεις των χρηστών. [50]

Κεφάλαιο 4ο: Σχεδίαση και Υλοποίηση του web k-means

Στο κεφάλαιο αυτό θα αναλυθεί η σχεδίαση και η υλοποίηση της διαδικτυακής εφαρμογής η οποία ονομάζεται web k-means. Πιο συγκεκριμένα θα μιλήσουμε για τις λειτουργικές απαιτήσεις της εφαρμογής,την αρχιτεκτονική, καθώς και κάποια κομμάτια κώδικα για να καταλάβουμε καλύτερα πως λειτουργεί η εφαρμογή.

4.1 Λειτουργικές Απαιτήσεις

Λειτουργική απαίτηση είναι μια περιγραφή της υπηρεσίας που πρέπει να προσφέρει το λογισμικό. Περιγράφει ένα σύστημα λογισμικού ή ένα στοιχείο του. Μια λειτουργία δεν είναι παρά μόνο είσοδος στο σύστημα λογισμικού, η συμπεριφορά του και η έξοδος του. [51]

Οι λειτουργικές απαιτήσεις του web k-means είναι οι εξής:

1. Εγγραφή νέου χρήστη

   Ένας χρήστης θα μπορεί να κάνει εγγραφή στο σύστημα μέσω μιας φόρμας η οποία θα αποτελείτε από τέσσερα πεδία: Όνομα, επίθετο, email, Κωδικός πρόσβασης. Έπειτα ο χρήστης θα λάβει ένα μήνυμα στο email για επιβεβαίωση του λογαριασμού.

2. Σύνδεση χρήστη στο σύστημα

   Ο χρήστης με βάση τα στοιχεία που έκανε εγγραφή, θα μπορεί να συνδεέτε στο σύστημα και να έχει πρόσβαση σε όλες τις λειτουργίες της εφαρμογής.

3. Επεξεργασία προσωπικών στοιχείων και κωδικού πρόσβασης

   Ο χρήστης θα έχει την δυνατότητα να αλλάξει τα προσωπικά του στοιχεία όπως το όνομα, επίθετο και τον κωδικό πρόσβασης.

4. Διαγραφή Λογαριασμού

   Ο χρήστης θα μπορεί να διαγράψει τον λογαριασμό του μετά από επιβεβαίωση από μήνυμα που θα του έχει σταλεί στο email του.

5. Ανάκτηση κωδικού πρόσβασης

   Ο χρήστης θα έχει την δυνατότητα να αλλάξει τον κωδικός πρόσβασης του εφόσον τον ξέχασε μέσω email επιβεβαίωσης.

6. Αρχική σελίδα

   Μια αρχική σελίδα η οποία θα περιγράφει στους χρήστες περί τίνος πρόκειται η εφαρμογή. Σε αυτήν την σελίδα θα έχουν πρόσβαση όλοι οι χρήστες είτε είναι εγγεγραμμένοι είτε όχι.

7. Ανέβασμα αρχείου

   Ο χρήστης θα μπορεί να ανεβάσει ένα αρχείο/σύνολο δεδομένων,οι έγκυροι τύποι αρχείων θα είναι csv, xls, xlsx. Το αρχείο που θα ανεβάζει ο χρήστης θα υπάρχει η δυνατότητα να είναι είτε προσωπικό είτε δημόσιο προς όλους τους χρήστες εφόσον του έχει δοθεί άδεια για να ανεβάσει δημόσια αρχεία από κάποιον διαχειριστή.

8. Διάβασμα αρχείου

   Ο χρήστης θα έχει την δυνατότητα επιλέγοντας κάποιο από τα αρχεία που είναι ανεβασμένα να δεί τα δεδομένα σε μορφή πίνακα.

9. Διαγραφή αρχείου

   Ο χρήστης θα έχει την δυνατότητα να διαγράψει κάποιο προσωπικό αρχείο ή κάποιο δημόσιο εφόσον του έχει δοθεί άδεια από καποιον διαχειριστη.

10. Μέθοδος του αγκώνα

    Ο χρήστης θα μπορεί να δώσει ένα μέγιστο αριθμό συστάδων για ένα συγκεκριμένο αρχείο. Στην συνέχεια εφαρμογή θα του εμφανίζει ένα γράφημα με τα αθροίσματα των τετραγωνικών σφαλμάτων και θα του προτείνει την καλύτερη τιμή για το κ (αριθμός συστάδων).

11. Συσταδοποίηση κ-μέσων

    Ο χρήστης θα μπορεί να τρέξει συσταδοποίηση κ-μέσων (k-means) σε ένα συγκεκριμένο αρχείο δίνοντας το κ (αριθμός συστάδων) είτε είναι το προτεινόμενο είτε κάποιο που θα επιλέξει αυτός. Στην συνέχεια θα εμφανίζεται ένας πίνακας με τα δεδομένα του αρχείου και σε ποιά συστάδα έχει ανατεθεί η κάθε γραμμή του αρχείου. Ο χρήστης θα έχει την δυνατότητα να κατεβάσει το αρχείο σε μορφή csv.

12. Ελεύθερο API

    Οι προγραμματιστές θα έχουν πρόσβαση σε κάποιες από τις λειτουργίες που περιγράφηκαν παραπάνω μέσα από API, καλώντας τα κατάλληλα endpoints, τα οποία θα επιστρέφουν το αποτέλεσμα σε JSON μορφή. Ο προγραμματιστής θα μπορεί να καλεί το API δίνοντας σε κάθε κλήση το API KEY το οποίο έχει παραχθεί κατά την εγγραφή στο σύστημα.

13. Οδηγίες για το API

    Θα υπάρχει μια σελίδα με οδηγίες για τους προγραμματιστές που θέλουν να χρησιμοποιήσουν το API. Η σελίδα θα έχει οδηγίες για κάθε endpoint όπως τις παραμέτρους που πρέπει να βάλει με παραδείγματα καθώς θα τον πληροφορεί και για το προσωπικό του API KEY.

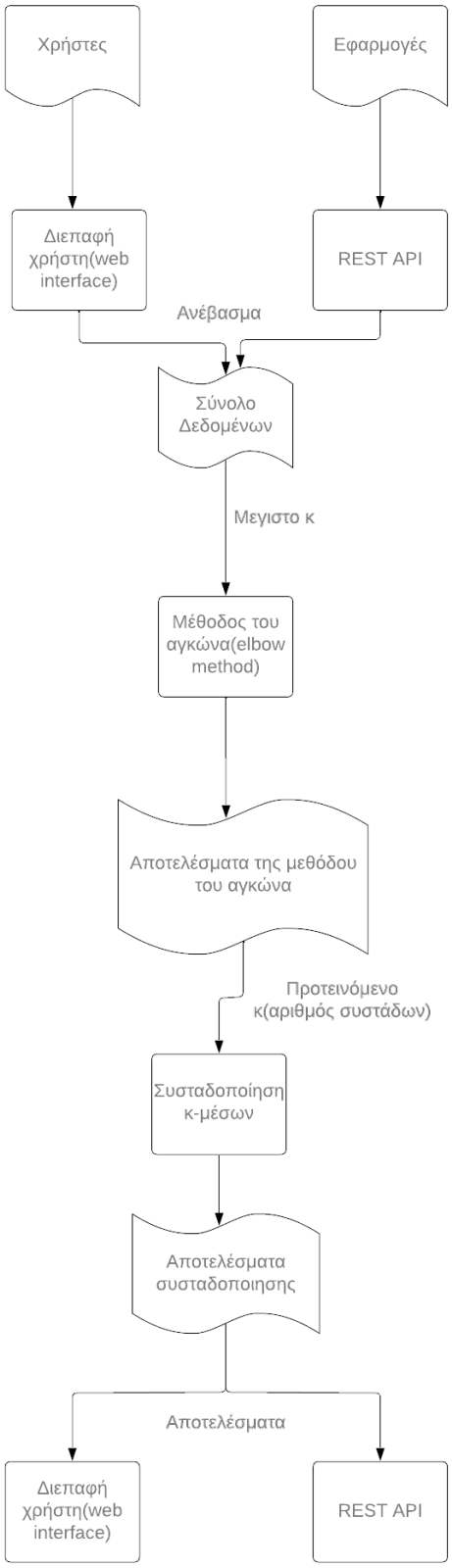
4.2 Αρχιτεκτονική

Η εφαρμογή έχει ως σκοπό την ανάπτυξη διαδικτυακής εφαρμογής όπου ο κάθε χρήστης θα μπορεί να ανεβάζει σύνολα δεδομένων και η εφαρμογή θα κατασκευάζει το γράφημα όπου θα παρουσιάζεται ο αγκώνας και θα προτείνει στον χρήστη πιθανή τιμή για την παράμετρο κ (αριθμός συστάδων) στην συνέχεια θα μπορεί να τρέξει συσταδοποίηση κ-μεσων δίνοντας την προτεινόμενη παράμετρο κ. Η ίδια διαδικασία θα είναι δυνατόν να γίνει μεσω ενός REST API. Η εφαρμογή αποτελείτε από 3 τμήματα. Τα module της python για τον υπολογισμό της μεθόδου του αγκώνα, και την συσταδοποίηση κ-μέσων. Μια σύγχρονη και φιλική πρός τον χρήστη διεπαφή ιστού και τέλος μια υπηρεσία REST API. Αυτά περιγράφονται και με με το σχήμα 4.1 που έχει ως εξής:

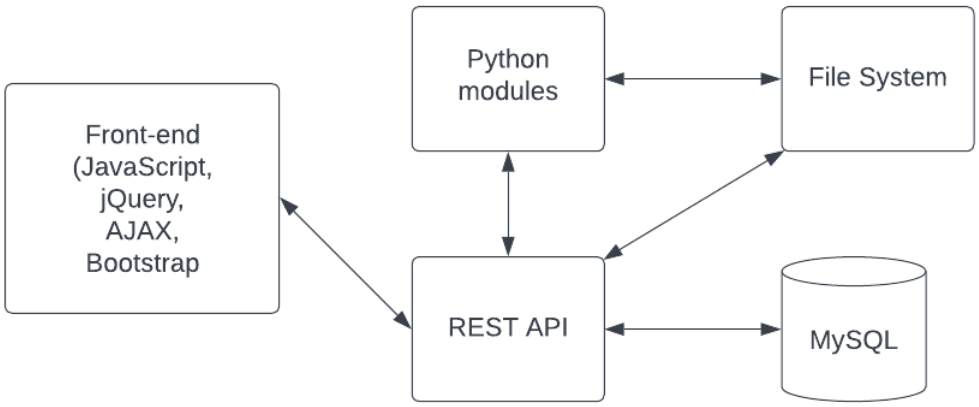
Μέσω της διεπαφής ιστού: Ο χρήστης συνδέετε με τα στοιχεία του (email, κωδικός πρόσβασης), εάν ο χρήστης δεν έχει κάνει εγγραφή θα πρέπει να κάνει. Ο χρήστης θα μπορεί να ανεβάσει σύνολα δεδομένων μέσω της διεπαφής στον διακομιστή. Στην συνέχεια μπορεί να επιλέξει κάποιο από τα σύνολα δεδομένων που υπάρχουν στον διακομιστή και να τρέξει την μέθοδο του αγκώνα ώστε να πάρει το προτεινόμενο κ και στην συνέχει να τρέξει την συσταδοποίηση. Στην συνέχει η διεπαφή θα εμφανίζει τα αποτελέσματα και θα δίνει την επιλογή στον χρήστη να τα κατεβάσει.

Μέσω της υπηρεσίας REST API: Ο χρήστης κάνει εγγραφή στο σύστημα ώστε να πάρει ενα API KEY. Μέσω αυτού του API KEY ο χρήστης μπορεί να καλεί τα endpoints για τις λειτουργίες που αναφέρθηκαν παραπάνω με την διαφορά ότι τα αποτελέσματα θα επιστρέφονται στον χρήστη σε μορφή JSON.

Οι τεχνολογίες που χρησιμοποιήθηκαν για την ανάπτυξη της διαδικτυακής εφαρμογής είναι PHP στο backend για το REST API και python για τα module της μεθόδου του αγκώνα και του αλγοριθμου κ-μέσων (k-means), στο front end χρησιμοποιήθηκαν javascript με την βιβλιοθήκη jQuery, αιτήματα AJAX και το framework Bootstrap για τη διεπαφή ιστού. Τέλος η βάση δεδομένων MySQL έχει δημιουργηθεί για τη διαχείριση των χρηστών, της εγγραφής και των επιπέδων πρόσβασης της εφαρμογής. Όλες αυτές οι τεχνολογίες και πως επικοινωνούν μεταξύ τους περιγράφονται στο σχήμα 4.2.



Σχήμα 4.1: Διάγραμμα ροής της εφαρμογής



Σχήμα 4.2: Διάγραμμα αρχιτεκτονικής της εφαρμογής

4.3 Δημόσια σύνολα δεδομένων και χρήστες

Τα δημόσια σύνολα δεδομένων είναι αυτά που μπορούν να χρησιμοποιηθούν ελεύθερα από όλους τους χρήστες που είναι εγγεγραμμένοι στην εφαρμογή. Για λόγους ασφαλείας, οι χρήστες που μπορούν να ανεβάσουν δημόσια σύνολα δεδομένων πρέπει να έχουν άδεια από κάποιον admin. Οι χρήστες μπορούν να ανεβάσουν προσωπικά σύνολα δεδομένων όπου θα μπορούν να τα βλέπουν και να τα χρησιμοποιήσουν μόνο αυτοί και δημόσια σύνολα δεδομένων όπου θα το βλέπουν όλοι οι χρήστες. Τα δημόσια είναι πολύ χρήσιμα για καθηγητές που πρέπει να μοιράζονται σύνολα δεδομένων με τους μαθητές τους για μαθητικούς ή πειραματικούς σκοπούς. Τα δημόσια είναι πολύ χρήσιμα για καθηγητές που πρέπει να μοιράζονται σύνολα δεδομένων με τους φοιτητές τους για πειραματικούς σκοπούς. Για να συμβεί αυτό, οι χρήστες πρέπει να κατηγοριοποιηθούν με βάση το επίπεδο προσβασιμότητα τους. Οι ρόλοι των χρηστών στην εφαρμογή μας είναι οι εξής:

- **Μη εγγεγραμμένος:** Αυτός ο τύπος χρήστη δεν έχει προνόμια και δεν μπορεί να χρησιμοποιήσει τον αλγόριθμο k-means, την μέθοδο του αγκώνα και το REST API. Φυσικά μπορούν να εγγραφούν για να γίνουν εγγεγραμμένος χρήστης.
- **Εγγεγραμμένος:** Ο εγγεγραμμένος χρήστης μπορεί να πραγματοποιήσει συσταδοποίηση είτε από τη διεπαφή ιστού είτε μέσω της υπηρεσίας REST API. Ο μόνος περιορισμός είναι το ανέβασμα δημόσιου συνόλου δεδομένων.
- **Δημιουργός δημόσιου συνόλου δεδομένων:** Αυτός ο χρήστης έχει τα δικαιώματα ενός εγγεγραμμένου χρήστη συν τη δυνατότητα δημιουργίας δημόσιων συνόλων δεδομένων. Ένας εγγεγραμμένος χρήστης πρέπει να αναβαθμιστεί σε δημιουργό δημόσιου συνόλου δεδομένων από κάποιον διαχειριστή ώστε να ανεβάσει δημόσιο σύνολο δεδομένων.
- **Διαχειριστής:** Έχει δικαιώματα δημιουργού δημόσιου συνόλου δεδομένων και δυνατότητα παραχώρησης ή ανάκλησης του ρόλου δημιουργού δημόσιου συνόλου δεδομένων μέσω του phpMyAdmin ή οποιουδήποτε client της MySQL, αλλάζοντας το συγκεκριμένο πεδίο στην βάση.

4.4 Back-end

4.4.1 Βάση δεδομένων

Για τη διαχείριση των χρηστών χρειάστηκε να δημιουργήσουμε μια σχεσιακή βάση δεδομένων, για αυτό το λόγο ως Σύστημα Διαχείρισης Βάσεων Δεδομένων επιλέχθηκε η mariaDB. Η mariaDB έχει αναπτυχθεί από την κοινότητα και είναι ένα fork του συστήματος διαχείρισης σχεσιακών βάσεων δεδομένων MySQL. [52] Η βάση αποτελείτε από δύο πίνακες και 6 αποθηκευμένες διαδικασίες (stored procedures) και παρακάτω περιγράφονται με περισσότερες λεπτομέρειες.

Οι πίνακες της βάσης είναι ο users και ο verification_users. Ο πίνακας users (Σχήμα 4.3) αποτελείτε από οκτώ πεδία και είναι τα εξής:

1. Το πεδίο id, το οποίο είναι το κύριο κλειδί του πίνακα, αποτελεί την ταυτότητα του χρήστη και παράγεται αυτόματα από τη βάση δεδομένων.

2. Το πεδίο email είναι η διεύθυνση ηλεκτρονικού ταχυδρομείου με την οποία κάνει εγγραφή ο χρήστης.
3. Το πεδίο password είναι ο κωδικός του χρήστη με τον οποίο κάνει εγγραφή και χρειάζεται για την είσοδο του στην εφαρμογή. Ο κωδικός είναι αποθηκευμένος σε hashed μορφή με τον αλγόριθμο MD5 για λόγους ασφαλείας. Ο αλγόριθμος MD5 είναι μια μονόδρομη κρυπτογραφική συνάρτηση που δέχεται ένα κείμενο οποιουδήποτε μήκους ως είσοδο και επιστρέφει ως έξοδο μια τιμή σύνοψης σταθερού μήκους που θα χρησιμοποιηθεί για τον έλεγχο ταυτότητας του αρχικού κειμένου. [53]
4. Τα πεδίο fname είναι το όνομα του χρήστη.
5. Τα πεδίο lname είναι το επίθετο του χρήστη.
6. Το πεδίο apiKey δημιουργείτε και αυτό με το αλγόριθμο MD5 χρησιμοποιώντας τυχαία αλφαριθμητικούς χαρακτήρες και είναι ένα είδος "κλειδιού" το οποίο χρησιμοποιείτε για την ταυτοποίηση του χρήστη κάθε φορά που κάνει κλήση σε κάποιο endpoint του REST API.
7. Το πεδίο verified το οποίο παίρνει τις τιμές 0 και 1 και χρησιμοποιείτε για να επιβεβαιώσουμε αν ο χρήστης έχει επαληθεύσει το email του.
8. Το πεδίο grandPublicDataset παίρνει τις τιμές 0 και 1 και χρησιμοποιείτε για να επιβεβαιώσουμε αν ο χρήστης έχει άδεια να ανεβάσει δημόσιο (public) σύνολο δεδομένων (dataset).

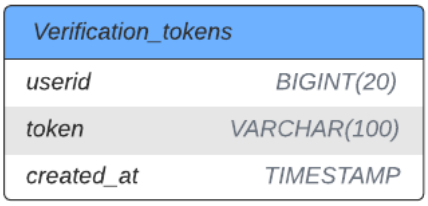
Users	
id	BIGINT(20)
email	VARCHAR(45)
password	VARCHAR(100)
fname	VARCHAR(45)
lname	VARCHAR(45)
apiKey	VARCHAR(100)
verified	TINYINT(1)
grandPublicDataset	TINYINT(1)

Σχήμα 4.3: Πίνακας users

Ο πίνακας verification_tokens (Σχήμα 4.4) είναι τα tokens των χρηστών, τα οποία δημιουργούνται όταν ο χρήστης κάνει κάποιο αίτημα προς την εφαρμογή. Τα αιτήματα αυτά είναι η εγγραφή στην εφαρμογή, αίτηση για ανάκτηση κωδικού πρόσβασης, αίτηση για διαγραφή του λογαριασμού. Ο πίνακας αποτελείτε από τρία πεδία και είναι τα εξής:

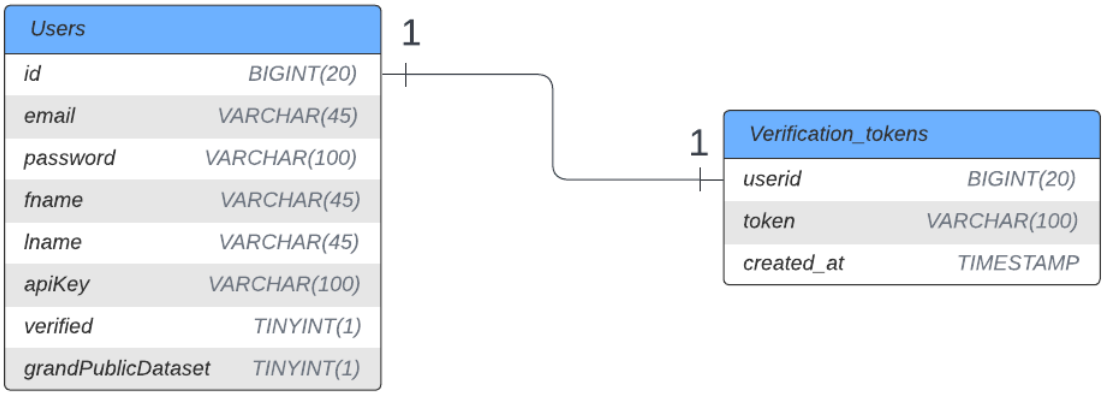
1. Το πεδίο userid είναι η ταυτότητα του χρήστη και είναι ξένο κλειδί στο πεδίο id του πίνακα users.
2. Το πεδίο token είναι σαν ένα είδος κλειδιού το οποίο χρησιμοποιείτε για να ταυτοποίηση κάποιο αίτημα του χρήστη. Το token δημιουργείτε με τον αλγόριθμο MD5 και αποτελείτε από τυχαία αλφαριθμητικά.

3. Το πεδίο created_at είναι η ημερομηνία και η ώρα της δημιουργίας του token και χρησιμοποιείτε για να ελέγξουμε αν το token έχει λήξει μετά από ένα καθορισμένο χρονικό διάστημα.



Σχήμα 4.4: Πίνακας verification_tokens

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 [54]. Το διάγραμμα ER για την εφαρμογή web k-means φαίνεται στο σχήμα 4.5.



Σχήμα 4.5: Διάγραμμα ER της εφαρμογής

Στήν βάση υπάρχουν και έξι αποθηκευμένες διαδικασίες (stored procedures). Μια αποθηκευμένη διαδικασία είναι ένας προετοιμασμένος κώδικας SQL που μπορούμε να αποθηκεύσουμε, ώστε ο κώδικας να μπορεί να χρησιμοποιηθεί ξανά και ξανά. Επίσης μπορούν να χρησιμοποιηθούν σε περίπτωση που θέλουμε να καλέσουμε πολλές SQL εντολές μαζί. [55]. Οι διαδικασίες είναι οι εξής:

1. forgotPassword η οποία δημιουργεί token σε περίπτωση που ο χρήστης κάνει αίτηση για ανάκτηση του κωδικού πρόσβασης του.

2. updatePassword η οποία ενημερώνει τον κωδικό πρόσβασης του χρήστη και τον αποθηκεύει σε hashed μορφή. Στην συνέχεια διαγράφει το token από τον πίνακα verification_tokens που δημιουργήθηκε από την προηγούμενη διαδικασία.

3. registerUser η οποία εισάγει τα στοιχεία του χρήστη όταν κάνει εγγραφή στον πίνακα users και δημιουργεί token για την επιβεβαίωση του email.

4. verifyAccount η οποία αλλάζει το πεδίο verified στον πίνακα Users. Στην συνέχεια διαγράφει το token από τον πίνακα verification_tokens που δημιουργήθηκε από την προηγούμενη διαδικασία.

5. deleteUser που χρησιμοποιείτε για την διαγραφή του χρήστη ώστε να δημιουργήσει ένα token στον πίνακα verification_tokens. Στην συνέχεια αυτό το token θα σταλεί με email στον χρήστη για να επιβεβαιώσουμε την ενέργεια αυτή.

6. verifyDelete η οποία διαγράφει τα στοιχεία του χρήστη από τον πίνακα Users. Στην συνέχεια διαγράφει το token από τον πίνακα verification_tokens που δημιουργήθηκε από την προηγούμενη διαδικασία.

```
1  CREATE DEFINER=`` PROCEDURE `verifyAccount`(vtoken varchar(100))
2  BEGIN
3  UPDATE users
4  JOIN verification_tokens ON users.id=verification_tokens.userid
5  SET users.verified=1 where verification_tokens.token=vtoken;
6
7  DELETE FROM verification_tokens where token=vtoken;
8
9  END
```

Σχήμα 4.6: Παράδειγμα αποθηκευμένης διαδικασίας

4.4.2 Υλοποίηση του REST API

Το REST API είναι από τα βασικά μέρη της εφαρμογής καθώς καλείτε από το front-end ή μπορεί να το καλέσει κάποιος προγραμματιστής για την δική του εφαρμογή ώστε να τρέξουν όλες οι λειτουργίες που περιγράψαμε στην ενότητα 4.1. Γενικά για να καλέσουμε μια λειτουργία του REST API καλούμε το κατάλληλο endpoint και αυτό μας επιστρέφει τα αποτελέσματα σε μορφή JSON. Το endpoint είναι στην ουσία ένα URI (Uniform Resource Identifier) στο οποίο καλείτε το αντίστοιχτο script στον διακομιστή. Το REST API για την εφαρμογή αποτελείτε από 10 scripts που φαίνονται στον πίνακα 4.1.

Πίνακας 4.1: Τα scripts του REST API

get-datasets.php	Επιστρέφει όλα τα dataset,public και personal
read-dataset.php	Επιστρέφει τα δεδομένα ενός dataset
elbow.php	Επιστρέφει τα τετραγωνικά λάθη(SSEs) για την δημιουργία του elbow chart
clusters.php	Αναθέτει σε συστάδες κάθε γραμμή ενός dataset
upload-dataset.php	Ανεβάζει στον διακομιστή ένα dataset
delete-dataset.php	Διαγράφει ένα dataset
register.php	Εγγραφή χρήστη
login.php	Σύνδεση ενός χρήστη
edit-profile.php	Αλλάζει τα στοιχεία του χρήστη
delete-user.php	Διαγράφει έναν χρήστη

Το REST API εχεί δημιουργηθεί με την γλώσσα σεναρίων (scripting) PHP. Κάθε φορά που κάποιος κάνει κλήση σε ένα endpoint στον διακομιστή εκτελείτε το αντίστοιχο PHP script. Προφανώς θα πρέπει να βάλουμε και κάποιες παραμέτρους, όπως για παράδειγμα το apiKey του χρήστη ώστε να γίνει η αυθεντικοποίηση του ή παραμέτρους που χρειάζονται σαν είσοδος στο script και ανάλογα με αυτές μπορεί να αλλάξει όλη η πορεία του. Τις παραμέτρους τις περνάμε με βάση τον τύπο της μεθόδου του endpoint. Για πα-

ράδειγμα αν το endpoint είναι τύπου GET τις παραμέτρους τις περνάμε στο URI. Για παράδειγμα αν καλέσουμε το endpoint "/server/api/read-dataset.php?apikey=usersApiKey&dataset=iris.csv&dataset-type=public" καλούμε το php script read-dataset και μετά το αγγλικό ερωτηματικό μπαίνουν οι παράμετροι, σε μορφή παράμετρος=τιμή και χωρίζονται μεταξύ τους με το σύμβολο "&", στο συγκεκριμένο παράδειγμα περνάμε το apiKey του χρήστη, το dataset με όνομα iris.csv και τον τύπο του dataset που παίρνει τις τιμές public ή personal. Αν ο τύπος του endpoint δεν είναι get και είναι POST, DELETE, PUT κ.α, τότε τις παραμέτρους τις περνάμε στο "body" μίας κλήσης όπως φαίνεται παρακάτω:

```
"apiKey":"usersApiKey",
"dataset":"iris.csv",
"type":"public"
```

Για να καλέσει κάποιος ένα endpoint όπως είδαμε παραπάνω θα πρέπει να βάλει πρώτα το domain του server ο οποίος είναι "https://nireas.iee.ihu.gr/webkmeans/", έπειτε ακολουθεί το endpoint το οποίο είναι ο φάκελος server μετά ο φάκελος api και μετά το script που θέλουμε να καλέσουμε. Το REST API της συγκεκριμένης εφαρμογής αποτελείται από δέκα endpoints.

Για να ελέγξουμε ότι ο χρήστης έχει περάσει όλες τις απαραίτητες παραμέτρους που χρειάζονται για το script ενός endpoint στην αρχή κάθε script γίνονται οι κατάλληλοι έλεγχοι, όπως φαίνεται στο σχήμα 4.7

```php
if($method!= "GET") {
    header("HTTP/1.1 403 Forbidden");
    print json_encode(['errormesg'=>"Method $method not allowed here."]);
    exit;
}

if(!isset($_GET['apikey'])){
    header("HTTP/1.1 400 Bad Request");
    print json_encode(['errormesg'=>"apikey is required"]);
    exit;
}

if(!isset($_GET['dataset'])){
    header("HTTP/1.1 400 Bad Request");
    print json_encode(['errormesg'=>"dataset is required"]);
    exit;
}

if(!isset($_GET['dataset-type'])){
    header("HTTP/1.1 400 Bad Request");
    print json_encode(['errormesg'=>"dataset-type is required"]);
    exit;
}
```

Σχήμα 4.7: Κώδικας για έλεγχο σωστών παραμέτρων

Για αρχή το πρώτο if statement ελέγχει αν το endpoint καλείτε με την σωστή μέθοδο σε αυτήν την περίπτωση είναι το GET. Αν ο χρήστης ή η εφαρμογή το έχει καλέσει με κάποια άλλη μέθοδο τότε επιστρέφετε ο κατάλληλος αριθμητικός κωδικός με κάποιο μήνυμα και με την εντολή exit τερματίζεται το script σε αυτό το σημείο, οπότε δεν συνεχίζει παρακάτω. Τα υπόλοιπα τρία if statements ελέγχουν αν εχουν δωθεί οι κατάλληλοι παραμέτροι, σε αυτήν την περίπτωση το apiKey, το dataset και το dataset-type. Στην συνέχεια που φαίνεται στο σχήμα 4.8 ελέγχει αν το dataset-type παίρνει τις σωστές τιμές, δηλαδή personal

ή public.

```
if(!($_GET['dataset-type']=="public"||$_GET['dataset-type']=="personal")){
    header("HTTP/1.1 400 Bad Request");
    print json_encode(['errormesg'=>"dataset-type value can only be personal or public"]);
    exit;
}
```

Σχήμα 4.8: Κώδικας για τον έλεγχο του τύπου του dataset

Αναλόγως την τιμή που έχει περαστεί για αυτήν την παράμετρο το script θα ψάξει για το dataset που έχει δώσει ο χρήστης είτε στον προσωπικό φάκελο του χρήστη είτε στον φάκελο public. Στην συνέχεια ελέγχουμε αν το ApiKey που έχει δώσει ο χρήστης είναι σωστό (σχήμα 4.9).

```
if(!checkApiKeyExists($_GET['apikey'])){
    header("HTTP/1.1 401 Unauthorized");
    exit;
}
```

Σχήμα 4.9: Κώδικας για τον έλεγχο του API Key

Το μοντέλο που είδαμε παραπάνω με τους ελέγχους εφαρμόζεται και στα δέκα script των endpoints της εφαρμογής, απλά αλλάζουν αντίστοιχα η μέθοδος και οι παραμέτροι για κάθε script.

Στο κεφάλαιο 2 αναφέραμε ότι ο αλγόριθμος k-means εφαρμόζεται μόνο σε αριθμητικά δεδομένα. Για αυτόν τον λόγο όταν καλούνται τα endpoints για την μέθοδο του αγκώνα ή για το clustering ο χρήστης πρέπει να επιλέξει τις στήλες για τις οποίες θα τρέξουν οι αντίστοιχοι αλγόριθμοι, οι οποίες πρέπει να περιέχουν αριθμητικά δεδομένα μόνο. Ο έλεγχος γίνετε με τον κώδικα που φαίνεται στο σχήμα 4.10 σε περίπτωση που ο χρήστης δώσει λανθασμένες στήλες επιστρέφετε κατάλληλο μήνυμα και τερματίζεται το script.

```
$numerical_columns=[];
foreach($headers_ as $value_){
    ${$value_}=array_column($full_csv, "$value_");
    if ( count( ${$value_} ) === count( array_filter( ${$value_}, 'is_numeric' ) ) ) {
       array_push($numerical_columns,$value_);
    }
}

if(!(array_intersect($columns, $numerical_columns) === $columns)){
    header("HTTP/1.1 400 Bad Request");
    print json_encode(['errormesg'=>"columns must be numerical"]);
    exit();
}
```

Σχήμα 4.10: Κώδικας για τον έλεγχο αριθμητικών δεδομένων

Σε όλα τα script που φαίνονται στον πίνακα 4.1 χρειάζεται να γίνει σύνδεση με την βάση είτε για να ελέγξει ότι το ApiKey είναι σωστό είτε για τα script register, login, edit-profile, delete-user που χρειάζονται να πάρουν πληροφορίες από την βάση ή να αλλάξουν κάτι στα δεδομένα της βάσης. Στο σχήμα 4.11 φαίνεται ο κώδικα με τον οποίο γίνετε η σύνδεση στην βάση.

```php
mysqli_report(MYSQLI_REPORT_ERROR | MYSQLI_REPORT_STRICT);
if(gethostname()=='nireas') {
    $mysqli = new mysqli($db_host, $db_user, $db_pass, $db_name);
} else {
        $mysqli = new mysqli($host, $user, $pass, $db);
}

if ($mysqli->connect_errno) {
    echo "Failed to connect to MySQL: (" .
    $mysqli->connect_errno . ") " . $mysqli->connect_error;
}
```

Σχήμα 4.11: Κώδικας για την σύνδεση με την βάση δεδομένων

Όταν κάποιος χρήστης κάνει εγγραφή, διαγραφή λογαριασμού, αίτηση για ανάκτηση κωδικού στο σύστημα είτε από το endpoint είτε από την διεπαφή χρήστη, λαμβάνει μήνυμα επιβεβαίωσης στο email με το οποίο έκανε εγγραφή. Ο κώδικας για την αποστολή του email φαίνεται στο σχήμα 4.12.

```php
try {
    //Server settings
    $mail->SMTPDebug = 0;                      //Enable verbose debug output
    $mail->isSMTP();                                            //Send using SMTP
    $mail->Host       = 'smtp-mail.outlook.com';                    //Set the SMTP server to send through
    $mail->SMTPAuth   = true;                                   //Enable SMTP authentication
    $mail->SMTPOptions=array('ssl'=>array(
        'verify_peer'=>false,
        'verify_peer_name'=>false,
        'allow_self_signed'=>true
    ));

    $mail->Username   = $username;                     //SMTP username
    $mail->Password   = $password;                               //SMTP password
    $mail->SMTPSecure = 'tls';            //Enable implicit TLS encryption
    $mail->Port       = 587;                                    //TCP port to connect to; use 587 if you have set `SMTPSecure

    //Recipients
    $mail->setFrom($username, 'WebKmeans');
    $mail->addAddress($recipient, $r_name);     //Add a recipient

    //Content
    $mail->isHTML(true);                                  //Set email format to HTML
    $mail->Subject = $subject;
    $mail->Body    = $body;
    $mail->AltBody = $altbody;

    $mail->send();

} catch (Exception $e) {
```

Σχήμα 4.12: Κώδικας για αποστολή email επιβεβαίωσης

Στην τελευταία γραμμή του κώδικα του σχήματος 4.12 υπάρχει το catch σε περίπτωση που κάτι πάει λάθος με το πρώτο email δοκιμάζει να τρέξει ξανά το script αλλά με διαφορετικό SMTP-SERVER και email.

Για τα script elbow και clusters γίνονται οι απαραίτητοι έλεγχοι και μετά καλείτε το python script μέσω του shell του διακομιστή και περνάμε τις κατάλληλες παραμέτρους. Στο σχήμα 4.13 φαίνεται ο κώδικας για την κλήση του elbow engine που έχει γραφτεί στην γλώσσα python.

```
$output=shell_exec("python ../python/elbow_module.py $path $colums_string $clusters $ext  2>&1");
echo ($output);
```

Σχήμα 4.13: Κώδικας για την κλήση ενός Python script μέσα από την PHP

Σχετικά με το script upload-dataset που αφορά το ανέβασμα συνόλου δεδομένων εάν ο χρήστης έχει επιλέξει να ανεβάσει το αρχείο στον δημόσιο φάκελο τότε γίνετε έλεγχος αν έχει άδεια να το κάνει. Στην συνέχεια ελέγχει αν υπάρχει ήδη το αρχείο που ανεβάζει. Εφόσον το αρχείο δεν υπάρχει ανεβαίνει στον φάκελο public_datasets. Αυτά φαίνονται και στον κώδικα του σχήματος 4.14. Η ιδία διαδικασία εκτελείται και σε περίπτωση που επιλέξει να ανεβάσει στον προσωπικό του φάκελο.

```
if($_POST['dataset-type']=="public"){
    $sql2 = 'SELECT grandPublicDataset FROM users WHERE apiKey=?';
    $st2 = $mysqli->prepare($sql2);
    $st2->bind_param('s',$_POST['apikey']);
    $st2->execute();
    $res = $st2->get_result();
    $res = $res->fetch_assoc();
    $grandpublicdataset=$res['grandPublicDataset'];

    if($grandpublicdataset==0){
        header("HTTP/1.1 403 Forbidden");
        print json_encode(['errormesg'=>"You dont have the permission to upload public dataset."]);
        exit;
    }
    $folder="../python/datasets/public_datasets/$filename";
    if(!file_exists($folder)){
        mkdir("$folder");
        move_uploaded_file($_FILES['dataset']['tmp_name'],"$folder/" . $file_name);
    }else{
        header("HTTP/1.1 400 Bad Request");
        print json_encode(['errormesg'=>"This dataset already exists"]);
        exit;
    }
```

Σχήμα 4.14: Κώδικας για ανέβασμα δημόσιου συνόλου δεδομένων

4.4.3 Υλοποίηση του elbow module

Όπως αναφέραμε και στο 2ο κεφάλαιο ο K-Means είναι ένας αλγόριθμος μηχανικής μάθησης χωρίς επίβλεψη που ομαδοποιεί δεδομένα σε k αριθμό συστάδων. Ο αριθμός των συστάδων ορίζεται από τον χρήστη και ο αλγόριθμος θα ομαδοποιήσει τα δεδομένα ακόμα κι αν αυτός ο αριθμός δεν είναι ο βέλτιστος για τη συγκεκριμένη περίπτωση.

Επομένως, πρέπει να αναπτύξουμε μια τεχνική που με κάποιο τρόπο θα μας βοηθήσει να αποφασίσουμε πόσα cluster θα χρησιμοποιήσουμε για τον αλγόριθμο K-Means.

Η μέθοδος Elbow είναι μια πολύ δημοφιλής τεχνική και η ιδέα είναι να εκτελεστεί συσταδοποίηση k-means για ένα εύρος συστάδων k (για παράδειγμα 1 έως το 10). Στην συνέχεια για κάθε τιμή, υ-πολογίζουμε το άθροισμα των τετραγωνικών αποστάσεων από κάθε στιγμιότυπο στο εκχωρημένο κέ-ντρο(παραμορφώσεις).

Όταν οι παραμορφώσεις σχεδιάζονται και η γραφική παράσταση μοιάζει με βραχίονα, τότε ο “αγκώνας”

(το σημείο καμπής στην καμπύλη) είναι η καλύτερη τιμή για το k.

Η υλοποίηση για την μέθοδο του αγκώνα γίνετε με την γλώσσα σεναρίων(scripting) Python καθώς διαθέτει έτοιμο τον αλγόριθμο k-means μέσα από την βιβλιοθήκη scikit-learn.

Ξεκινάμε δίνοντας σαν είσοδο το σύνολο δεδομένων διαβάζοντας το με την βιβλιοθήκη της Python Pandas. Στον κώδικα που φαίνεται στο σχήμα 4.15 ελέγχουμε τον τύπο του αρχείου αν είναι csv ή excel, ωστέ να καλέσουμε την σώστη μέθοδο της Pandas.

```
13  if (sys.argv[4]=="csv"):
14      df=pd.read_csv(sys.argv[1])
15  else:
16      df=pd.read_excel(sys.argv[1])
```

Σχήμα 4.15: Κώδικας για διάβασμα αρχείου

Στην συνέχεια θα πρέπει να κάνουμε κανονικοποίηση (normalize) τα δεδομένα. Η κανονικοποίηση των δεδομένων είναι σημαντική για να διασφαλιστεί ότι το μέτρο απόστασης έχει ίσο βάρος σε κάθε μεταβλητή. Χωρίς κανονικοποίηση, η μεταβλητή με τη μεγαλύτερη κλίμακα θα κυριαρχεί στο μέτρο [56]. Στον κώδικα του σχήματος 4.16 χρησιμοποιούμε την μέθοδο MinMaxScaler για να κάνουμε κανονικοποίηση σε όλες τις γραμμές του συνόλου δεδομένων.

```
21  scaler=MinMaxScaler()
22
23  for i in range(len(columns)):
24      scaler.fit(df[[columns[i]]])
25      df[columns[i]]=scaler.transform(df[[columns[i]]])
```

Σχήμα 4.16: Κώδικας για κανονικοποίηση των δεδομένων

Έπειτα, μπορούμε εύκολα να εκτελέσουμε τον K-Means για ένα εύρος από συστάδες που έχει επιλέξει ο χρήστης, χρησιμοποιώντας έναν βρόχο (loop) for και συλλέγοντας τις παραμορφώσεις σε μια λίστα (Σχήμα 4.17).

```
28  sse = []
29
30  for i in range(1,clusters+1):
31      kmeans = KMeans(n_clusters=i,n_init='auto')
32      kmeans.fit(df[columns])
33      sse.append(kmeans.inertia_)
```

Σχήμα 4.17: Κώδικας για τον υπολογισμό του αθροίσματος των τετραγωνικών αποστάσεων

Τέλος, υπολογίζουμε την καλύτερη τιμή για το κ με την μέθοδο KneeLocator της βιβλιοθήκης kneed. Στην συνέχεια επιστρέφουμε τα αποτελέσματα των SSE και την προτεινόμενη τιμή κ σε μορφή JSON (Σχήμα 4.18). Τα δεδομένα αυτά τα διαχειρίζεται για το πώς θα επιστρέψουν στον τελικό χρήστη η PHP στο script elbow.php που έκανε από την αρχή την κλήση του python script (Σχήμα 4.13).

4.4.4 Υλοποίηση του K-Means module

Η υλοποίηση του K-Means module αναλαμβάνει την συσταδοποιήση με τον αλγόριθμο k-means με την γλώσσα Python και την βιβλιοθήκη scikit-learn. Η διαδικασία ξεκινάει διαβάζοντας το αρχείο με την

```
result=list(map(str, sse))
kl=KneeLocator(range(1,clusters+1),sse,curve="convex",direction="decreasing")

print(json.dumps({"sse":result,"suggested-k":str(kl.elbow)}))
```

Σχήμα 4.18: Κώδικας για τον υπολογισμό της προτεινόμενης τιμής κ

βιβλιοθήκη Pandas και κανονικοποιώντας τα δεδομένα όπως είδαμε και στην προηγούμενη ενότητα. Στην συνέχεια εκτελούμε τον αλγόριθμο k-means με αριθμό συστάδων (κ) που θα επιλέξει ο χρήστης. Ο χρήστης μπορεί να επιλέξει όποια τιμή θέλει για το κ ή να επιλέξει την τιμή που του προτείνετε από την μέθοδο του αγκώνα. Το Python script που αναπτύξαμε καλείτε μέσα από το REST API και συγκεκριμένα από το script clusters.php.

Στον κώδικα του σχήματος 4.19 εκτελούμε τον k-means και στην συνέχεια προσθέτουμε μια καινούρια στήλη στην μεταβλητή όπου έχουμε διαβάσει το σύνολο δεδομένων. Η καινούργια στήλη έχει τις συστάδες που έχουν ανατεθεί σε κάθε γραμμή του συνόλου δεδομένων. Στην συνέχεια εξάγουμε το καινούριο σύνολο δεδομένων που έχει προκύψει μαζί με την στήλη των συστάδων σε ένα csv αρχείο. Το REST API και η PHP έχουν μετά την ευθύνη, ώστε να διαβάσουν το csv αρχείο και να επιστρέψουν στον τελικό χρήστη τα δεδομένα του.

```
kmeans=KMeans(n_clusters=clusters,n_init='auto')
predicted=kmeans.fit_predict(df[columns])
df1['cluster']=predicted+1

columns.append('cluster');

df1[columns].to_csv(sys.argv[5],index=False,encoding='utf-8')
```

Σχήμα 4.19: Κώδικας για την συσταδοιποίηση k-means

4.4.5 Απόδοση του k-means

Η εκτέλεση του αλγορίθμου k-means για την μέθοδο του αγκώνα και την ανάθεση συστάδων, έχει υψηλό κόστος στην CPU και στην RAM. Για αυτόν τον λόγο επιλέχθηκε η γλώσσα προγραμματισμού Python. Η βιβλιοθήκη Pandas, μπορεί να χειριστεί αποτελεσματικά μεγάλα σύνολα δεδομένων που απαιτούνται για την αποθήκευση συναλλαγών στη μνήμη. Με την χρήση της βιβλιοθήκης scikit-learn η εκτέλεση του αλγορίθμου γίνετε πιο αποδοτική.

Στον πίνακα 4.2 έχουν γίνει κάποιες πειραματικές μετρήσεις σε σχέση με την εκτέλεση του αλγορίθμου. Για το πείραμα ο αλγόριθμος εκτελέστηκε σε 6 σύνολα δεδομένων. Η μέθοδος του αγκώνα εκτελέστηκε για 20 συστάδες. Η συσταδοποίηση εκτελέστηκε με τον προτεινόμενο αριθμό συστάδων από την μέθοδο του αγκώνα. Από τον πίνακα μπορούμε να καταλάβουμε ότι οι χρόνοι εκτέλεσης εξάγονται με βάση το μέγεθος του αρχείου και το πλήθος των γραμμών που περιέχονται σε αυτό. Η μέθοδος του αγκώνα χρειάζεται περισσότερο χρόνο για να επιστρέψει το αποτέλεσμα καθώς ο αλγόριθμος θα τρέξει τόσες φορές όσες είναι ο αριθμός των συστάδων που θα επιλέξει ο χρήστης, σε αυτήν την περίπτωση 20, ώστε να επιστρέψει τα SSEs για αυτό το εύρος συστάδων.

Το σύνολο δεδομένων poker έχει μέγεθος 24MB και περιέχει πλήθος γραμμών πάνω από ένα εκατομμύ-

ριο και για αυτόν τον λόγο η μέθοδος του αγκώνα χρειάστηκε πάνω από 1 λεπτό και η συσταδοποίηση περίπου μισό λεπτό. Η εκτέλεση σε αυτό το σύνολο δεδομένων έγινε καθαρά και μόνο για πειραματικούς σκοπούς σε τοπικό υπολογιστή. Καθώς ο διακομιστής που φιλοξενείται η ιστοσελίδα έχει 4GB μνήμη RAM συνολικά για όλες τις υπηρεσίες που φιλοξενούνται. Η εκτέλεση για το συγκεκριμένο σύνολο δεδομένων χρειάστηκε πάνω από 2GB μνήμης RAM από τον τοπικό υπολογιστή. Για τον λόγο αυτό υπάρχει ο περιορισμός του 1GB σε χρήση RAM ανά κλήση προς τα συγκεκριμένα endpoints. Επίσης το μέγεθος ενός αρχείου δεν μπορεί να ξεπερνάει τα 10MB.

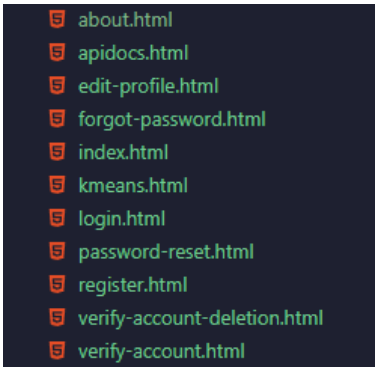
Πίνακας 4.2: Πειραματικές μετρήσεις

Dataset	**Size (Kb)**	**Γραμμές**	**Χρόνος για την μέθοδο του αγκώνα (κ=20)**	**Προτεινό-μενο κ**	**Χρόνος για την ανάθεση συστάδων**
penbased	538	10,992	6.47s	6	3.14s
letter	716	20,000	7.58s	7	4.09s
magic	1,462	19,020	8.45s	5	5s
texture	1,495	5,500	8.2s	4	4.44s
shuttle	1,559	57,999	8.52s	5	4.56s
poker	24,563	1,025,009	84s	6	28.25s

4.5 Υλοποίηση του Front-end

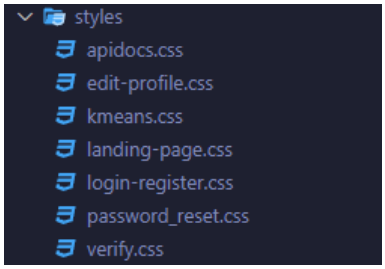
Το Front-end, δηλαδή η ιστοσελίδα της εφαρμογής υλοποιήθηκε με τις τεχνολογίες HTML, CSS, JavaScript, jQuery , Bootstrap. Οι τεχνολογίες αυτές αναλαμβάνουν την σχεδίαση της διεπαφής χρήστη, ώστε οι χρήστες να μπορούν να προβάλλουν και να αλληλεπιδρούν με αυτόν τον ιστότοπο.

Η HTML χρησιμοποιήθηκε για την ανάπτυξη της δομής και του περιεχομένου της ιστοσελίδας. Δηλαδή το κείμενο, τα διάφορα κουμπιά (buttons), τα γραφήματα κ.α. Με την χρήση της CSS μπορέσαμε να δομήσουμε καλύτερα και να δώσουμε ένα πιο ωραίο “style” στα στοιχεία της HTML. Στο σχήμα 4.20 φαίνονται τα HTML αρχεία τα οποία είναι 11, κάθε HTML αρχείο αντιπροσωπεύει και μια διαφορετική διαδρομή (route) προς το domain του διακομιστή μας. Στην ουσία ο χρήστης μπορεί να περιηγηθεί στις περισσότερες από αυτές τις διαδρομές μέσω της διεπαφής, ενώ κάποιες άλλες χρησιμεύουν ως βοηθητικές προς τον χρήστη όταν κάνει κάποια επιβεβαίωση με το email.



Σχήμα 4.20: HTML αρχεία

Τα css αρχεία που φαίνονται στο σχήμα 4.21 συλλέγονται στον φάκελο styles και συνδέονται με ένα ή περισσότερα HTML αρχεία μέσω της σήμανσης link και βάζοντας την διαδρομή προς το css αρχείο.



Σχήμα 4.21: CSS αρχεία

Όπως αναφέραμε και στο 2ο κεφάλαιο το framework Bootstrap χρησιμοποιείτε για να εξοικονομήσει χρόνο στους προγραμματιστές ως προς την σχεδίαση της ιστοσελίδας, καθώς περιέχει προκατασκευασμένα στοιχεία και κλάσεις. Κάποια από τα προκατασκευασμένα στοιχεία που χρησιμοποιήσαμε είναι τα modals. Τα Modal είναι κατασκευασμένα με HTML, CSS και JavaScript. Τοποθετούνται πάνω από οποιοδήποτε άλλο στοιχείο στο έγγραφο και αφαιρούν την κύλιση από την ιστοσελίδα έτσι ώστε μόνο το τοπικό περιεχόμενο να κάνει κύλιση. Με αυτόν τον τρόπο μπορούμε να κάνουμε τον χρήστη να εστιάσει μόνο σε αυτό το περιεχόμενο όταν πρόκειται για κάποιο σημαντικό μήνυμα. Για παράδειγμα όταν έχει πάει στον χρήστη ενα email επιβεβαίωσης, ή όταν θέλει η διεπαφή να πάρει μια επιβεβαίωση σχετικά με κάποια ενέργεια του χρήστη. Επίσης χρησιμοποιήθηκαν κάποιες από τις έτοιμες κλάσεις που παρέχει το Bootstrap, όπως τα rows και τα columns. Με αυτόν τον τρόπο η ιστοσελίδα είναι “full responsive”. Αυτό σημαίνει ότι η διάταξη και το περιεχόμενο της ιστοσελίδας προσαρμόζεται και αναδιοργανώνεται δυναμικά με βάση το μέγεθος της οθόνης στην οποία προβάλλεται, είτε πρόκειται για επιτραπέζιο υπολογιστή, φορητό υπολογιστή, tablet ή smartphone. Αυτή η προσέγγιση στο σχεδιασμό ιστοσελίδων βελτιώνει την εμπειρία του χρήστη, καθιστώντας ευκολότερη την πλοήγηση και την αλληλεπίδραση με τον ιστότοπο σε οποιαδήποτε συσκευή.

Με την χρήση της JavaScript και της βιβλιοθήκης jQuery μπορέσαμε να κάνουμε την ιστοσελίδα πιο δυναμική ως πρός το περιεχόμενο, καθώς και προς την αλληλεπίδραση με τον χρήστη. Για παράδειγμα εμφανίζουμε δυναμικά περιεχόμενο ανάλογο με τις ενέργειες που κάνει ο χρήστης, όπως error ή απαγορευτικά μηνύματα σε περίπτωση που ο χρήστης κάνει κάτι λάθος στις διάφορες φόρμες που υπάρχουν. Με την χρήση της jQuery μπορούμε να κάνουμε κλήσεις πρός το REST API της εφαρμογής που αναπτύξαμε και αναλύσαμε σε προηγούμενη ενότητα. Αυτό γίνετε με την βοήθεια του AJAX. Το AJAX (Asynchronous JavaScript και XML) είναι μια τεχνική προγραμματισμού που χρησιμοποιείται για τη δημιουργία γρήγορων και δυναμικών ιστοσελίδων. Επιτρέπει σε έναν ιστότοπο να ενημερώνει δυναμικά το περιεχόμενό του χωρίς να φορτώνει ξανά ολόκληρη τη σελίδα, καθιστώντας τον πιο φιλικό προς τον χρήστη. Το AJAX λειτουργεί με την αποστολή και τη λήψη δεδομένων από έναν διακομιστή ασύγχρονα (στο παρασκήνιο) χρησιμοποιώντας JavaScript και XML, έτσι ώστε μόνο τμήματα της ιστοσελίδας να ενημερώνονται αντί για ολόκληρη τη σελίδα. Αυτό έχει ως αποτέλεσμα μια πιο γρήγορη και πιο ομαλή εμπειρία, καθώς οι χρήστες μπορούν να εκτελούν ενέργειες και να λαμβάνουν περιεχόμενο χωρίς να χρειάζεται να περιμένουν για πλήρη επαναφόρτωση της σελίδας. Στο σχήμα 4.22 μπορούμε να δούμε ένα παράδειγμα κλήσης προς το REST API μας. Πιο συγκεκριμένα όταν ο χρήστης επιλέξει για να δεί κάποιο dataset θα γίνει κλήση πρός το endpoint read-dataset. Εφόσον η μέθοδος είναι η GET οι παράμετροι μπαίνουν μέσα στο URI πρός το endpoint.

```
$.ajax({url: `./server/api/read-dataset.php?dataset=${dataset}&dataset-type=${id.toLowerCase()}&apikey=${apikey}`,
  method: 'GET',
  dataType: 'json',
  success: function(data) {
```

Σχήμα 4.22: Κώδικας για κλήση πρός το API

Στην συνέχεια η AJAX μας παρέχει τις μεθόδους success και error, ώστε να διαχειριστούμε το περιεχόμενο ανάλογα με αυτό που μας επέστρεψε ο διακομιστής. Στην περίπτωση του success, σημαίνει ότι τα δεδομένα ήρθαν με επιτυχία. Στο συγκεκριμένο παράδειγμα που αναλύουμε στην περίπτωση του success όπως φαίνεται στο σχήμα 4.23 δημιουργούμε έναν HTML πίνακα (table). Στην συνέχεια διαβάζουμε τις κεφαλίδες (headers) των δεδομένων, ώστε να τα περάσουμε στον πίνακα μέσω των HTML στοιχείων <th>. Στην συνέχεια γεμίζουμε τον πίνακα με τα δεδομένα με την μέθοδο bootstrapTable.

```
success: function(data) {
  tableLength=Object.keys(data.items).length;
  console.log(tableLength)
 $('.cont-dataset-table>.card>.card-body').append(table);
 $('.cont-dataset-table').show();
 keys=Object.keys(data.items[0]);
 $.each(keys,(i,key)=>{
  $('#dataset-table>thead>tr').append(`<th data-field=${key}>${key}</th>`)
})
 $('#dataset-table').bootstrapTable({
  data:data.items,
  reinit: true,
  // height:500
 })
```

Σχήμα 4.23: Κώδικας στην περίπτωση success

Στην περίπτωση του error που σημαίνει ότι τα δεδομένα δεν έφτασαν με επιτυχία λόγο κάποιου σφάλματος που επέστρεψε ο διακομιστής. Στην συνέχεια του παραδείγματος μας στην περίπτωση του error, όπως φαίνεται στο σχήμα 4.24, ελέγχουμε αν ο κωδικός του σφάλματος είναι 401 που σημαίνει “unauthorized” και παραπέμπουμε τον χρήστη να κάνει πάλι σύνδεση (login) στην εφαρμογή. Αν δεν ισχύει αυτό εμφανίζουμε το μήνυμα σφάλματος στην διεπαφή χρήστη.

```
error:function(xhr, status, error) {
 if(status=="401"){
  $('#loading-spinner-table').hide();
  sessionStorage.clear();
  window.location.href = "./login.html";
 }
 else{
  $('#error-text-big').text(xhr.responseText);
  $('.error-message-big').show();
 }
```

Σχήμα 4.24: Κώδικας στην περίπτωση error

Όταν ένας χρήστης κάνει σύνδεση στην εφαρμογή τα στοιχεία του αποθηκεύονται στο session storage του προγράμματος περιήγησης (browser). Αυτά είναι το ονοματεπώνυμο του, το email του και το API KEY (Σχήμα 4.25). Αυτό μας βοηθάει να κρατάμε τον χρήστη συνδεδεμένο στην εφαρμογή, χρησιμοποιώντας αυτά τα στοιχεία για να κάνουμε τις κλήσεις προς το REST API μας.

```
59          sessionStorage.setItem("apikey", data.apiKey);
60          sessionStorage.setItem("email", data.email);
61          sessionStorage.setItem("fname", data.fname);
62          sessionStorage.setItem("lname", data.lname);
```

Σχήμα 4.25: Κώδικας για χρήση του sessionStorage

4.6 Επίλογος

Σε αυτό το κεφάλαιο παρουσιάσαμε την υλοποίηση της εφαρμογής,με διαγράμματα και κομμάτια κώδικα ώστε να καταλάβουμε την λειτουργία της. Τον πλήρες κώδικα για το back-end και το front-end μπορείτε να τα βρείτε στον σύνδεσμο `https://github.com/KostisGrf/WebKmeans`

Κεφάλαιο 5ο: Παρουσίαση του REST API

Στο κεφάλαιο αυτό θα γίνει παρουσίαση του REST API. Θα αναλυθούν ολα τα endpoint με παραδείγματα κλήσης και απόκρισης.

Στον πίνακα 5.1 παρουσιάζονται τα endpoints που μπορεί να χρησιμοποιήσει κάποιος στην εφαρμογή του ώστε να κάνει κλήση στο REST API μας.

Πίνακας 5.1: Τα endpoints του REST API

Endpoint	Τύπος request	Περιγραφή
/server/api/get-datasets.php	GET	Επιστρέφει όλα τα dataset, public και personal
/server/api/read-dataset.php	GET	Επιστρέφει τα δεδομένα ενός dataset
/server/api/elbow.php	POST	Επιστρέφει τα τετραγωνικά λάθη(SSEs) για την δημιουργία του elbow chart
/server/api/clusters.php	POST	Αναθέτει σε συστάδες κάθε γραμμή ενός dataset
/server/api/upload-dataset.php	POST	Ανεβάζει στον διακομιστή ένα dataset
/server/api/delete-dataset.php	DELETE	Διαγράφει ένα dataset
/server/api/register.php	POST	Εγγραφή χρήστη
/server/api/login.php	POST	Σύνδεση ενός χρήστη
/server/api/edit-profile.php	POST	Αλλάζει τα στοιχεία του χρήστη
/server/api/delete-user.php	DELETE	Διαγράφει έναν χρήστη

5.1 Μέθοδοι GET

/server/api/get-datasets.php
Το /server/api/get-datasets.php?apikey=usersApiKey παίρνει ως παράμετρο το APIKEY του χρήστη και επιστρέφει όλα τα dataset που έχει ανεβάσει ο χρήστης και τα dataset που είναι δημόσια προς όλους τους χρήστες.

Παράδειγμα response:

```
{
   "personal_datasets": [
    "datatab.xlsx",
    "file_example_XLSX_100.xlsx",
    "file_example_XLS_50.xls",
    "fuel_prices_52.csv",
    "iris.csv",
    "test.xlsx"
       ],
    "public_datasets": [
        "iris.csv"
        ]
}
```

/server/api/read-dataset.php

Το /server/api/read-dataset.php?apikey=usersApiKeydataset=iris.csvdataset-type=public||personal παίρνει σαν παραμέτρους το API KEY του χρήστη, το αρχείο dataset που θέλει να διαβάσει και τον τύπο του dataset δηλαδή personal σε περίπτωση που είναι προσωπικό του χρήστη ή public στην περίπτωση που είναι δημόσιο προς όλους τους χρήστες. Επιστρέφει τα δεδομένα του dataset καθώς και τις αριθμητικές στήλες στις οποίες μπορεί ο χρήστης να εφαρμόσει την συσταδοποίηση.

Παράδειγμα response:


```
{
  "items": [
   {
     "Gender": "Male",
     "Age": 48,
     "Salary": 1000
   },
   {
     "Gender": "Male",
     "Age": 33,
     "Salary": 1500
   },
   {
     "Gender": "Female",
     "Age": 29,
     "Salary": 2000
    },
    {
      "Gender": "Male",
      "Age": 52,
      "Salary": 3500
    },
 ],
   "numerical_columns": [
     "Age",
     "Salary"
   ]
}
```

5.2 Μέθοδοι POST

/server/api/elbow.php
Επιστρέφει τα τετραγωνικά λάθη SSEs και ένα προτεινόμενο κ (αριθμός συστάδων). Ο χρήστης πρέπει να δώσει τις παραμέτρους μέσα στο body αυτήν την φορά και όχι στο URI. Οι παραμέτροι που πρέπει να δώσει είναι το dataset, dataset-type , ApiKey , clusters το οποίο είναι το εύρος των cluster για το οποίο θα τρέξει ο αλγόριθμος και τα columns το οποίο είναι οι στήλες για τις οποίες θα τρέξει ο αλγόριθμος. Οι στήλες που θα δώσει πρέπει να έχουν αριθμητικό περιεχόμενο.

Παράδειγμα request:

```
{
  "dataset":"sample.csv",
  "dataset-type":"personal",
  "clusters":"5",
  "columns":["Age","Salary"],
  "apikey":"0a8366a07c0d8fccc48bab2e657f12d0"
}
```

Παράδειγμα response:

```
{
  "sse": [
  "1.2898762444864522",
  "0.5491175803402647",
  "0.23902558286074355",
  "0.05170359168241964",
  "0.016580718336483915"
 ],
 "suggested-k": "2"
}
```

/server/api/clusters.php
Αναθέτει σε συστάδες κάθε γραμμή ενός dataset.

Παράδειγμα request:

```
{
  "dataset":"sample.csv",
  "dataset-type":"personal",
  "clusters":"3",
  "columns":["Age","Salary"]
  "apikey":"0a8366a07c0d8fccc48bab2e657f12d0"
}
```

Παράδειγμα response:

```
{
  "items": [
    {
      "Age": "48",
      "Salary": "1000",
      "cluster": "1"
    },
    {
      "Age": "33",
      "Salary": "1500",
      "cluster": "1"
    },
    {
      "Age": "29",
      "Salary": "2000",
      "cluster": "2"
    },
    {
      "Age": "52",
      "Salary": "3500",
      "cluster": "3"
    },

 ]
}
```

/server/api/upload-dataset.php

Ανεβάζει στον διακομιστή ένα dataset.

Παράδειγμα request:

```
{
  form-data:
    dataset:{file},
    dataset-type:{personal|public},
    apikey:{usersApiKey}
}
```

Παράδειγμα response:

```
{ "message"=>"file uploaded."}
```

/server/api/register.php

Εγγραφή χρήστη.

Παράδειγμα request:

```
{
  "email":"example@email.com",
  "password":"verysecret123",
  "fname":"Konstantinos",
  "lname":"Hliadhs"
}
```

Παράδειγμα response:

```
{ "message"=>"User registered."}
```

/server/api/login.php

Σύνδεση ενός χρήστη.

Παράδειγμα request:

```
{
 "email":"example@email.com",
 "password":"verysecret123"
}
```

Παράδειγμα response:

```
{
  "email":"example@email.com",
  "fname":"Konstantinos",
  "lname":"Hliadhs",
  "apikey":"0a8366a07c0d8fccc48bab2e657f12d0"
}
```

/server/api/edit-profile.php

Αλλάζει τα στοιχεία του χρήστη.

Παράδειγμα request:

```
{
  "fname":"Kostas",
  "password":"verysecret12345",
  "apikey":"0a8366a07c0d8fccc48bab2e657f12d0"
}
```

5.3 Μέθοδοι DELETE

/server/api/delete-dataset.php
Διαγράφει ένα dataset.

Παράδειγμα request:

```
{
  "dataset":"sample.csv",
  "dataset-type":"personal",
  "apikey":"0a8366a07c0d8fccc48bab2e657f12d0"
}
```

Παράδειγμα response:

```
{ "message"=>"dataset deleted."}
```

/server/api/delete-user.php
Διαγράφει έναν χρήστη.

Παράδειγμα request:

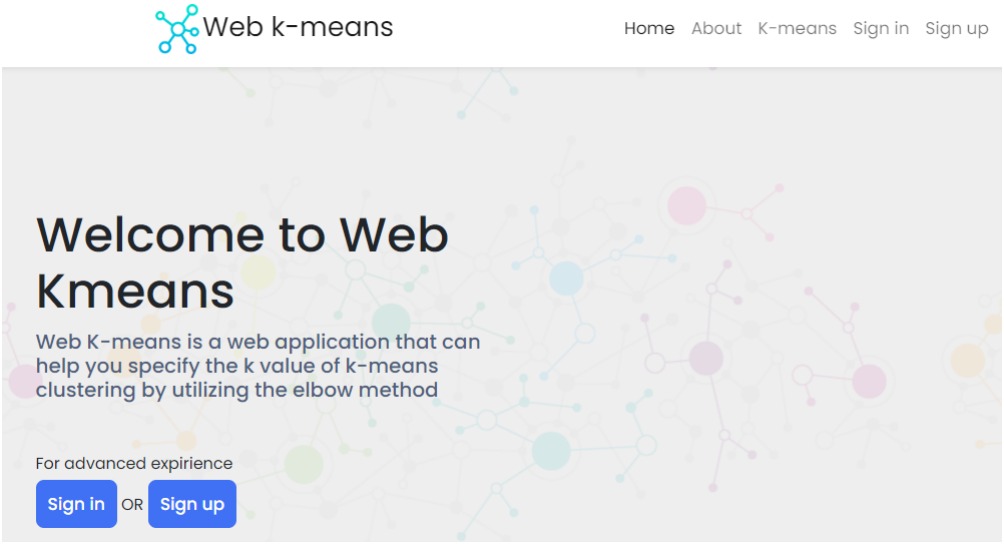
```
 {"apikey":"0a8366a07c0d8fccc48bab2e657f12d0"}
```

Κεφάλαιο 6ο: Παρουσίαση του web k-means

Στο κεφάλαιο αυτό θα γίνει παρουσίαση της διεπαφής του χρήστη της εφαρμογής, δηλαδή της ιστοσελίδας.

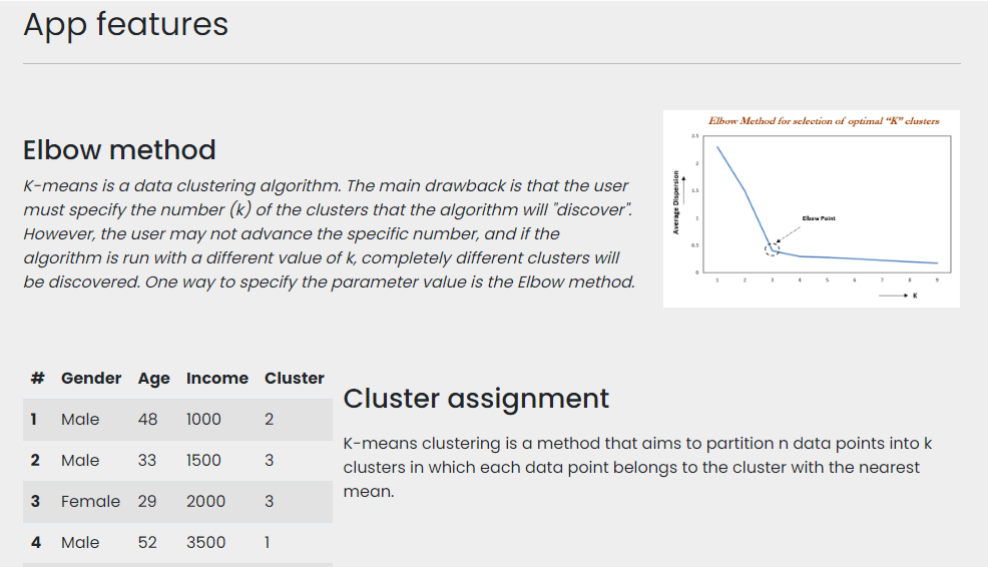
6.1 Αρχική σελίδα

Όταν ένας χρήστης μπαίνει στη ιστοσελίδα το πρώτο πράγμα που βλέπει είναι η αρχική σελίδα η αλλιώς “landing page”. Στο πάνω μέρος της ιστοσελίδας που φαίνεται στο σχήμα 6.1 υπάρχει ένα μήνυμα που καλωσορίζει τον χρήστη και έχει μια πρώτη περιγραφή ώστε να καταλάβει περί τίνος πρόκειται η εφαρμογή. Στην συνέχεια υπάρχουν δύο κουμπιά (buttons) τα οποία παροτρύνουν τον χρήστη να πραγματοποιήσει σύνδεση στην εφαρμογή ή να δημιουργήσει καινούργιο λογαριασμό ώστε να έχει πρόσβαση σε όλες τις λειτουργίες της εφαρμογής.



Σχήμα 6.1: Πάνω μέρος της αρχικής σελίδας

Στην συνέχεια στο κάτω μέρος της σελίδας που φαίνεται στο σχήμα 6.2 περιγράφονται οι λειτουργίες της εφαρμογής, πιο συγκεκριμένα της μεθόδου του αγκώνα και της συσταδοποίησης.

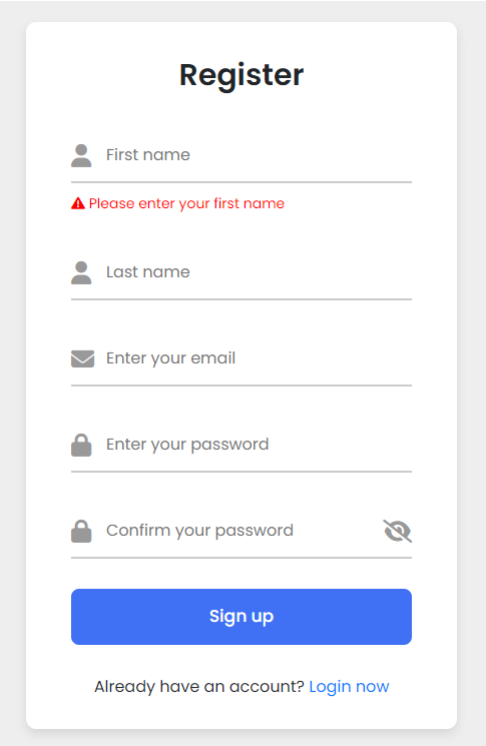


Σχήμα 6.2: Κάτω μέρος της αρχικής σελίδας

6.2 Φόρμες της εφαρμογής

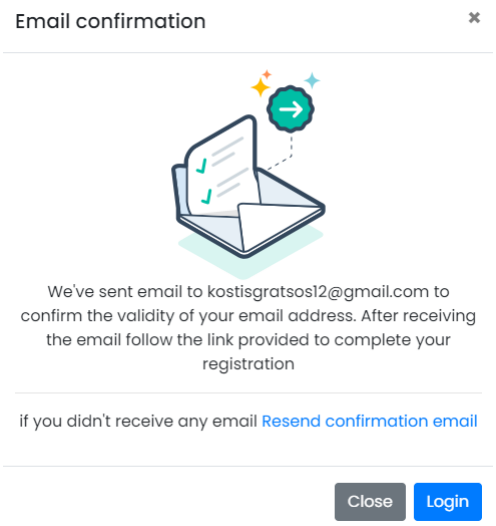
Φόρμα Register

Όπως αναφέραμε και παραπάνω, ο χρήστης για να μπορέσει να έχει πρόσβαση σε όλες τις λειτουργίες της εφαρμογής θα πρέπει να δημιουργήσει λογαριασμό και να συνδεθεί σε αυτόν. Στο σχήμα 6.3 φαίνεται η φόρμα που πρέπει να συμπληρώσει ο χρήστης. Θα πρέπει να συμπληρώσει όνομα, επίθετο, email, κωδικό πρόσβασης και να επιβεβαιώσει τον κωδικό του. Σε περίπτωση που ο χρήστης δεν συμπληρώσει σωστά κάποια από τα πεδία εμφανίζεται κάτω από το αντίστοιχο πεδίο μήνυμα λάθους. Τα λάθη που μπορεί να κάνει είναι να αφήσει κενό ένα πεδίο, το email που συμπλήρωσε να είναι σε λάθος μορφή, να έβαλε λιγότερους από 8 αλφαριθμητικούς χαρακτήρες στον κωδικό πρόσβασης ή να μην ταιριάζουν τα πεδία του κωδικού και της επιβεβαίωσης.



Σχήμα 6.3: Φόρμα register

Στήν περίπτωση που ο χρήστης συμπληρώσει σωστά όλα τα πεδία και πατήσει το button “sign up”, θα εμφανιστεί ένα μήνυμα όπου θα αναφέρει στον χρήστη ότι έχει λάβει email επαλήθευσης (Σχήμα 6.4).

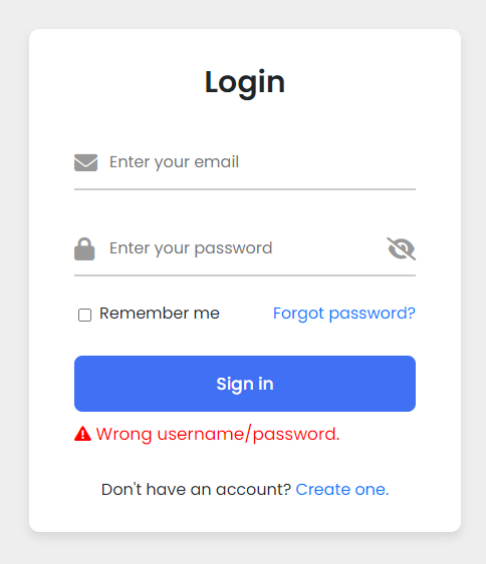


Σχήμα 6.4: Προειδοποίηση για email επαλήθευσης

Φόρμα Login

Στο σχήμα 6.5 φαίνεται η φόρμα που πρέπει να συμπληρώσει ο χρήστης ώστε να συνδεθεί στην εφαρμογή. Θα πρέπει να συμπληρώσει το email με το οποίο έκανε εγγραφή και τον κωδικό πρόσβασης του. Στο πεδίο του κωδικού στην δεξιά μεριά υπάρχει ένα εικονίδιο που μοιάζει με μάτι, σε περίπτωση που το πατήσει ο χρήστης θα εμφανιστεί ο κωδικός, αν το ξαναπατήσει θα γυρίσει πάλι στην μορφή με τις τελείες. Από κάτω υπάρχει ένα link “forgot password” σε περίπτωση που ο χρήστης ξεχάσει τον κωδικό

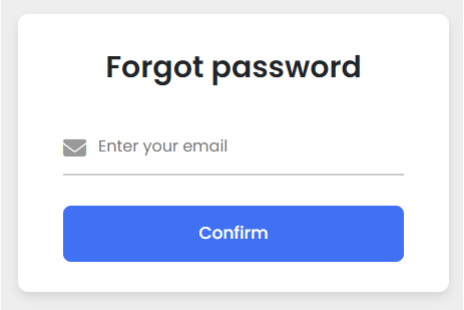
του (Θα το δούμε παρακάτω). Πατώντας το κουμπί “Sign in” αν τα στοιχεία που έδωσε ο χρήστης είναι σωστά θα μεταφερθεί στην αρχική σελίδα και πλέον θα έχουν εξαφανιστεί τα buttons που προτρέπουν τον χρήστη να κάνει σύνδεση ή εγγραφή. Σε περίπτωση που ο χρήστης δώσει λάθος στοιχεία θα εμφανιστεί ένα μήνυμα με κόκκινα γράμματα όπως φαίνεται στο σχήμα 6.5.



Σχήμα 6.5: Φόρμα login

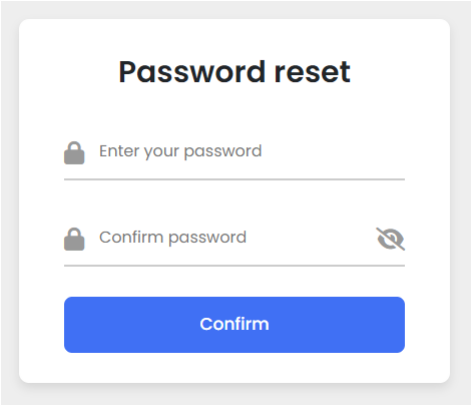
Φόρμες για το forgot password

Σε περίπτωση που ο χρήστης ξέχασε τον κωδικό του μπορεί να πατήσει τον σύνδεσμο “forgot password” και θα του εμφανιστεί η φόρμα που φαίνεται στο σχήμα 6.6.



Σχήμα 6.6: Φόρμα forgot password

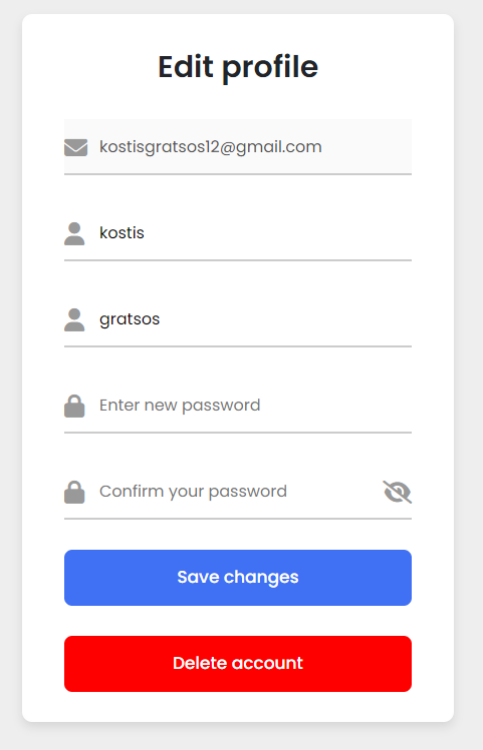
Αφού συμπληρώσει το email του στην φόρμα θα λάβει ένα μήνυμα στο email του, το οποίο θα περιέχει έναν σύνδεσμο για να μπορέσει να ορίσει καινούργιο κωδικό πρόσβασης (Σχήμα 6.7).



Σχήμα 6.7: Φόρμα για ορισμό νέου κωδικού πρόσβασης

Φόρμα edit profile

Εφόσον ο χρήστης έχει κάνει σύνδεση στην εφαρμογή μπορεί να έχει πρόσβαση στην σελίδα edit-profile. Στην σελίδα edit-profile όπως φαίνεται στο σχήμα 6.8 εμφανίζεται μία φόρμα όπου ο χρήστης μπορεί να αλλάξει τα στοιχεία του, για παράδειγμα το όνομα, επίθετο ή να αλλάξει τον κωδικό πρόσβασης του. Μπορεί επίσης να διαγράψει τον λογαριασμό του μετά από μήνυμα επιβεβαίωσης που του λαμβάνει στο email του.

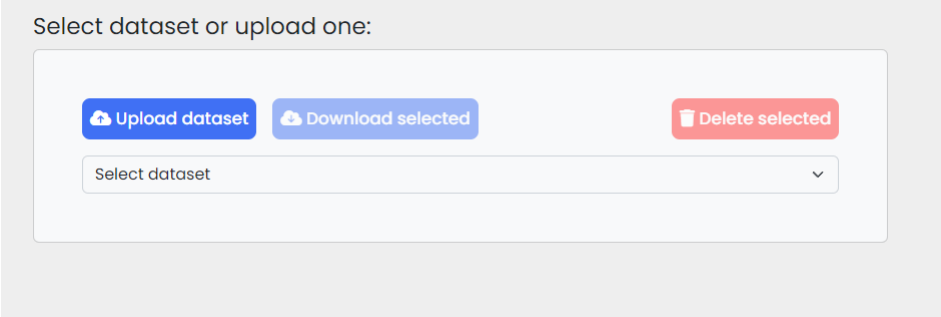


Σχήμα 6.8: Φόρμα για ορισμό νέου κωδικού πρόσβασης

6.3 Σελίδα k-means

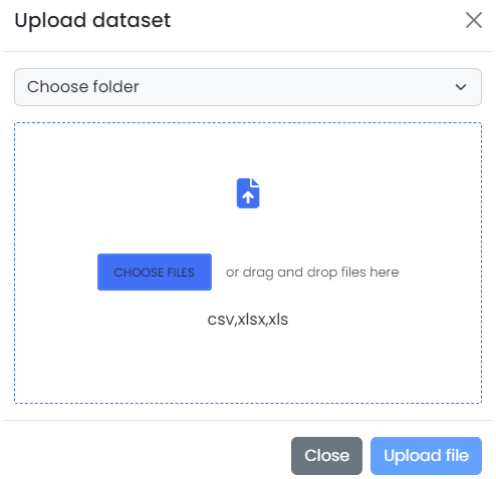
Εφόσον ο χρήστης είναι συνδεδεμένος στην εφαρμογή μπορεί να έχει πρόσβαση στην σελίδα k-means. Όταν ένας χρήστης μπαίνει σε αυτήν την σελίδα όπως φαίνεται και στο σχήμα 6.9, το μόνο πράγμα που του εμφανίζεται αρχικά είναι μια φόρμα που τον παροτρύνει είτε να επιλέξει ένα σύνολο δεδομένων

(dataset) είτε να ανεβάσει ένα. Σε περίπτωση που ο χρήστης κάνει κλικ στον επιλογέα (selector) που γράφει "select dataset" του εμφανίζεται μια λίστα με τα σύνολα δεδομένων που έχει ανεβάσει ο ίδιος και τα δημόσια σύνολα δεδομένων που ανεβάζουν άλλοι χρήστες σε αυτήν την κατηγορία. Σε περίπτωση που πατήσει το button "download selected" μπορεί να κατεβάσει το dataset που έχει επιλέξει. Αν πατήσει το button "delete selected" μπορεί να διαγράψει το dataset που έχει επιλέξει, αν το dataset που έχει επιλέξει είναι δημόσιο γίνετε έλεγχος αν ο χρήστης έχει άδεια να ανεβάσει και να διαγράψει δημόσια σύνολα δεδομένων και εμφανίζεται αντίστοιχο μήνυμα σφάλματος.



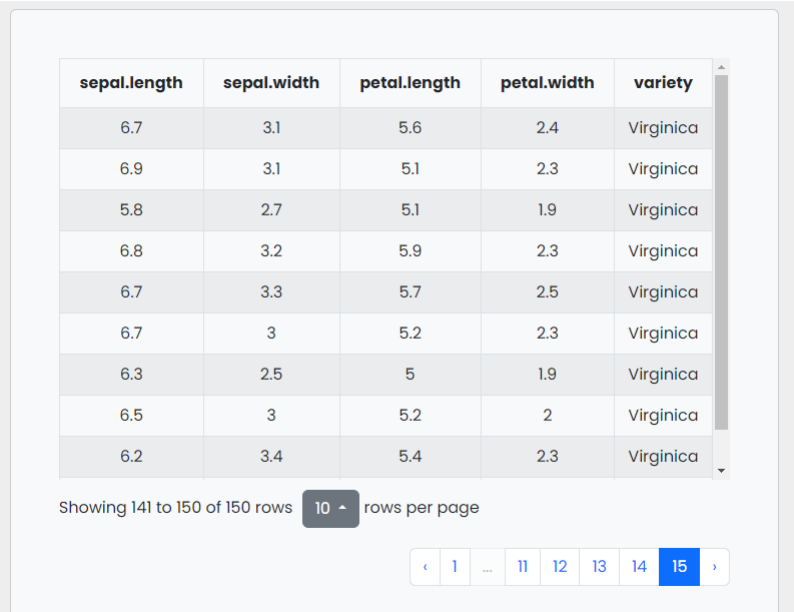
Σχήμα 6.9: Τμήμα επιλογής dataset

Στην περίπτωση που ο χρήστης πατήσει το button "upload dataset" του εμφανίζεται η φόρμα που φαίνεται στο σχήμα 6.10, στην οποία μπορεί να ανεβάσει ένα dataset στον διακομιστή μας. Για αρχή θα πρέπει να επιλέξει στον επιλογέα "select folder" τον φάκελο στον οποίο θέλει να ανεβάσει το dataset, δηλαδή στον προσωπικό του φάκελο (Personal) ή στον δημόσιο φάκελο (Public). Στην συνέχεια υπάρχει ένα πλαίσιο όπου μπορεί είτε να πατήσει το button "Choose files", ώστε να ανοίξει ο explorer του λειτουργικού συστήματος του, είτε να τραβήξει και να αφήσει (drag and drop) το αρχείο που θέλει μέσα στο πλαίσιο. Στην συνέχεια γίνετε έλεγχος αν το αρχείο που επέλεξε είναι τύπου csv ή xlsx ή xls.



Σχήμα 6.10: Φόρμα για ανέβασμα dataset

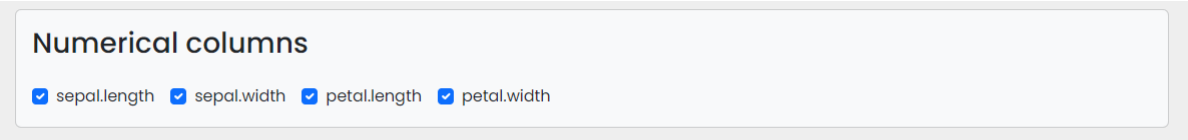
Όταν ο χρήστης επιλέξει ενα dataset στο επιλογέα "select dataset" στην σελίδα εμφανίζονται τέσσερα ακόμα τμήματα. Το πρώτο τμήμα που φαίνεται στο σχήμα 6.11 είναι ένας πίνακας με τα δεδομένα του dataset που έχει επιλέξει.



sepal.length	sepal.width	petal.length	petal.width	variety
6.7	3.1	5.6	2.4	Virginica
6.9	3.1	5.1	2.3	Virginica
5.8	2.7	5.1	1.9	Virginica
6.8	3.2	5.9	2.3	Virginica
6.7	3.3	5.7	2.5	Virginica
6.7	3	5.2	2.3	Virginica
6.3	2.5	5	1.9	Virginica
6.5	3	5.2	2	Virginica
6.2	3.4	5.4	2.3	Virginica

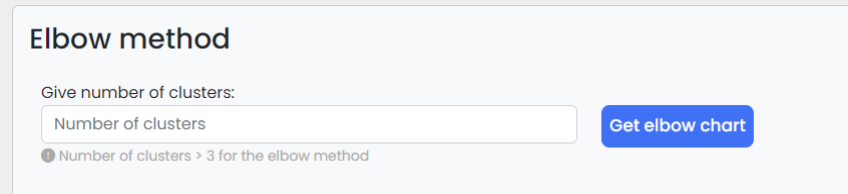
Σχήμα 6.11: Πίνακας με τα δεδομένα του dataset

Το δεύτερο τμήμα (Σχήμα 6.12) είναι η αριθμητικές στήλες του dataset τις οποίες μπορεί να επιλέξει ώστε να εκτελέσει τον αλγόριθμο k-means στα επόμενα τμήματα.



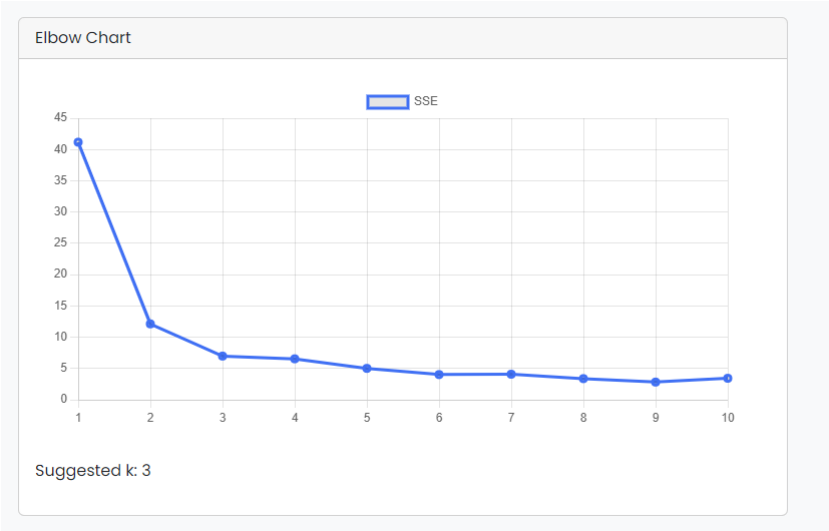
Σχήμα 6.12: Αριθμητικές στήλες του dataset

Το τρίτο τμήμα (Σχήμα 6.13) είναι για την εκτέλεση της μεθόδου του αγκώνα.



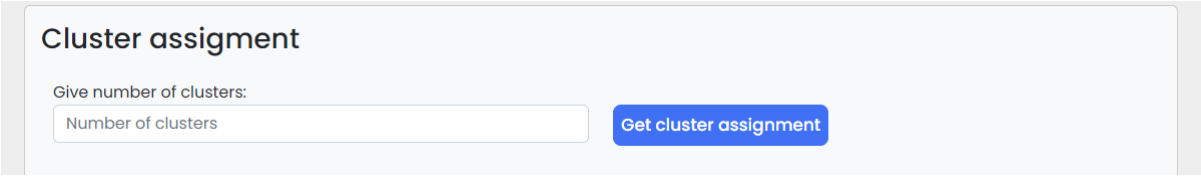
Σχήμα 6.13: Τμήμα για την εκτέλεση της μεθόδου του αγκώνα

Αφού ο χρήστης συμπληρώσει τον αριθμό συστάδων στο πεδίο “Number of clusters” και πατήσει το button “Get elbow chart” θα εμφανιστεί μετά από λίγα δευτερόλεπτα το γράφημα του αγκώνα (Σχήμα 6.14) με τα τετραγωνικά λάθη(SSEs) και μια προτεινόμενη τιμή για το κ (αριθμός συστάδων).



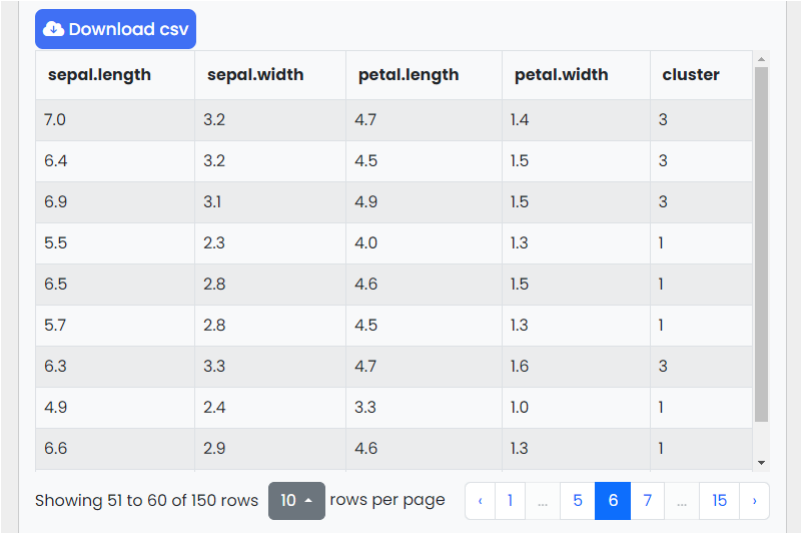
Σχήμα 6.14: Γράφημα για την μέθοδο του αγκώνα

Στο τέταρτο τμήμα (Σχήμα 6.15) ο χρήστης μπορεί να εκτελέσει την συσταδοποίηση για τις στήλες που έχει επιλέξει.



Σχήμα 6.15: Τμήμα για την εκτέλεση συσταδοποίησης κ-μέσων

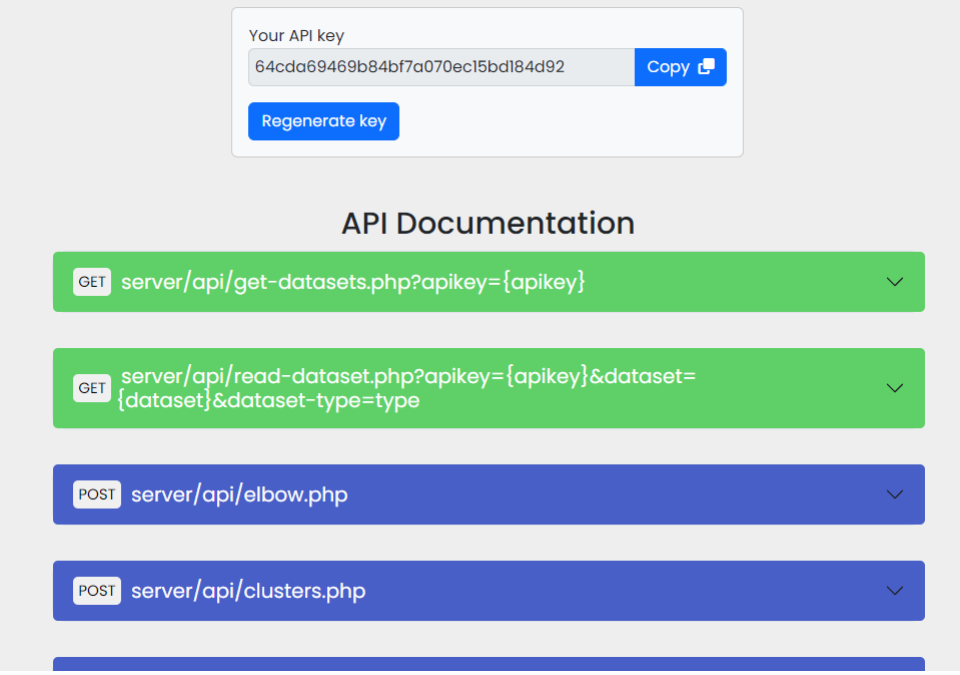
Αφού ο χρήστης συμπληρώσει τον αριθμό συστάδων στο πεδίο “Number of clusters” και πατήσει το button “Get cluster assignment” θα εμφανιστεί μετά από λίγα δευτερόλεπτα ένας πίνακας όπως φαίνεται στο σχήμα 6.16 με τις στήλες που έχει επιλέξει για συσταδοποίηση και μία στήλη με τις συστάδες που ανατέθηκαν για τις συγκεκριμένες στήλες. Επίσης υπάρχει ενα button “Download csv” με το οποίο ο χρήστης μπορεί να κατεβάσει τον πίνακα σε μορφή csv.



Σχήμα 6.16: Πίνακας με τα δεδομένα συστοδοποίσης

6.4 Σελίδα API Docs

Στην σελίδα API Docs ο χρήστης εφόσον είναι συνδεδεμένος, όπως φαίνεται στο σχήμα 6.17 μπορεί να δεί το API KEY του και να το κάνει αντιγραφή.



Σχήμα 6.17: Σελίδα API Docs

Στην συνέχεια από κάτω εμφανίζονται τα endpoint του REST API, κάνοντας κλίκ σε αυτά εμφανίζεται ένα τμήμα με οδηγίες και παραδείγματα κλήσης και ανταπόκρισης από το API (Σχήμα 6.18)

Σχήμα 6.18: Οδηγίες προς κάποιο endpoint

6.5 Επίλογος

Στο κεφάλαιο αυτό έγινε παρουσίαση της ιστοσελίδας της εφαρμογής. Μπορείτε να δοκιμάσετε την live ιστοσελίδα στον σύνδεσμο `https://nireas.iee.ihu.gr/webkmeans/`

Κεφάλαιο 7ο: Συμπεράσματα και μελλοντικές επεκτάσεις

7.1 Συμπεράσματα

Με τον όρο συσταδοποίηση (clustering) περιγράφουμε τον διαχωρισμό των δεδομένων σε ομάδες παρόμοιων χαρακτηριστικών. Μπορεί να επιτευχθεί με διάφορους αλγόριθμους που διαφέρουν σημαντικά ως προς την κατανόησή τους. Μετά από ερεύνα εντοπίσαμε ότι δεν υπάρχουν πολλές διαδικτυακές εφαρμογές για συσταδοποίηση δεδομένων που να είναι ανοιχτές στην παγκόσμια επιστημονική κοινότητα, και οι υπάρχουσες έχουν αρκετά περιορισμένες δυνατότητες. Στην web εφαρμογή που αναπτύχθηκε για την πτυχιακή αυτή, κάθε χρήστης μπορεί να ανεβάσει σύνολα δεδομένων και η εφαρμογή κατασκευάζει το γράφημα όπου παρουσιάζεται ο αγκώνας και προτείνει στον χρήστη πιθανή τιμή για την παράμετρο κ. Στη συνέχεια, ο χρήστης μπορεί να κάνει την συσταδοποίηση με το προτεινόμενο κ, καθώς και να κατεβάσει το γράφημα ή το σύνολο δεδομένων οπού θα αναφέρετε η συστάδα που ανατέθηκε σε κάθε αντικείμενο του συνόλου δεδομένων.

7.2 Μελλοντικές επεκτάσεις

Παρόλο που η πτυχιακή αποτελεί μια πλήρη εφαρμογή για τους ενδιαφερόμενους που θέλουν να εκτελέσουν συσταδοποίηση με τον αλγόριθμο κ-μέσων, θα μπορούσαν να πραγματοποιηθούν κάποιες επεκτάσεις ώστε να γίνει ακόμα καλύτερη η εφαρμογή.

Scatter plot

Το scatter plot στην συσταδοποίηση είναι ένας τύπος γραφήματος που χρησιμοποιείται για την οπτικοποίηση της κατανομής των στιγμιοτύπων σε έναν δισδιάστατο χώρο. Τα στιγμιότυπα αντιπροσωπεύονται ως μεμονωμένες κουκκίδες στο γράφημα, με τη θέση κάθε κουκκίδας να καθορίζεται από τις τιμές της για δύο επιλεγμένα χαρακτηριστικά. Το scatter μπορεί να χρησιμοποιηθεί με διαφορετικά χρώματα ή σύμβολα που χρησιμοποιούνται για να υποδείξουν ποια στιγμιότυπα ανήκουν στην ίδια συστάδα.

Αποθήκευση αποτελεσμάτων και αποστολή τους με email

Σε περίπτωση που κάποιος χρήστης δώσει μεγάλο αριθμό συστάδων σε μεγάλο σύνολο δεδομένων (για παράδειγμα 1.000.000 γραμμές) θα πρέπει να περιμένει αρκετή ώρα ώστε να πάρει τα αποτελέσματα είτε της μεθόδου του αγκώνα είτε της συσταδοποίησης. Μια επέκταση θα μπορούσε να είναι, όταν αργούν να εμφανιστούν τα αποτελέσματα κάποιο χρονικό διάστημα θα μπορούν να αποθηκευτούν τα αποτελέσματα στον διακομιστή. Όταν τελειώσει ο αλγόριθμος ο χρήστης θα λαμβάνει μήνυμα στο email του που θα τον ενημερώνει ότι τα αποτελέσματα είναι έτοιμα. Στην συνέχεια θα μπορεί να μπεί στην ιστοσελίδα και να τα δεί.

Pagination

Το pagination στον προγραμματισμό ιστού αναφέρεται στη διαδικασία διαίρεσης ενός μεγάλου συνόλου δεδομένων σε μικρότερα, πιο διαχειρίσιμα κομμάτια ή "σελίδες" για εμφάνιση σε έναν ιστότοπο. Αυτό χρησιμοποιείται συνήθως για λίστες, πίνακες και άλλους τύπους δεδομένων που μπορούν να καθυστερήσουν την εμφάνιση των αποτελεσμάτων. Η ιδέα είναι να χωριστούν τα δεδομένα σε σελίδες, επιτρέποντας στους χρήστες να προβάλλουν ένα μικρότερο σύνολο δεδομένων κάθε φορά και να πλοηγούνται μεταξύ των σελίδων για να προβάλλουν άλλα μέρη των δεδομένων. Στην δικιά μας εφαρμογή

αυτό μπορεί να γίνει όταν ένας χρήστης ζητάει να δεί τα δεδομένα ενός dataset. Αρχικά αυτό θα πρέπει να γίνει στο back-end ώστε να επιστρέφονται τα αποτελέσματα σε "σελίδες". Τα πόσα αποτελέσματα θα εμφανίζονται σε κάθε σελίδα θα το δίνει ο χρήστης.

ΒΙΒΛΙΟΓΡΑΦΙΑ

[1] K. D. Bailey, "Cluster analysis," *Sociological Methodology*, vol. 6, pp. 59–128, 1975.

[2] "Supervised vs. unsupervised learning." `https://www.alteryx.com/glossary/supervised-vs-unsupervised-learning`, 2022.

[3] B. Everitt, S. Landau, M. Leese, D. Stahl, and a. O. M. C. Safari, *Cluster Analysis, 5th Edition*. John Wiley & Sons, 2011.

[4] V. Estivill-Castro, "Why so many clustering algorithms: a position paper," *SIGKDD Explor.*, vol. 4, pp. 65–75, 2002.

[5] R. J. G. B. Campello, P. Kröger, J. Sander, and A. Zimek, "Density-based clustering," *WIREs Data Mining and Knowledge Discovery*, vol. 10, no. 2, p. e1343, 2020.

[6] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, KDD'96, p. 226–231, AAAI Press, 1996.

[7] S. Äyrämö and T. Kärkkäinen, "Introduction to partitioning-based clustering methods with a robust example," *Reports of the Department of Mathematical Information Technology. Series C, Software engineering and computational intelligence*, no. 1/2006, 2006.

[8] Divyanshu Anand, "Partitional Clustering." `https://medium.com/analytics-vidhya/partitional-clustering`, 2020.

[9] J. Hartigan, "Distribution problems in clustering," in *Classification and Clustering* (J. Van Ryzin, ed.), pp. 45–71, Academic Press, 1977.

[10] G. Seif, "The 5 clustering algorithms data scientists need to know," Feb 2022.

[11] "Weka the data platform for ai." `https://www.weka.io/`.

[12] "What is MATLAB?." https://www.mathworks.com/discovery/what-is-matlab.html. [Online; accessed 2023-01-17].

[13] J. MacQueen, "K-means clustering," *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability*, 1967.

[14] R. Jain and R. Dubes, *A Comparative Study of Clustering Algorithms*. Prentice-Hall, Inc., 1988.

[15] X. Wu and Y. Kumar, "A survey of clustering algorithms," *IEEE Transactions on Neural Networks*, vol. 19, no. 11, pp. 1505–1530, 2008.

[16] P. J. Rousseeuw, "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis," *Journal of computational and applied mathematics*, vol. 20, pp. 53–65, 1987.

[17] R. Tibshirani, G. Walther, and T. Hastie, "Estimating the number of clusters in a data set via the gap statistic," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 63, no. 2, pp. 411–423, 2001.

[18] H. Akaike, "A new look at the statistical model identification," *IEEE Transactions on Automatic Control*, vol. 19, no. 6, pp. 716–723, 1974.

[19] D. Arthur and S. Vassilvitskii, "k-means++: The advantages of careful seeding," in *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pp. 1027–1035, 2007.

[20] L. Kaufman and P. J. Rousseeuw, *Finding groups in data: An introduction to cluster analysis*. Wiley, 1990.

[21] J. C. Bezdek, *Pattern recognition with fuzzy objective function algorithms*. Plenum Press, 1981.

[22] Z. Huang, "Extensions to the k-modes algorithm for clustering large data sets with categorical values," *Data mining and knowledge discovery*, vol. 2, no. 3, pp. 283–304, 1997.

[23] Z. Huang, "Clustering large data sets with mixed numeric and categorical values," in *Proceedings of the First Pacific Asia Knowledge Discovery and Data Mining Conference*, pp. 21–34, 1998.

[24] C. Bouveyron and C. Brunet-Saumard, "High-dimensional data clustering," *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 1, no. 1, pp. 31–40, 2007.

[25] F. Nie, E. P. Xing, and S. C. Hoi, "Spherical k-means: A low-dimensional clustering algorithm," in *Proceedings of the 28th international conference on machine learning (ICML-11)*, pp. 681–688, October 2011.

[26] D. Sculley, "Web-scale k-means clustering," in *Proceedings of the 19th international conference on World wide web*, pp. 1177–1178, ACM, April 2010.

[27] "Php manual." `http://php.net/manual/en/`.

[28] "Learn php." `https://www.codecademy.com/learn/learn-php`.

[29] Composer, "Introduction - composer." Accessed 20 Jan. 2023.

[30] P. Gazarov, "What is an api? in english, please.." `https://www.freecodecamp.org/news/what-is-an-api-in-english-please-b880a3214a82/`, 2021. Accessed: January 20, 2023.

[31] "What is a rest api?." `https://www.redhat.com/en/topics/api/what-is-a-rest-api`, 2020.

[32] "Restful api designing guidelines." `https://restfulapi.net/`, 2022. Accessed: January 20, 2023.

[33] "Welcome to python.org." `https://www.python.org/`. Accessed: January 20, 2023.

[34] S. Kumar, "Top advantages of python over other programming languages." `https://www.edoxi.com/studyhub-detail/advantages-of-python-over-other-programming-languages`, 2022. Accessed: January 20, 2023.

[35] "Tensorflow." `https://www.tensorflow.org/`. Accessed: January 20, 2023.

[36] "Scikit-learn: Machine learning in python." `https://scikit-learn.org/stable/`. Accessed: January 20, 2023.

[37] "Keras: The python deep learning api." https://keras.io/. Accessed: January 20, 2023.

[38] "Pytorch." https://www.pytorch.org. Accessed: January 20, 2023.

[39] "Pandas - python data analysis library." https://pandas.pydata.org/. Accessed: January 20, 2023.

[40] "Numpy." https://numpy.org/. Accessed: January 20, 2023.

[41] "Mysql." https://www.mysql.com/. Accessed: January 20, 2023.

[42] Derek, "7 benefits of using mysql in your business." https://bootcamp.uxdesign.cc/7-benefits-of-using-mysql-in-your-business-e587e326144f, 2022. Accessed: January 20, 2023.

[43] Datamation, "8 big advantages of using mysql." https://www.datamation.com/storage/8-major-advantages-of-using-mysql/, 2016. Accessed: January 20, 2023.

[44] "Xampp tutorial - javatpoint." https://www.javatpoint.com/xampp. Accessed: January 20, 2023.

[45] "Meet postman." https://www.postman.com/api-platform/meet-postman. Accessed: January 20, 2023.

[46] M. D. Network, "Html: Hypertext markup language." Accessed 20 Jan. 2023.

[47] M. D. Network, "Css: Cascading style sheets." Accessed 20 Jan. 2023.

[48] WhatIs.Com, "What is a bootstrap and how does it work?." Accessed 20 Jan. 2023.

[49] M. D. Network, "What is javascript? - learn web development." Accessed 20 Jan. 2023.

[50] J. Foundation, "jquery." Accessed 20 Jan. 2023.

[51] M. Martin, "What is a functional requirement in software engineering?." https://www.guru99.com/functional-requirement-specification-example.html, 2023. Accessed: January 25, 2023.

[52] MariaDB, "Mariadb vs. mysql - open source relational databases." Accessed: January 30, 2023.

[53] M. Shacklett and P. Loshin, "What is md5 (md5 message-digest algorithm)," 2021. Accessed: January 30, 2023.

[54] R. S, "Er diagrams in dbms: Entities relationship diagram model," 2022. Accessed: January 30, 2023.

[55] W3Schools, "Sql stored procedures." Accessed: January 31, 2023.

[56] Solver, "Using k-means clustering," 2019. Accessed: January 31, 2023.