



Αναφορά Εργασίας στην Τεχνολογία Λογισμικού

Όνομα: Κωνσταντίνος Καφτεράνης

ΑΜ: inf2021090

Σεπτέμβριος 2024

Περιεχόμενα

1	Εισαγωγή	2
2	Μεθοδολογία Υλοποίησης - ΚλυκλοςΖωής Λογισμικού	3
2.1	Μοντέλο Επαναληπτικής Ανάπτυξης (Iterative Model)	3
3	Δομή Εφαρμογής	3
3.1	App	4
3.1.1	display_data	4
3.1.2	split_data	4
3.1.3	main	4
3.2	ML	4
3.2.1	Clustering	4
3.2.2	Classification	5
3.2.3	Visualization	5
3.3	Utils	5
3.3.1	Data Loading	5
3.3.2	Classifier Comparison	5
3.3.3	Cluster Comparison	6
3.3.4	App Information	6
4	Παρουσίαση Εφαρμογής	6
4.0.1	Navigation	6
4.0.2	Φόρτωση Δεδομένων	7
4.0.3	Οπτικοποίηση	8
4.0.4	Classification	9
4.0.5	Clustering	11
5	Χρήση μέσω Docker	13
6	Ομάδα	13
7	Σύνδεσμοι Κώδικα,Αναφοράς Διαγράμματος	13

1 Εισαγωγή

Για την εργασία του μαθήματος Τεχνολογία Λογισμικού, δημιουργήθηκε μια διαδικτυακή εφαρμογή μηχανικής μάθησης για την ανάλυση δεδομένων. Η υλοποίηση πραγματοποιήθηκε με τη γλώσσα προγραμματισμού Python χρησιμοποιώντας τη βιβλιοθήκη Streamlit καθώς και άλλες βιβλιοθήκες μηχανικής μάθησης όπως Scikit-learn, Pandas και Matplotlib. Η εφαρμογή ενσωματώνει διάφορους αλγόριθμους μηχανικής μάθησης για κατηγοριοποίηση και ομαδοποίηση, ενώ προσφέρει δυνατότητες οπτικοποίησης των δεδομένων μέσω γραφημάτων και διαγραμμάτων. Η εφαρμογή παρέχει μια φιλική προς τον χρήστη διεπαφή για τη φόρτωση δεδομένων, την επεξεργασία τους και την εξαγωγή πολύτιμων πληροφοριών μέσω στατιστικής ανάλυσης και οπτικοποίησης.

2 Μεθοδολογία Υλοποίησης - ΚλυκλοςΖωής Λογισμικού

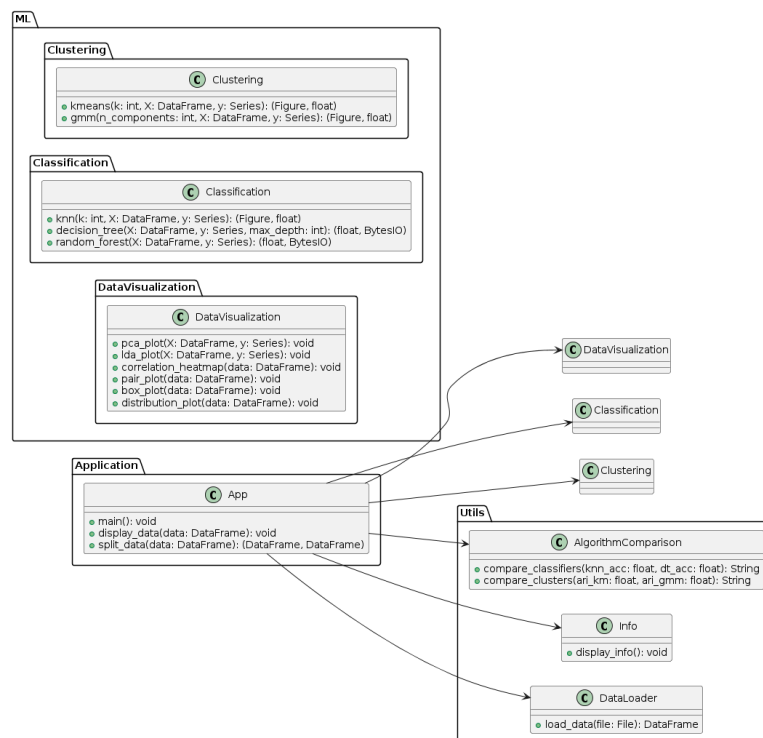
2.1 Μοντέλο Επαναληπτικής Ανάπτυξης (Iterative Model)

Το Μοντέλο Επαναληπτικής Ανάπτυξης είναι μια ευέλικτη προσέγγιση ανάπτυξης λογισμικού που βασίζεται στη συνεχή βελτίωση του λογισμικού μέσα από επαναλαμβανόμενες φάσεις ανάπτυξης.

- Ξεκινάει με τη συλλογή των απαιτήσεων και την ανάλυση των αναγκών του έργου.
- Ο κώδικας γράφεται σε μικρές, διαχειρίσιμες μονάδες, με κάθε μονάδα να δοκιμάζεται ξεχωριστά, δίνοντας έμφαση στη διόρθωση λαθών πριν προχωρήσουμε στην επόμενη ενότητα.
- Η ανάπτυξη γίνεται τμηματικά, με την ολοκλήρωση κάθε μονάδας να συμβάλλει στη συνολική ανάπτυξη του συστήματος.
- Οι τελικές διορθώσεις και βελτιώσεις γίνονται μόνο όταν το κύριο σύστημα είναι σταθερό και λειτουργικό.

Αυτή η προσέγγιση επιτρέπει την γρήγορη διόρθωση λαθών και την καλύτερη διαχείριση των απαιτήσεων που μπορεί να αλλάζουν κατά την ανάπτυξη της εφαρμογής.

3 Δομή Εφαρμογής



Σχήμα 1: Το UML διάγραμμα.

3.1 App

3.1.1 display_data

- Η συνάρτηση `display_data` εμφανίζει τον πίνακα δεδομένων στην εφαρμογή.
- Συγκεκριμένα, χρησιμοποιεί την `st.write` για να προβάλλει τα δεδομένα στην πλευρική γραμμή της εφαρμογής.

3.1.2 split_data

- Η συνάρτηση `split_data` διαχωρίζει τα δεδομένα σε χαρακτηριστικά (X) και στόχο (y).
- Τα χαρακτηριστικά περιλαμβάνουν όλες τις στήλες εκτός από την τελευταία, ενώ η τελευταία στήλη θεωρείται η μεταβλητή στόχος.
- Εμφανίζει τα χαρακτηριστικά και τη μεταβλητή στόχο στην εφαρμογή χρησιμοποιώντας την `st.write` και επιστρέφει τα X και y .

3.1.3 main

- Η συνάρτηση `main` είναι η κύρια συνάρτηση της εφαρμογής και διαχειρίζεται την κύρια ροή της εφαρμογής.
- Διαχειρίζεται την πλοήγηση μέσω της sidebar μή ί ά ή ά μ μέ ό.
- Η ενότητα Upload Data φορτώνει και εμφανίζει δεδομένα από αρχεία CSV ή Excel.
- Η ενότητα Data Visualization επιτρέπει την απεικόνιση των δεδομένων με διάφορους τύπους γραφημάτων, όπως PCA και LDA.
- Η ενότητα Machine Learning εκτελεί αλγορίθμους μηχανικής μάθησης, όπως KNN και Decision Trees για ταξινόμηση, και K-Means και GMM για ομαδοποίηση.
- Η ενότητα Information καλεί τη συνάρτηση `display_info` για να εμφανίσει πληροφορίες σχετικά με την εφαρμογή.

3.2 ML

3.2.1 Clustering

Η ομαδοποίηση είναι μια μη επιβλεπόμενη τεχνική μηχανικής μάθησης που έχει ως στόχο να βρει ομάδες ή συσσωματώματα παρόμοιων δεδομένων. Εδώ είναι μερικές από τις χρησιμοποιούμενες μεθόδους:

- **K-Means Clustering:** Η συνάρτηση `'kmeans'` χρησιμοποιεί τον αλγόριθμο K-Means για να ομαδοποιήσει τα δεδομένα σε k ομάδες. Τα δεδομένα κανονικοποιούνται και στη συνέχεια γίνεται πρόβλεψη των συστάδων για το δοκιμαστικό σύνολο δεδομένων. Ο υπολογισμός του προσαρμοσμένου δείκτη Ραντ (ARI) χρησιμοποιείται για την αξιολόγηση της ακρίβειας.
- **Gaussian Mixture Model (GMM):** Η συνάρτηση `'gmm'` εφαρμόζει το μοντέλο Gaussian Mixture για να εντοπίσει συστάδες με βάση την κανονική κατανομή των δεδομένων. Το αποτέλεσμα αξιολογείται επίσης μέσω του προσαρμοσμένου δείκτη Rand (ARI).

3.2.2 Classification

Η κατηγοριοποίηση είναι η διαδικασία της ανάθεσης μιας ετικέτας σε κάθε δείγμα βάσει των χαρακτηριστικών του. Μερικές από τις χρησιμοποιούμενες μεθόδους περιλαμβάνουν:

- K-Nearest Neighbors (KNN): Η συνάρτηση `knn` υλοποιεί τον αλγόριθμο K-Nearest Neighbors (KNN) για την κατηγοριοποίηση των δεδομένων. Ο αλγόριθμος βασίζεται στους πιο κοντινούς γείτονες ενός δείγματος για να προσδιορίσει την κλάση του.
- Decision Tree: Η συνάρτηση `decision_tree` υλοποιεί τον αλγόριθμο Decision Tree για την κατηγοριοποίηση των δεδομένων.
- Random Forest: Η συνάρτηση `random_forest` υλοποιεί τον αλγόριθμο Random Forest, ο οποίος συνδυάζει πολλούς Decision Trees για να βελτιώσει την ακρίβεια της πρόβλεψης.

3.2.3 Visualization

Η οπτικοποίηση των δεδομένων είναι σημαντική για την κατανόηση της δομής και των σχέσεων που υπάρχουν στα δεδομένα. Μερικές από τις διαθέσιμες μεθόδους περιλαμβάνουν:

- PCA (Principal Component Analysis): Η συνάρτηση `pca_plot` χρησιμοποιεί την PCA για τη μείωση των διαστάσεων των δεδομένων και την απεικόνισή τους σε έναν διδιάστατο χώρο, διατηρώντας όσο το δυνατόν περισσότερη πληροφορία.
- LDA (Linear Discriminant Analysis): Η συνάρτηση `lda_plot` εφαρμόζει την LDA για την απεικόνιση των δεδομένων σε δύο διαστάσεις, διαχωρίζοντας τις κλάσεις με βάση τα γραμμικά διακριτικά.
- Correlation Heatmap: Η συνάρτηση `correlation_heatmap` δημιουργεί έναν θερμικό χάρτη που εμφανίζει τις συσχετίσεις μεταξύ των αριθμητικών χαρακτηριστικών ενός συνόλου δεδομένων.
- Box Plot: Η συνάρτηση `box_plot` δημιουργεί ένα box plot για να εμφανίσει την κατανομή των αριθμητικών δεδομένων και να εντοπίσει πιθανές ακραίες τιμές.
- Pair Plot: Η συνάρτηση `pair_plot` απεικονίζει ζεύγη μεταβλητών για να εξετάσει τις σχέσεις τους μεταξύ τους.
- Distribution Plot: Η συνάρτηση `distribution_plot` εμφανίζει την κατανομή των χαρακτηριστικών σε ένα σύνολο δεδομένων, συνδυάζοντας ιστογράμματα και καμπύλες πυκνότητας.

3.3 Utils

3.3.1 Data Loading

Η συνάρτηση `load_data` φορτώνει δεδομένα από διάφορους τύπους αρχείων και ελέγχει αν είναι CSV ή Excel. Επιστρέφει τα δεδομένα ή ένα μήνυμα σφάλματος.

3.3.2 Classifier Comparison

Η συνάρτηση `compare_classifiers` συγκρίνει τις επιδόσεις των αλγορίθμων K-Nearest Neighbors (KNN) και Decision Tree (DT) και εμφανίζει προειδοποιήσεις για ενδεχόμενη υπερεκπαίδευση.

3.3.3 Cluster Comparison

Η συνάρτηση `compare_clusters` sugkrneiticepidseictwnalgorjmw nK – MeanskaiGaussianMixtureMode

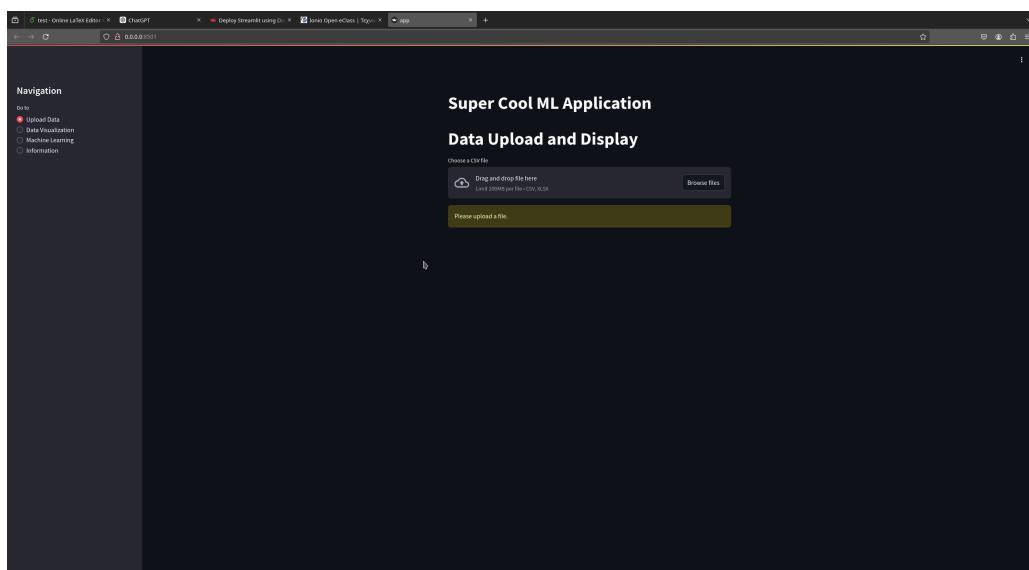
3.3.4 App Information

Πληροφορίες Εφαρμογής

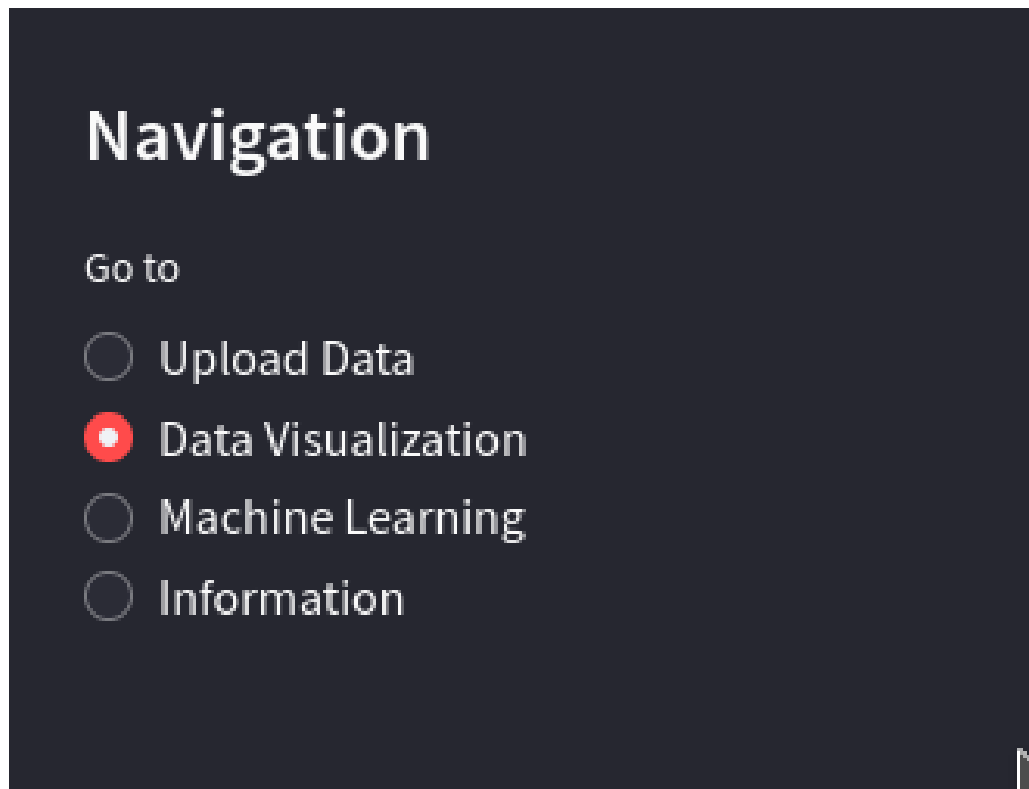
4 Παρουσίαση Εφαρμογής

4.0.1 Navigation

Παρακάτω εξηγείται πώς χρησιμοποιούμε την εφαρμογή. Κατά την είσοδο στην εφαρμογή, ο χρήστης έρχεται σε επαφή με το ΥΙ, το οποίο περιλαμβάνει ένα ωιδγετ για ανέβασμα αρχείων, καθώς και ένα σιδεβαρ μενυ που χρησιμοποιεί για να περιγηθεί στην εφαρμογή.



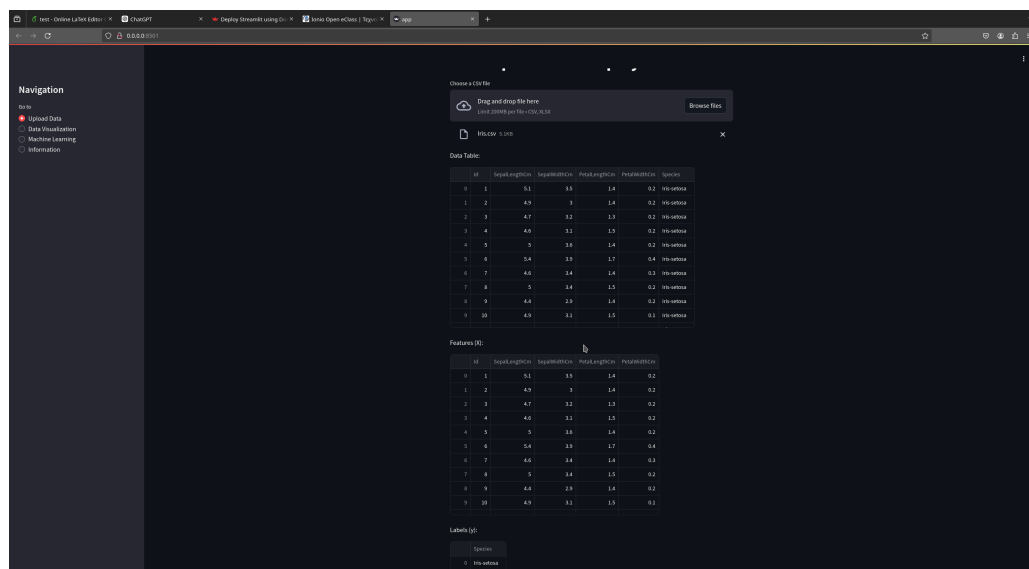
Σχήμα 2: Οθόνη Έναρξης



Σχήμα 3: Μενού

4.0.2 Φόρτωση Δεδομένων

Τα αρχεία δεδομένων πρέπει να είναι της μορφής Excel ή CSV, αλλιώς η εφαρμογή δεν θα επιτρέψει το ανέβασμά τους και θα ζητήσει από τον χρήστη να τα ανεβάσει ξανά.



Σχήμα 4: Φόρτωση Δεδομένων

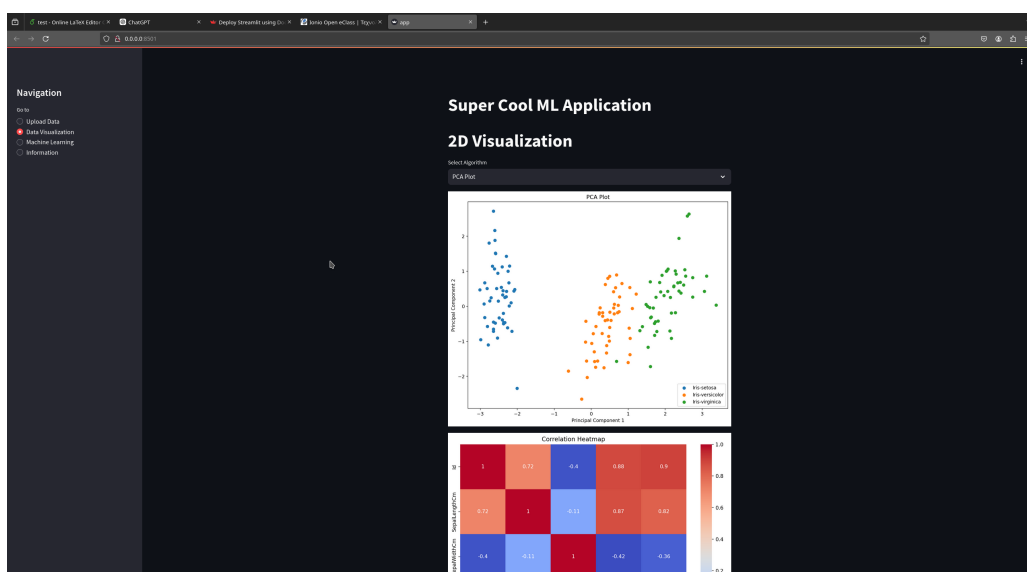
Αφού το αρχείο ανέβει, εμφανίζονται τα δεδομένα χωρισμένα σε χαρακτηριστικά (features) και ετικέτες.

Στο μενού, ο χρήστης μπορεί να επιλέξει την επεξεργασία που θα εφαρμόσει στα δεδομένα: οπτικοποίηση, εφαρμογή μηχανικής μάθησης ή τη σελίδα με τις οδηγίες.

Αν επιστρέψει στο ανέβασμα αρχείων, τότε τα δεδομένα χάνονται και ζητείται να ανέβει ένα καινούργιο αρχείο.

4.0.3 Οπτικοποίηση

Επιλέγοντας οπτικοποίηση, ο χρήστης έχει δύο δυνατότητες. Από ένα dropdown menu επιλέγει είτε τον αλγόριθμο PCA ή LDA και να δημιουργήσει ένα διάγραμμα, καθώς επίσης και μερικά διαγράμματα επεξηγηματικών δεδομένων, τα οποία θα εμφανιστούν κάθετα κάτω από το διάγραμμα του αλγορίθμου οπτικοποίησης. Τα διαγράμματα αυτά είναι:

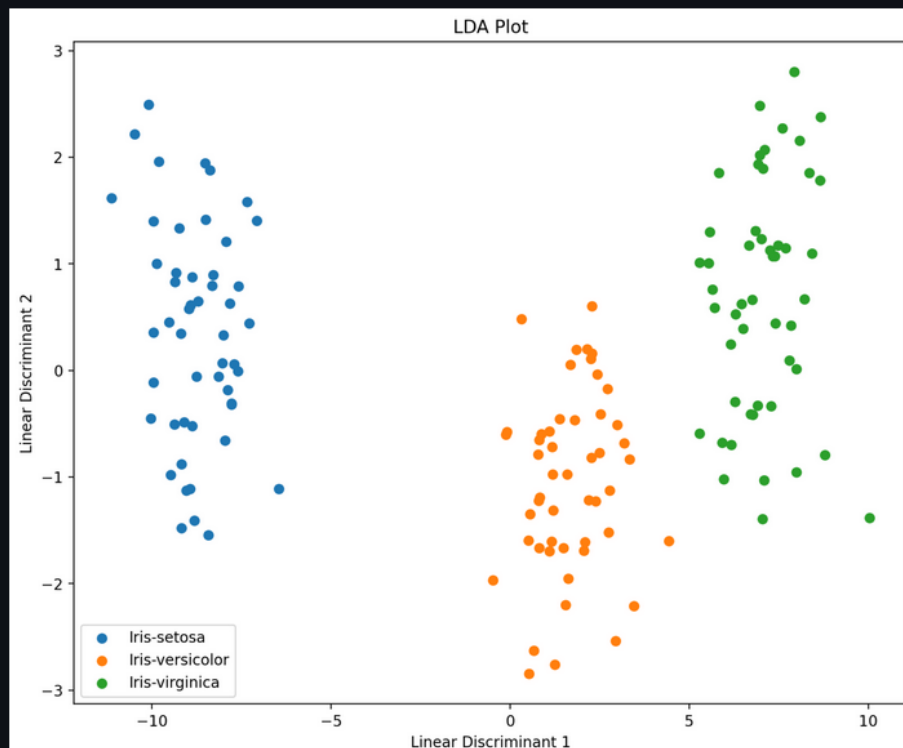


Σχήμα 5: Οπτικοποίηση Δεδομένων

2D Visualization

Select Algorithm

LDA Plot



Σχήμα 6: Ο αλγόριθμος LDA

4.0.4 Classification

Όσον αφορά τη μηχανική μάθηση, ο χρήστης μπορεί να επιλέξει ανάμεσα σε δύο προβλήματα, classification και clustering, τσεκάροντας το αντίστοιχο κυκλάκι. Στην περίπτωση της κατηγοριοποίησης, ο χρήστης επιλέγει μεταξύ των αλγορίθμων k-means και decision tree. Σε κάθε περίπτωση, χρησιμοποιεί μια μπάρα για να επιλέξει το k, και εφαρμόζονται και οι δύο αλγόριθμοι. Τα αποτελέσματα συγκρίνονται με βάση τον δείκτη accuracy για να αξιολογηθεί η επίδοσή τους. Τις περισσότερες φορές, το decision tree υπερέχει του k-nearest neighbors (KNN), καθώς για μερικές τιμές του k το KNN υπερπροσαρμόζεται (overfits). Εμφανίζεται μήνυμα που προειδοποιεί τον χρήστη να επιλέξει τον κατάλληλο αλγόριθμο κάθε φορά.

Machine Learning

Select problem(s):

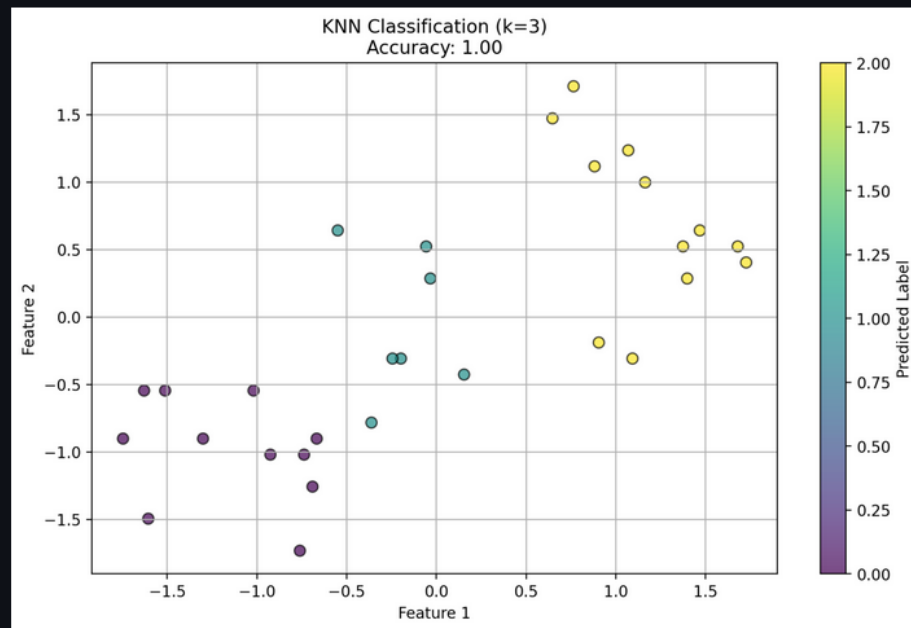
Select Problem Type

☒ Classification

☐ Clustering

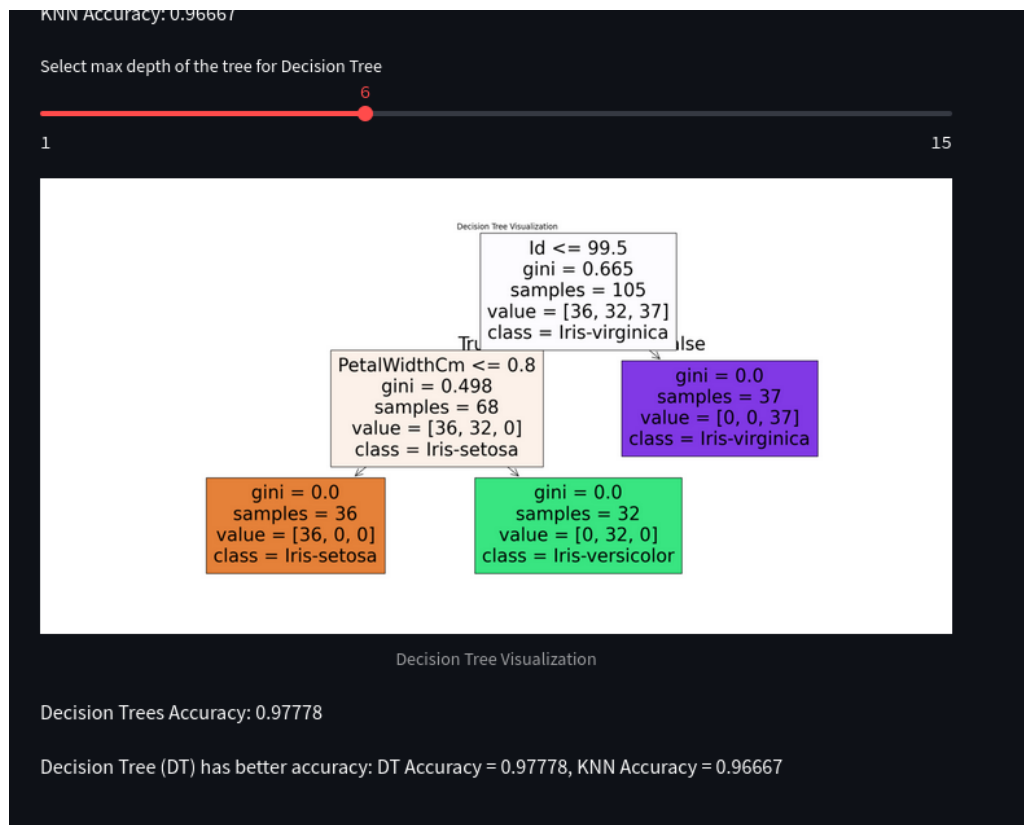
Executing both Random Forests and Decision Trees for Classification...

Select number of neighbors (k) for K-Nearest Neighbors

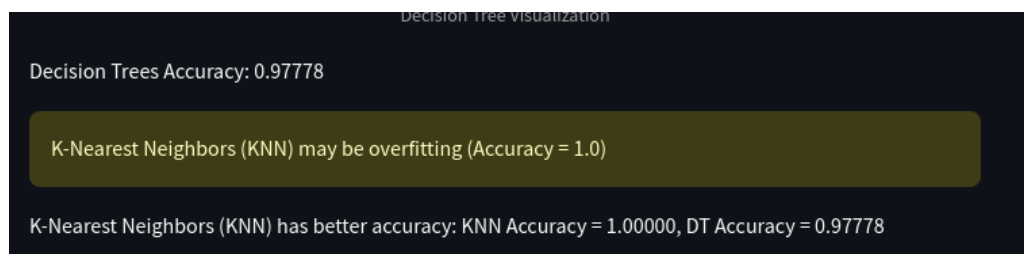


KNN Accuracy: 1.00000

Σχήμα 7: KNN



Σχήμα 8: Δέντρο Επιλογής



Σχήμα 9: Υπερπροσαρμογή

4.0.5 Clustering

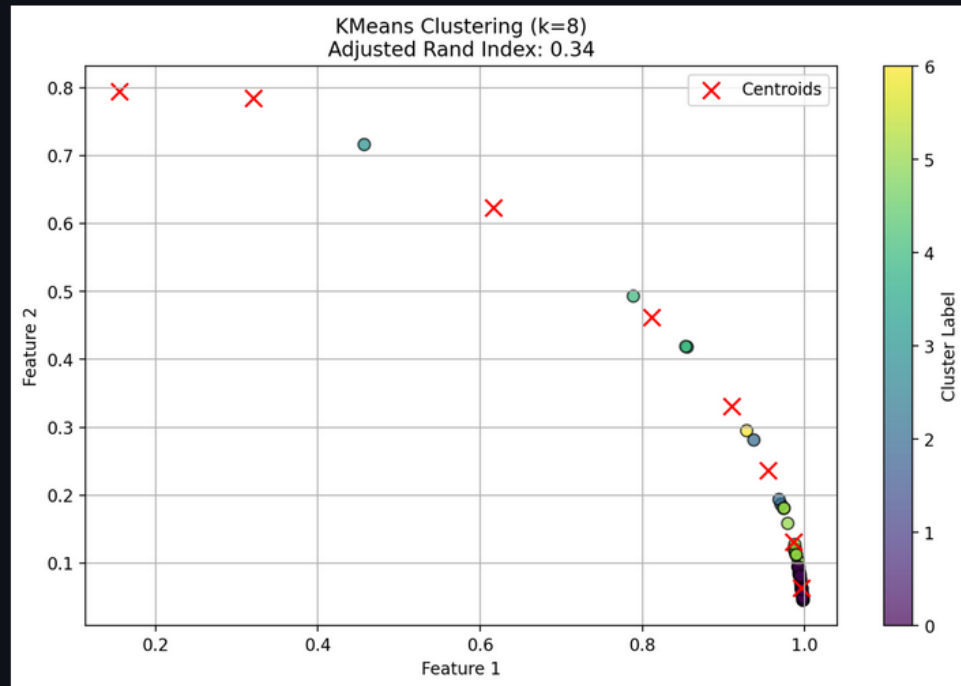
Όσον αφορά το clustering, ο χρήστης μπορεί να επιλέξει μεταξύ των αλγορίθμων Gaussian Mixture Model (GMM) και k-means. Κανένας από τους δύο αλγορίθμους δεν υπερπροσαρμόζεται (overfits), αλλά η απόδοσή τους μπορεί να διαφέρει ανάλογα με τα δεδομένα. Σε ορισμένες περιπτώσεις, το GMM αποδίδει καλύτερα, ενώ σε άλλες το k-means είναι πιο αποδοτικό. Για τη σύγκριση των αποτελεσμάτων χρησιμοποιείται ο δείκτης Adjusted Rand Index (ARI), ο οποίος επιτρέπει την αξιολόγηση της ποιότητας των ομάδων που δημιουργούνται από κάθε αλγόριθμο.

Executing both K-Means and Gaussian Mixture Model (GMM) for Clustering...

Select number of clusters (k) for K-Means



K is: 8

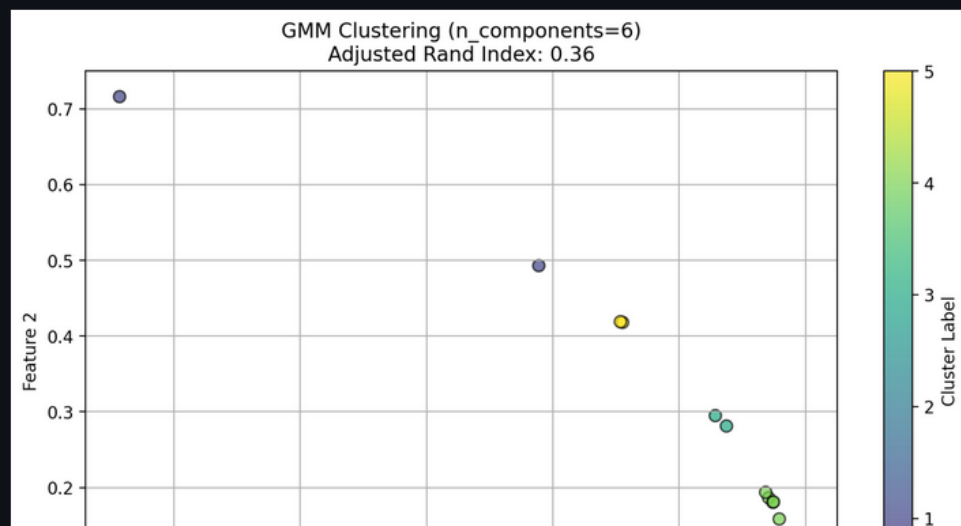


K-Means Adjusted Rand Index: 0.33655

Select number of components for GMM



Number of components: 6



Σχήμα 10: Clustering

5 Χρήση μέσω Docker

- Η εικόνα Python 3.9 Slim χρησιμοποιείται ως βάση.
- Working directory to /app.
- Αντιγράφει το αρχείο requirements.txt στο κοντέινερ.
- Κατεβάζει τα πακέτα από requirements.txt ζασηινγ.
- Αντιγράφει τον φάκελο src/ στο κοντέινερ.
- Exposes port 8501 for the app.
- Η εντολή CMD τρέχει την εφαρμογή Streamlit στην πόρτα 8501, ώστε να είναι προσβάσιμη.

Για την ανάπτυξη και διανομή της εφαρμογής:

- Εκτελέστε την εντολή `docker build -t my-streamlit-app .` για να δημιουργήσετε την εικόνα.
- Εκτελέστε την εντολή `docker run -p 8501:8501 my-streamlit-app` για να τρέξετε την εφαρμογή στη διεύθυνση `http://localhost:8501`.

6 Ομάδα

Το εξάμηνο του μαθήματος έλλειπα ερασμους οπότε δεν μπορούσα να βρώ ομάδα μέσα στο καλοκαίρι και έτσι ολοκλήρωσα την εργασία μόνος μου.

7 Σύνδεσμοι Κώδικα, Αναφοράς Διαγράμματος

- [Project GitHub Repository](#)
- [GitHub Profile](#)
- [Source Code](#)
- [Report Raw LaTeX File](#)
- [Report PDF](#)
- [UML Diagram](#)
- [UML Diagram Code](#)