

# Αναφορά Εργασίας

Ιόνιο Πανεπιστήμιο



Μάθημα:  
Τεχνολογία Λογισμικού

Ομάδα:

- Νικόλαος Τρυπάκης inf2021229
- Στέφανος Σφηναρολάκης inf2021218
- Ορέστης Ραφαήλ Μακρής inf2021129

## Περιεχόμενα

1	Εισαγωγή	3
2	Προγραμματιστικές Επιλογές	4
3	UML Διάγραμμα	5
3.1	Κλάσεις Φόρτωσης Δεδομένων . . . . .	5
3.2	Οπτικοποίηση 2D . . . . .	5
3.3	Μηχανική Μάθηση . . . . .	6
3.4	Tab Πληροφοριών . . . . .	6
4	Εκτέλεσης της Εφαρμογής με Docker	7
5	Guide της Εφαρμογής	8
5.1	Είσοδος στην εφαρμογή . . . . .	8
5.2	Data Loader Tab . . . . .	9
5.3	2D Visualization . . . . .	11
5.3.1	PCA Plot . . . . .	12
5.3.2	t-SNE Plot . . . . .	13
5.3.3	EDA Διαγράμματα . . . . .	14
5.4	Μηχανική Μάθηση . . . . .	16
6	Κύκλος Ζωής Έκδοσης Λογισμικού	19
7	Περιγραφή της συνεισφοράς κάθε μέλους της ομάδας	20
8	Github Repositories	21

## 1 Εισαγωγή

Μας ζητήθηκε η ανάπτυξη μιας web-based εφαρμογής, η οποία παρέχει τις λειτουργίες της φόρτωσης, της ανάλυσης και της οπτικοποίησης δεδομένων μέσω ευέλικτων λειτουργιών. Η εφαρμογή δημιουργήθηκε χρησιμοποιώντας την βιβλιοθήκη Streamlit, ενώ υποστηρίζει διάφορες αναλυτικές και οπτικοποιητικές λειτουργίες. Στο πλαίσιο αυτό, η εφαρμογή διαθέτει:

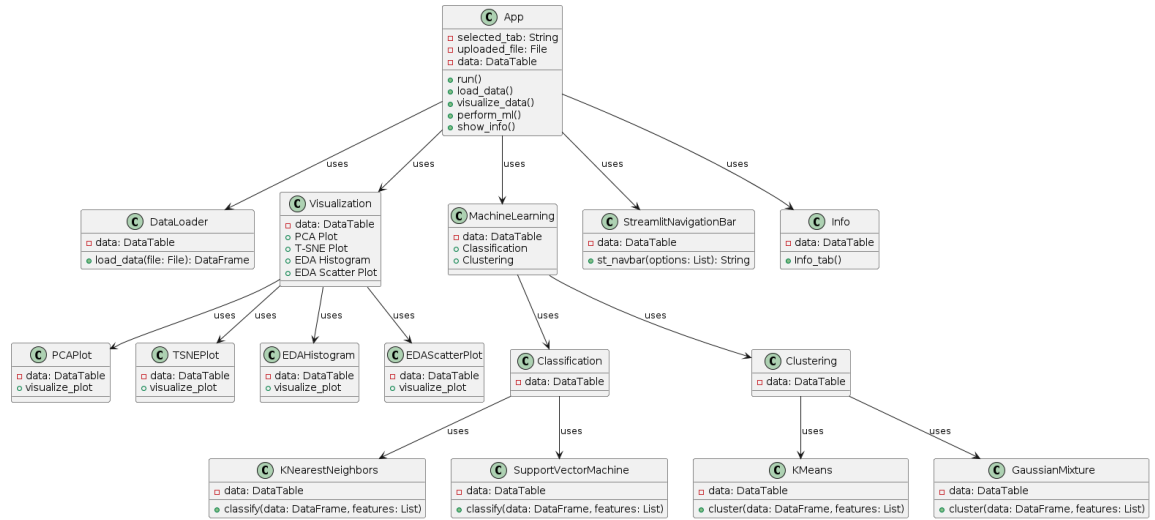
- Φόρτωση Δεδομένων: Η δυνατότητα φόρτωσης δεδομένων σε μορφή πίνακα από αρχεία δεδομένων.
- Οπτικοποίηση Δεδομένων: Οπτικοποιήσεις που επιτρέπουν την εξαγωγή σημαντικών πληροφοριών από τα δεδομένα.
- Σύγκριση Αλγορίθμων: Δυνατότητα σύγκρισης της απόδοσης διάφορων αλγορίθμων μηχανικής μάθησης για τα δεδομένα που αναλύονται.
- Παρουσίαση: Παρουσίαση των αποτελεσμάτων ανάλυσης.

Στόχος είναι η εφαρμογή να αποτελέσει ένα πολύτιμο εργαλείο για την ανάλυση και διευκολία στην ανακάλυψη νέων πληροφοριών από τα δεδομένα των χρηστών.

## 2 Προγραμματιστικές Επιλογές

- **Θέμα Φιλικό προς τον Χρήστη:** Το αρχείο `app.py` είναι εφαρμοσμένο για τον χρήστη χρησιμοποιώντας τις δυνατότητες του Streamlit καθώς και της χρήσης του Streamlit Navigation Bar. Αυτές οι επιλογές βελτιώνουν την εμπειρία του χρήστη παρέχοντας ένα οπτικά κατανοητό περιβάλλον ως προς την διάταξη της σελίδας.
- **Δομή Κώδικα σε Ανεξάρτητα Μέρη:** Ο κώδικας οργανώνεται σε ξεχωριστά modules (`data_loader.py`, `visualization.py`, `machinelearning.py`, `info.py`). Αυτό επιτρέπει ευκολότερη πλοήγηση, εντοπισμό σφαλμάτων και μελλοντικές ενημερώσεις, καθώς κάθε module επικεντρώνεται σε ένα συγκεκριμένο κομμάτι της λειτουργικότητας της εφαρμογής.
- **Λειτουργεία Ανεβάσματος Αρχείων:** Η λειτουργία `st.file_uploader` από το Streamlit χρησιμοποιείται για να επιτρέψει στους χρήστες να ανεβάζουν τα αρχεία δεδομένων τους (CSV, Excel). Αυτή η επιλογή παρέχει ευελιξία και άνεση στους χρήστες, επιτρέποντάς τους να αναλύουν τα δικά τους datasets εντός της εφαρμογής.
- **Οπτικοποίηση:** Διαδραστικά γραφήματα δημιουργούνται χρησιμοποιώντας το `plotly.express` στο module `visualization.py`. Το Plotly Express προσφέρει μια υψηλού επιπέδου διεπαφή για τη δημιουργία εκφραστικών και διαδραστικών οπτικοποιήσεων με ελάχιστο κώδικα. Αυτή η επιλογή βελτιώνει την ανάλυση των δεδομένων επιτρέποντας στους χρήστες να δουν εύκολα τις σχέσεις στα δεδομένα τους.
- **Αλγόριθμοι Μηχανικής Μάθησης:** Αλγόριθμοι ταξινόμησης και ομαδοποίησης όπως οι K-Nearest Neighbors, Support Vector Machine, K-Means και Gaussian Mixture Models χρησιμοποιήθηκαν μέσω της βιβλιοθήκης `scikit-learn`.

### 3 UML Διάγραμμα



#### 3.1 Κλάσεις Φόρτωσης Δεδομένων

**DataLoader:** Αυτή η κλάση είναι υπεύθυνη για τη φόρτωση δεδομένων από αρχεία CSV και Excel. Τα δεδομένα φορτώνονται στο DataTable.

**DataTable:** Η κλάση DataTable διαχειρίζεται τα δεδομένα που φορτώθηκαν.

#### 3.2 Οπτικοποίηση 2D

Αυτή η κλάση δημιουργεί τα ακόλουθα γραφήματα οπτικοποίησης:

- **EDAHistogram:** Προβάλλει ιστόγραμμα.
- **EDAScatterPlot:** Απεικονίζει την κατανομή των δεδομένων χρησιμοποιώντας scatter plot.
- **PCAPlot:** Παρουσιάζει τα αποτελέσματα της PCA.
- **TSNEPlot:** Εμφανίζει τα αποτελέσματα του t-SNE.

### 3.3 Μηχανική Μάθηση

Αυτή η κλάση περιλαμβάνει λειτουργίες μηχανικής μάθησης. Περιέχει δύο υποσυστήματα:

- **Classification:** Χρησιμοποιείται για την εκτέλεση αλγορίθμων ταξινόμησης.
  - **KNearestNeighbors**
  - **Support Vector Machine**
- **Clustering:** Χρησιμοποιείται για την εκτέλεση αλγορίθμων ομαδοποίησης.
  - **KMeans**
  - **GaussianMixture**

### 3.4 Tab Πληροφοριών

Η καρτέλα InfoTab παρέχει πληροφορίες σχετικά με την εφαρμογή και τα μέλη της ομάδας που δούλεψαν στην ανάπτυξη της.

## 4 Εκτέλεσης της Εφαρμογής με Docker

Αρχικά, χρειάζεται να κατέβει το Docker Desktop στο σύστημα. Μετά χρειάζεται η δημιουργία του dockerfile:

```
FROM python:3.11.9  
WORKDIR /app  
COPY requirements.txt requirements.txt  
RUN pip install -r requirements.txt  
COPY . .  
CMD ["streamlit", "run", "app/app.py"]
```

Μέσω αυτών των εντολών όπου αποτελούν το περιεχόμενο του αρχείου "Dockerfile" ορίζεται τον ευρετήριο της εφαρμογής, κατεβαίνουν οι απαραίτητες βιβλιοθήκες και μέσω του Streamlit τρέχει το βασικό αρχείο της εφαρμογής. Έπειτα την δημιουργία του αρχείου θα χρειαστεί να εκτελεστούν οι ακόλουθες εντολές:

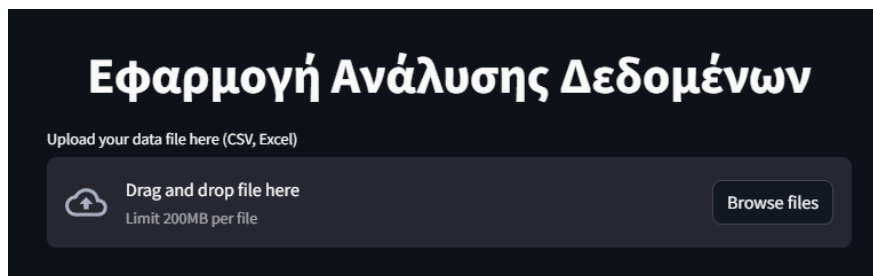
- **docker build -t "your\_app\_name" .**
- **docker run -p 8501:8501 "your\_app\_name"**

## 5 Guide της Εφαρμογής

Κατά την χρήση της εφαρμογής θα πρέπει να χρησιμοποιηθεί ένα αρχείο με data από τον χρήστη. Για το παράδειγμα του guide θα χρησιμοποιηθεί ένα αρχείο με data από δείγματα λουλουδιών ίρις.

### 5.1 Είσοδος στην εφαρμογή

Κατά την είσοδο στην ιστοσελίδα ο χρήστης θα βρεθεί με την επιλογή να εισάγει τα δεδομένα του μέσω της επιλογής σχετικού αρχείου.



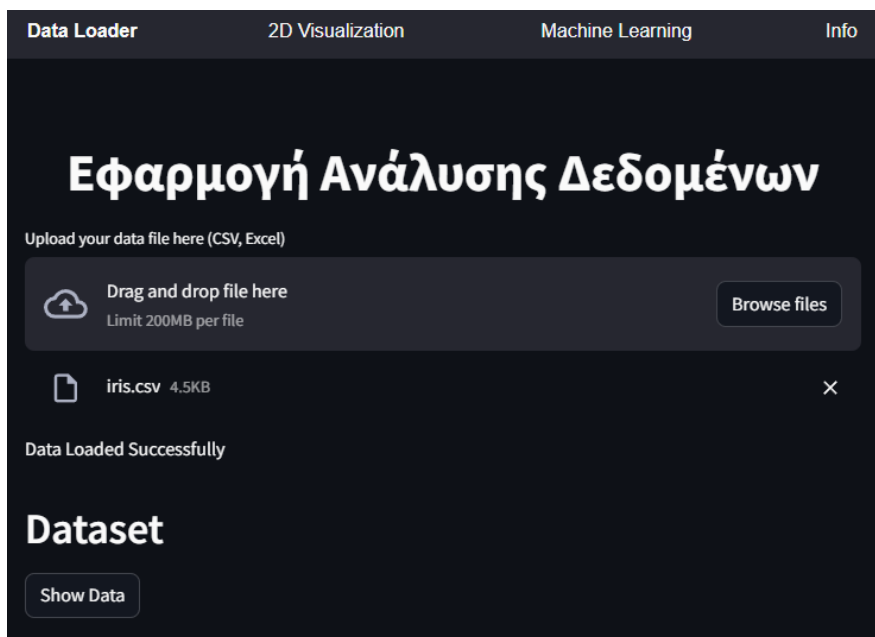
Σχήμα 1: Data Uploader Button



## 5.2 Data Loader Tab

Μετά την εισαγωγή του αρχείου δεδομένων, είναι αρκετά εμφανής η ύπαρξη διαφορετικών σελιδών όπου ο χρήστης έχει διάφορες δυνατότητες σε κάθε μία ξεχωριστά.

Η πρώτη σελίδα της εφαρμογής είναι αυτή του Data Loader όπου ο χρήστης μπορεί να δει τα δεδομένα του καθώς και τις ετικέτες τους πατώντας το κουμπί **Show Data**.




Σχήμα 2: Data Loader


Παράδειγμα φορτωμένων δεδομένων:

Data Loader2D VisualizationMachine LearningInfo

Upload your data file here (CSV, Excel)

 Drag and drop file here  
Limit 200MB per file

Browse files

 iris.csv 4.5KB

×

Data Loaded Successfully

Dataset

Show Data

Data Table

Labels

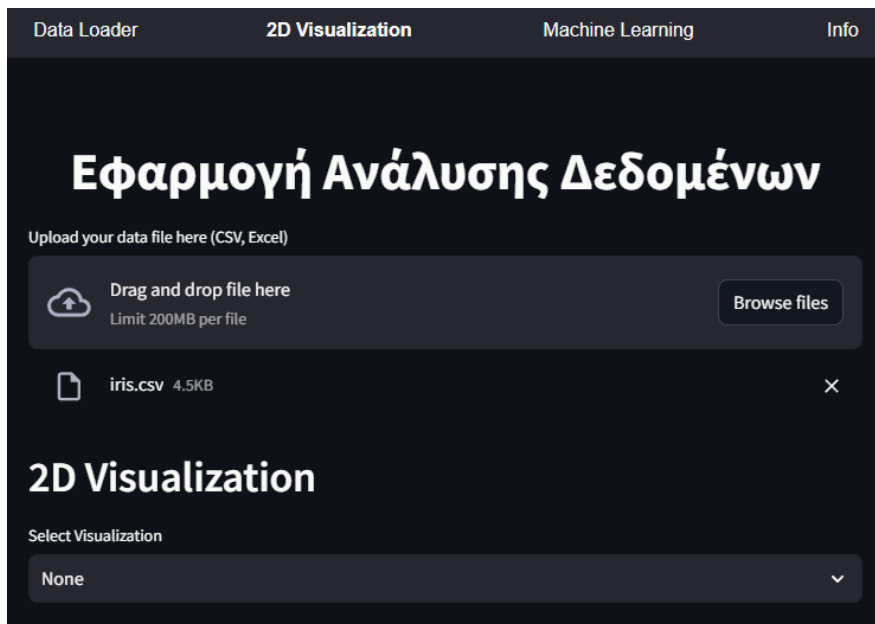
	SepalLength	SepalWidth	PetalLength	PetalWidth
0	5.1	3.5	1.4	
1	4.9	3	1.4	
2	4.7	3.2	1.3	
3	4.6	3.1	1.5	
4	5	3.6	1.4	
5	5.4	3.9	1.7	
6	4.6	3.4	1.4	
7	5	3.4	1.5	
8	4.4	2.9	1.4	
9	4.9	3.1	1.5	

	Species
0	Iris-setosa
1	Iris-setosa
2	Iris-setosa
3	Iris-setosa
4	Iris-setosa
5	Iris-setosa
6	Iris-setosa
7	Iris-setosa
8	Iris-setosa
9	Iris-setosa

Σχήμα 3: Φορτωμένα Δεδομένα

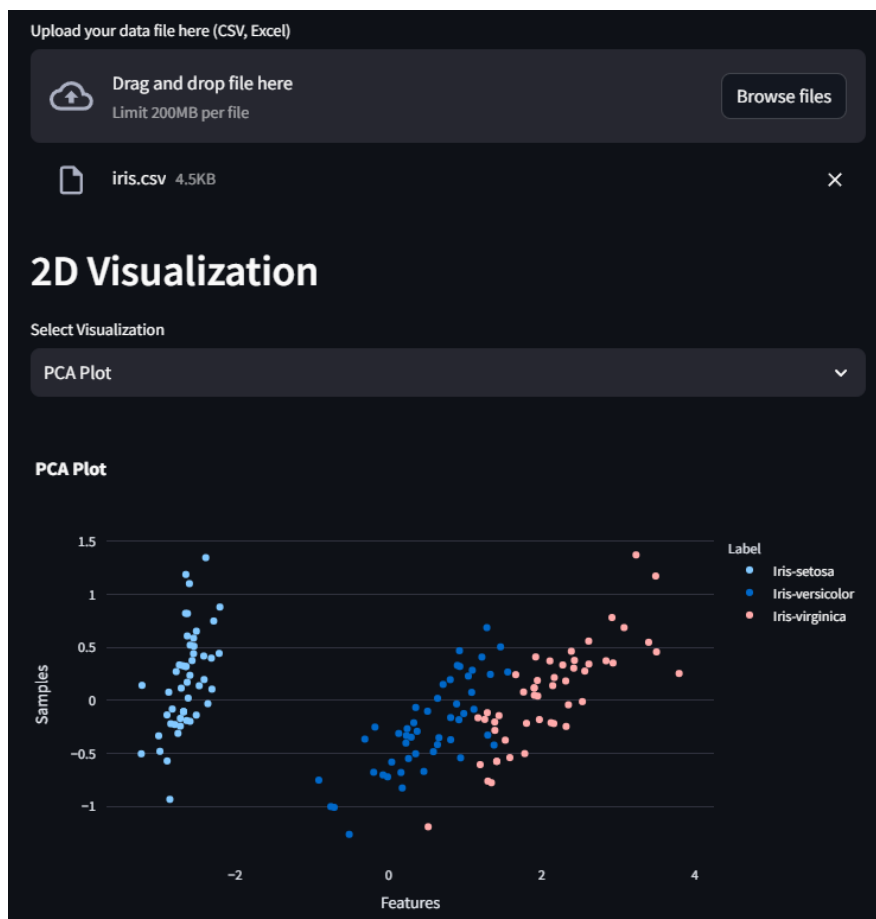
### 5.3 2D Visualization

Στην δεύτερη σελίδα "2D Visualization" ο χρήστης έχει την δυνατότητα να εκτελέσει οπτικοποιήσεις έτοιμων αλγορίθμων (PCA, t-SNE) με τα δεδομένα του και να τα αναλύσει με κατάλληλα σχήματα (EDA Charts) μέσω επίλογης από το διαθέσιμο μενού.



Σχήμα 4: 2D Visualization

### 5.3.1 PCA Plot



Σχήμα 5: PCA Plot

Το PCA μετασχηματίζει τα δεδομένα από τον αρχικό χώρο των 4 διαστάσεων (μήκος και πλάτος πετάλων και σέpalων) σε ένα 2D διάγραμμα που διατηρεί τη μέγιστη δυνατή διαχύμανση. Οι τρεις κατηγορίες (Iris-setosa, Iris-versicolor, Iris-virginica) απεικονίζονται σε διαφορετικά χρώματα, επιτρέποντας τον οπτικό διαχωρισμό τους.

### 5.3.2 t-SNE Plot

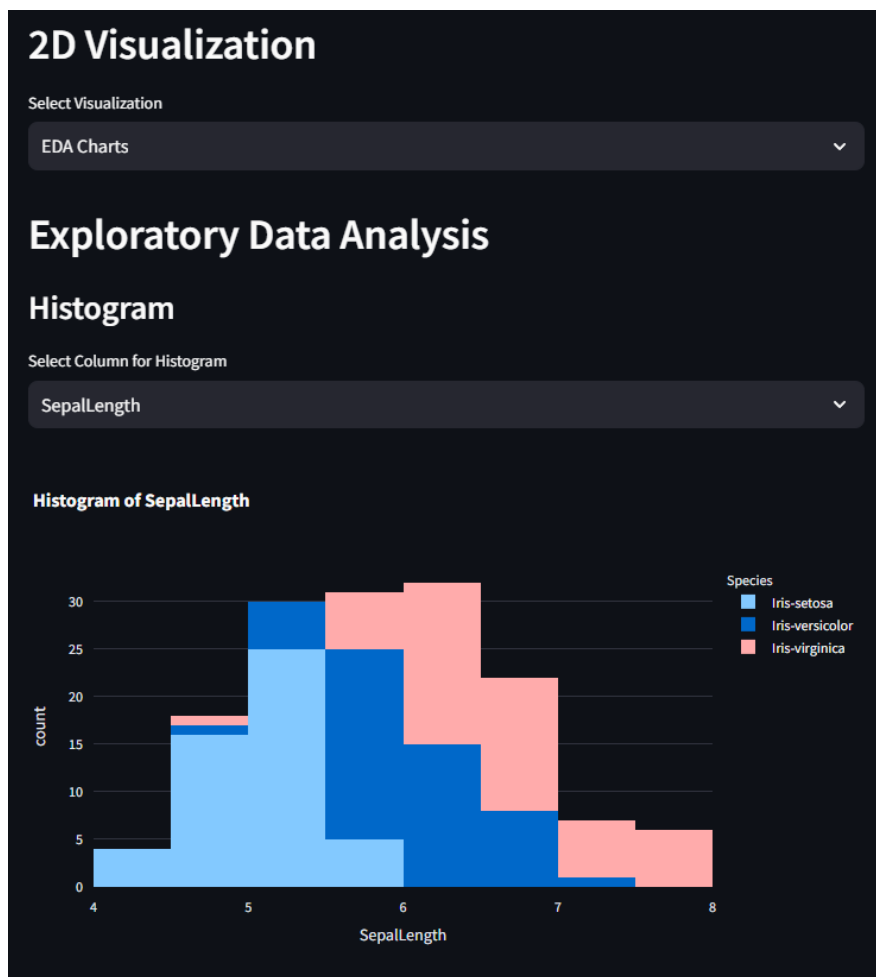


Σχήμα 6: t-SNE Plot

Το t-SNE χαρτογραφεί τα δεδομένα σε έναν 2D χώρο, διατηρώντας τις τοπικές σχέσεις μεταξύ των σημείων. Αυτό το καθιστά εξαιρετικά χρήσιμο για την οπτικοποίηση συστάδων στα δεδομένα Iris, δείχνοντας πώς τα σημεία δεδομένων συγκεντρώνονται και διαχωρίζονται μεταξύ των τριών κατηγοριών. Όπως και πριν, οι τρεις κατηγορίες απεικονίζονται σε διαφορετικά χρώματα.

### 5.3.3 EDA Διαγράμματα

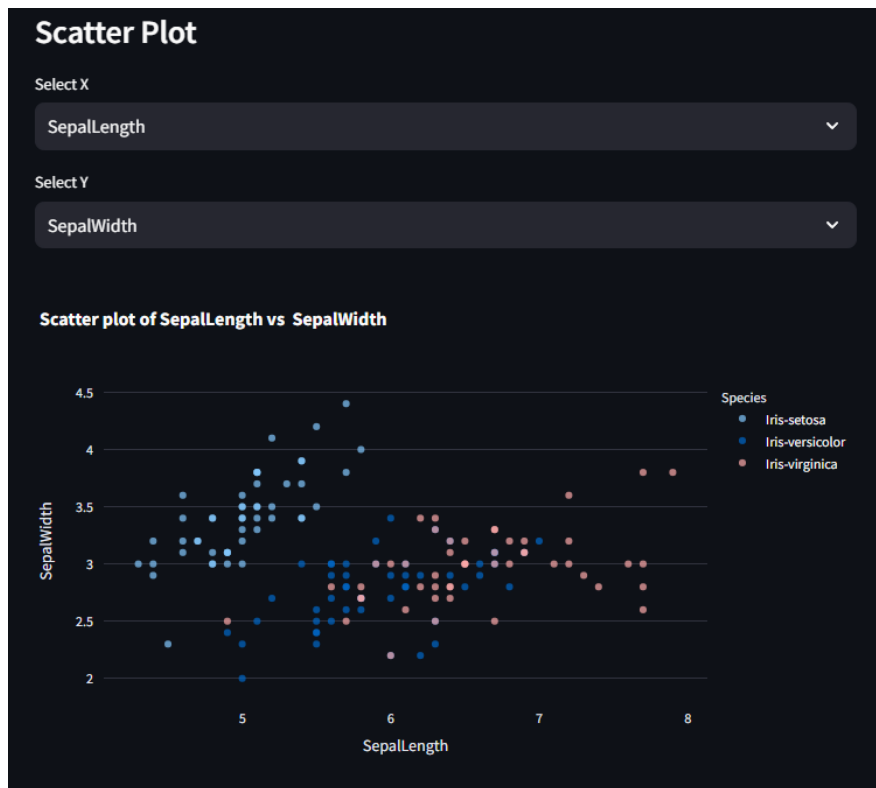
#### Ιστόγραμμα



Σχήμα 7: Ιστόγραμμα

Επιλέγοντας μια στήλη από τα δεδομένα Iris (π.χ. μήκος σεφάλου), το ιστόγραμμα δείχνει πόσο συχνά εμφανίζονται συγκεκριμένες τιμές για αυτή τη μεταβλητή. Τα δεδομένα χρωματίζονται ανάλογα με την κατηγορία τους, επιτρέποντας την ανάλυση της κατανομής κάθε κατηγορίας.

## Διάγραμμα Διασποράς

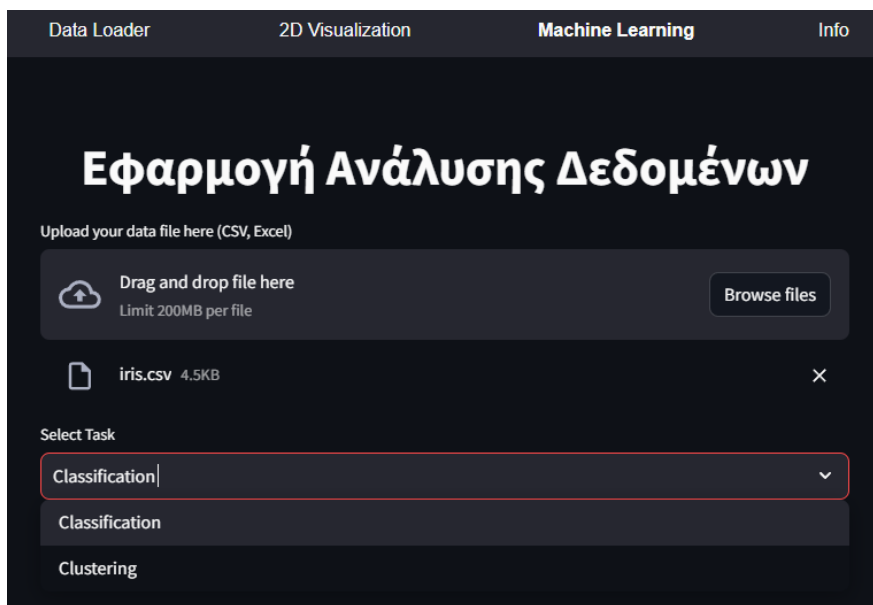


Σχήμα 8: Διάγραμμα Διασποράς

Επιλέγοντας δύο στήλες από τα δεδομένα Iris (π.χ. μήκος σεφάλου και πλάτος σεφάλου), το διάγραμμα διασποράς δείχνει πώς οι τιμές αυτών των μεταβλητών σχετίζονται μεταξύ τους. Τα δεδομένα χρωματίζονται σύμφωνα με τις κατηγορίες τους

## 5.4 Μηχανική Μάθηση

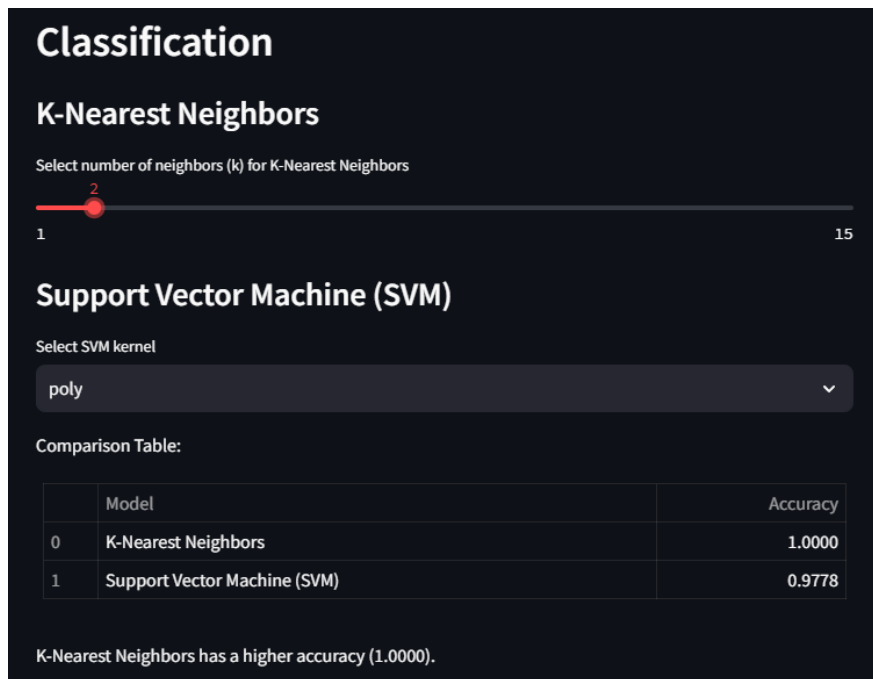
Στην τρίτη σελίδα "Machine Learning" ο χρήστης μπορεί να χρησιμοποιήσει αλγορίθμους μηχανικής μάθησης έχοντας δύο επιλογές: Αλγορίθμους Κατηγοριοποίησης ή Αλγορίθμους Ομαδοποίησης. Αυτοί οι αλγόριθμοι βγάζουν ποσοστό επιτυχίας και γίνεται σύγκριση μεταξύ τους έτσι ώστε ο χρήστης να γνωρίσει ποιος είναι πιο συμβατός με τα δεδομένα του.



Σχήμα 9: Machine Learning



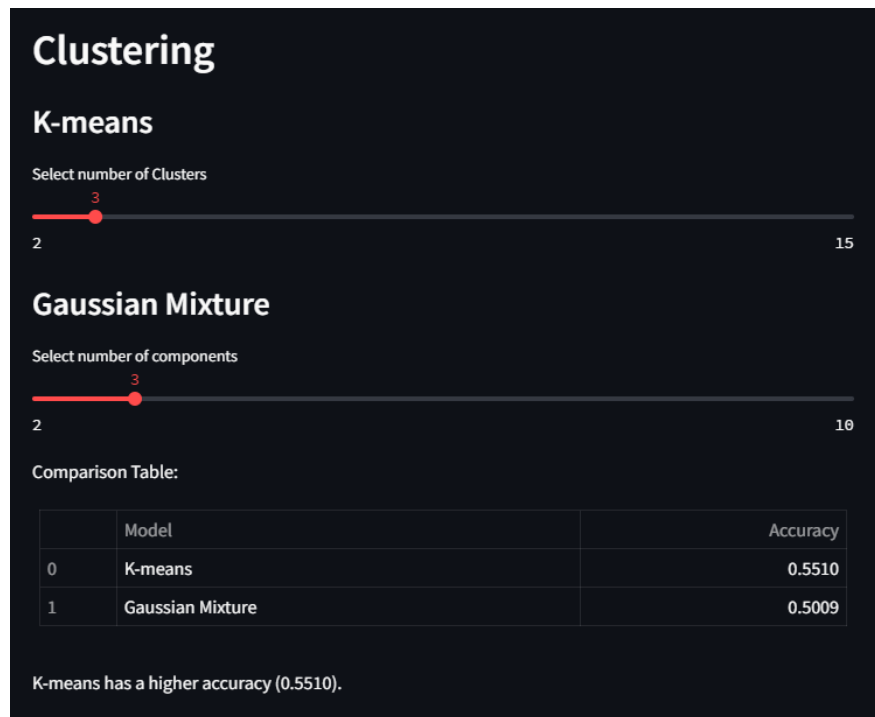
## Παράδειγμα Κατηγοριοποίησης



Σχήμα 10: Classification

**KNN VS SVM:** Το KNN επιτυγχάνει υψηλότερη ακρίβεια (1.0) λόγω της απλότητας και του καθαρού διαχωρισμού στο σύνολο δεδομένων Iris. Το SVM με πολυωνυμικό πυρήνα (0.97) είναι ελαφρώς λιγότερο ακριβές λόγω της πολυπλοκότητας του.

## Παράδειγμα Ομαδοποίησης



Σχήμα 11: Clustering

**K-means VS GM:** Το K-means αποδίδει λίγο καλύτερα (0.55) λόγω της προσέγγισής του με κεντροειδή, ενώ το Gaussian Mixture (0.50) δυσκολεύεται με τη φύση των δεδομένων αν αυτά δεν ακολουθούν κανονική κατανομή.

## 6 Κύκλος Ζωής Έκδοσης Λογισμικού

Για την ανάπτυξη της web-based εφαρμογής για εξόρυξη και ανάλυση δεδομένων, θα χρησιμοποιήσουμε το Agile. Αυτό το μοντέλο μας επιτρέπει συνεχείς βελτιώσεις και προσαρμογές μέσω επαναλαμβανόμενων κύκλων ανάπτυξης (sprints).

Παρακάτω περιγράφεται το προσαρμοσμένο Agile μοντέλο που θα ακολουθήσουμε:

- **Ανάλυση των Απαιτήσεων**

- Συλλογή και ανάλυση των απαιτήσεων των χρηστών μέσω διάφορων τρόπων συλλογής feedback.
- Καθορισμός των βασικών λειτουργιών και χαρακτηριστικών της εφαρμογής.

- **Ανάπτυξη Λειτουργιών**

- Σχεδιασμός των tasks και των στόχων για κάθε χρονικό όριο που έχει τεθεί (sprint).
- Παρακολούθηση της προόδου και επίλυση των προβλημάτων.

- **Testing και Υποστήριξη**

- Δοκιμές από τους χρήστες για την εξασφάλιση της ικανοποίησης των απαιτήσεων.
- Δοκιμή λογισμικού για την εξασφάλιση της ορθότητας του κώδικα.
- Παροχή υποστήριξης και εκπαίδευσης προς στους χρήστες.

Για να διατεθεί η εφαρμογή σε ευρύ κοινό, θα εξασφαλίσουμε την ευχρηστία, την αξιοπιστία και την ασφάλεια της εφαρμογής. Θα παρέχουμε τεκμηρίωση και οδηγίες χρήσης, και θα διασφαλίσουμε ότι η εφαρμογή μπορεί να υποστηρίξει πολλούς χρήστες ταυτόχρονα μέσω δοκιμών φορτίου και βελτιστοποίησης απόδοσης. Με αυτήν την προσέγγιση, η ανάπτυξη της εφαρμογής θα είναι συνεχής και ευέλικτη, επιτρέποντας την προσαρμογή στις ανάγκες των χρηστών και την αντιμετώπιση πιθανών προβλημάτων που μπορεί να προκύψουν κατά τη διάρκεια της ανάπτυξης.

## 7 Περιγραφή της συνεισφοράς κάθε μέλους της ομάδας

### **Data Loader Tab:**

Στην δημιουργία του Data Loader Tab είχε συνεισφορά ο Νικόλαος Τρυπάκης.

### **2D Visualization Tab:**

Στην δημιουργία του 2D Visualization Tab είχε συνεισφορά ο Νικόλαος Τρυπάκης.

### **Machine Learning Tab:**

Στην δημιουργία του Machine Learning Tab είχαν συνεισφορά ο Ορέστης Ραφαήλ Μακρής και ο Στέφανος Σφηναρολάκης.

### **Info Tab:**

Στην δημιουργία του Info Tab είχε συνεισφορά ο Στέφανος Σφηναρολάκης.

### **UML:**

Στην δημιουργία του Info Tab είχε συνεισφορά ο Ορέστης Ραφαήλ Μακρής.

### **Μοντέλο Κύκλου Ζωής Λογισμικού και Latex:**

Στην δημιουργία του Μοντέλου Κύκλου Ζωής του Λογισμικού και έκθεσης Latex είχε συνεισφορά ο Νικόλαος Τρυπάκης.

## 8 Github Repositories

Νικόλαος Τρυπάκης inf2021229

- Profile: [inf2021229](#)
- Project: [App Repository](#)
- Latex files: [Latex Repository](#)
- UML diagram: [UML Repository](#)

Στέφανος Σφηναρολάκης inf2021218

- Profile: [inf2021218](#)
- Project: [App Repository](#)
- Latex files: [Latex Repository](#)
- UML diagram: [UML Repository](#)

Ορέστης Παφαήλ Μακρής inf2021129

- Profile: [inf2021129](#)
- Project: [App Repository](#)
- Latex files: [Latex Repository](#)
- UML diagram: [UML Repository](#)

Ομάδας

- Profile: [Team](#)
- Project: [App Repository](#)
- Latex files: [Latex Repository](#)
- UML diagram: [UML Repository](#)