doi:
Github Repository:
https://github.com/Brigade-01/Software-Engineering
Report

REPORT

Software Engineering

Spyridon Eftychios Kokotos,[1*] Florian Dima[2*] and Nikolaos Balatos[3*]

[1,2,3]Department of Informatics, Ionian University, Plateia Tsirigoti, 7, 49100, Ionioi Nisoi, Greece
[*]{skokotos - inf2021098, inf2021044, inf2021151}@ionio.gr
FOR PUBLISHER ONLY Received on Date Month Year; revised on Date Month Year; accepted on Date Month Year

Abstract

This project focuses on building a web-based application for data mining and analysis, leveraging either Streamlit or RShiny. Key features include tabular data loading, structured representation, 2D visualization, machine learning algorithm comparison, comprehensive result analysis, and project information. Also, this project aims to provide a user-friendly interface for in-depth data exploration and algorithm evaluation, catering to a wide audience.

Key words: Data Mining, Data Analysis, 2D Visualization, Machine Learning

Introduction

As part of the **Software Engineering** course for the academic year 2023-2024, this project involves the development of a sophisticated web-based data analysis application utilizing machine learning (ML) algorithms. The primary aim is to create a comprehensive platform capable of processing tabular data, applying various ML algorithms, and providing detailed comparative analysis of their accuracy and performance.

The primary objective of this project is for students to leverage existing online code resources to develop a sophisticated, user-centric application. This application will be designed to accept tabular data, process it through multiple machine learning algorithms, and output the corresponding results. Additionally, the application will compare the accuracy levels of each algorithm, providing a comprehensive analysis of their performance.

Design of the Application

The implemented application encompasses a range of sophisticated features tailored for comprehensive data analysis and visualization. It offers a user-friendly interface facilitating seamless exploration of datasets. Within its framework, users can access a dedicated 2D Visualization Tab, enabling data visualization through advanced algorithms such as PCA and t-SNE, alongside the presentation of Exploratory Data Analysis (EDA) plots.

Furthermore, the Classification Tab empowers users to execute data classification tasks utilizing robust algorithms like Random Forest and SVC (Support Vector Classifier), while the Clustering Tab facilitates data grouping through the application of K-Means and Agglomerative Clustering algorithms, thus enhancing the depth and efficiency of data processing and interpretation.

Information about the Tabs

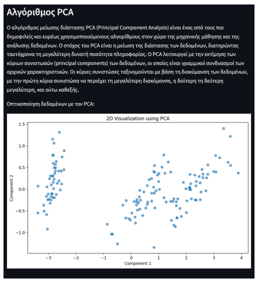
2D Visualization Tab

Exploratory Data Analysis (EDA) diagrams are visual tools used to summarize the main characteristics of a dataset, often with visual methods. These diagrams play a critical role in data analysis by helping to uncover underlying patterns, detect anomalies, test hypotheses, and check assumptions through a graphical representation. Common types of EDA diagrams include histograms, box plots, scatter plots, and bar charts, each serving different purposes such as displaying distribution, variability, relationships, and categorical data insights.

By transforming raw data into visual formats, EDA diagrams facilitate a better understanding of the data's structure and guide subsequent analytical or modeling steps, ensuring a robust and informed approach to data science and statistical analysis.

A Box Plot, or whisker plot, displays data distribution based on a five-number summary: minimum, first quartile (Q1), median, third quartile (Q3), and maximum. It highlights central tendency, spread, and skewness, and identifies outliers, making it easy to compare distributions across different datasets.

A Density Diagram, or density plot, provides a smoothed curve that represents the probability distribution of a continuous variable. It offers a clear view of the data's shape, central tendency, and variability without the noise of histogram bins.

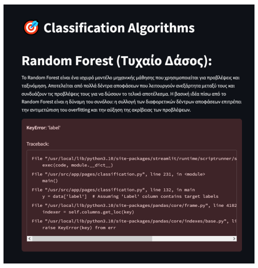


Αλγόριθμος PCA

Ο αλγόριθμος μείωσης διάστασης PCA (Principal Component Analysis) είναι ένας από τους πιο δημοφιλείς και ευρέως χρησιμοποιούμενους αλγορίθμους στον χώρο της μηχανικής μάθησης και της ανάλυσης δεδομένων. Ο στόχος του PCA είναι η μείωση της διάστασης των δεδομένων, διατηρώντας ταυτόχρονα τη μεγαλύτερη δυνατή ποσότητα πληροφορίας. Ο PCA λειτουργεί με την εκτίμηση των κύριων συνιστωσών (principal components) των δεδομένων, οι οποίες είναι γραμμικοί συνδυασμοί των αρχικών χαρακτηριστικών. Οι κύριες συνιστώσες ταξινομούνται με βάση τη διακύμανση των δεδομένων, με την πρώτη κύρια συνιστώσα να περιέχει τη μεγαλύτερη διακύμανση, η δεύτερη τη δεύτερη μεγαλύτερη, και ούτω καθεξής.

Οπτικοποίηση δεδομένων με τον PCA:

Fig. 1. The results of visualization after processing them through the PCA algorithm.



Classification Algorithms

Random Forest (Τυχαίο Δάσος):

Το Random Forest είναι ένα ισχυρό μοντέλο μηχανικής μάθησης που χρησιμοποιείται για προβλέψεις και ταξινόμηση. Αποτελείται από πολλά δέντρα αποφάσεων που λειτουργούν ανεξάρτητα μεταξύ τους και συνδυάζουν τις προβλέψεις τους για να δώσουν το τελικό αποτέλεσμα. Η βασική ιδέα πίσω από το Random Forest είναι η δύναμη του συνόλου: η συλλογή των διαφορετικών δέντρων αποφάσεων επιτρέπει την αντιμετώπιση του overfitting και την αύξηση της ακρίβειας των προβλέψεων.

KeyError: 'label'

Traceback:

```
File "/usr/local/lib/python3.10/site-packages/streamlit/runtime/scriptrunner/s
    exec(code, module.__dict__)
File "/usr/src/app/pages/classification.py", line 231, in <module>
    main()
File "/usr/src/app/pages/classification.py", line 132, in main
    y = data['label']  # Assuming 'Label' column contains target labels
File "/usr/local/lib/python3.10/site-packages/pandas/core/frame.py", line 4102
    indexer = self.columns.get_loc(key)
File "/usr/local/lib/python3.10/site-packages/pandas/core/indexes/base.py", li
    raise KeyError(key) from err
```

Fig. 2. The **KeyError: 'label"** error that show no label was found in the .csv file

Right above, you can find one of the many figures (that will be included in the report in a specific section at the end) related to the PCA algorithm.

A Heatmap uses color to represent the values of a matrix, allowing for a quick visual summary of data. It is useful for displaying correlations, identifying patterns, and highlighting areas of high or low intensity, simplifying complex data for easier interpretation.

Classification Tab

The Classification Tab of the application showcases two powerful machine learning algorithms: Random Forest and Support Vector Classifier (SVC). Random Forest, a robust ensemble learning technique, is adept at making predictions and classifications by aggregating the outputs of multiple decision trees. Its strength lies in the diversity of these trees, which operate independently and combine their predictions to yield more accurate results.

By mitigating overfitting and enhancing prediction accuracy, Random Forest effectively addresses various classification challenges. On the other hand, Support Vector Classifier (SVC) is a formidable algorithm for binary classification tasks. It endeavors to delineate data points in feature space using an optimal hyperplane, thereby creating a clear demarcation between classes. SVC's flexibility and efficacy in diverse classification scenarios, coupled with its resilience to high-dimensional data, render it a versatile tool for classification tasks.

Through interactive interfaces, users can explore the intricacies of these algorithms, analyze their predictive performance, and gain valuable insights for decision-making and problem-solving.

It's imperative to note that in the categorization tab, the .csv file must incorporate the respective label for the process to advance, as it's deemed indispensable in machine learning. Consequently, if file upload is sanctioned without the inclusion of the label, the following error will manifest on the specific tab:

KeyError: 'label'

Clustering Tab

The Clustering Tab of the application presents two clustering algorithms: K-Means and Hierarchical Clustering (Agglomerative Clustering). K-Means is a widely used unsupervised learning algorithm for partitioning data into a predetermined number of clusters. It iteratively assigns each data point to the nearest centroid, optimizing the centroid positions until convergence. K-Means finds applications in pattern recognition, group analysis, and as a preprocessing step for other algorithmic applications.

On the other hand, Hierarchical Clustering constructs hierarchical groupings of data points. It starts by considering each data point as a separate cluster and then iteratively merges the closest clusters until a single cluster or the desired number of clusters is achieved. Hierarchical Clustering comes in two main types: Agglomerative, which starts with small clusters and merges them, and Divisive, which starts with one large cluster and divides it.

Through interactive interfaces, users can explore these clustering algorithms, analyze their clustering results, and compare their performance using evaluation metrics such as Silhouette Score, Calinski-Harabasz Index, Davies-Bouldin Index, Dunn Index, and Between-Cluster Sum of Squares (BSS).

Team Contributions & Information Tab

In the Information tab, detailed insights regarding the developmental endeavors of the scientific mode are delineated. The team, spearheaded by ***Florian Dima***, encompasses pivotal tasks including the conceptualization of the **Clustering Tab**, crafting **UML Diagrams**, orchestrating the **Software Development Life Cycle**, and the formulation of **Dockerfile**.

Spyridon Eftychios Kokotos assumes a multifaceted role, spearheading the establishment of the **Brigade-01 Organization** and the **Software-Engineering repository** on **GitHub**, orchestrating the **Implementation of the Classification Tab**, and **facilitating seamless data integration from CSV to pertinent tabs**. Additionally, ***Kokotos*** undertakes the onus of drafting **User Guidelines** and the **Final Report**.

Nikolaos Balatos, on the other hand, undertakes the technical implementations of the **2D Visualization** and **Home Tabs**, alongside infusing the **Info Tab** with requisite functionalities. Furthermore, ***Balatos*** orchestrates the seamless **linkage of data from CSV to the Home and 2D Visualization Tabs**, complementing the efforts with **UML Diagrams** and the **Software Development Life Cycle** adherence.

Agile Model for the Software Version Lifecycle of the Web-Based Data Mining and Analysis Application

For the development of the web-based data mining and analysis application using Streamlit, we propose the utilization of the Agile model. The Agile model is characterized by its flexibility, enabling continuous improvement and adaptation through iterative development cycles (sprints) and continuous feedback loops. Below, we outline the key stages of the software version lifecycle based on Agile:

1. Initial Planning:

- **Requirement Gathering**: Collection of functional and non-functional requirements from stakeholders.
- **Purpose and Goals Definition**: Determination of the application's overall purpose and the core goals to be achieved.
- **Architectural Design:** Development of the application's architectural design, including UML diagrams.

2. Backlog Creation:

- **Product Backlog:** Recording of all requirements and features in the form of user stories in the product backlog.
- **Sprint Backlog:** Selection of user stories to be completed in each sprint from the product backlog.

3. Sprint Planning:

- **Sprint Duration Definition:** Setting the duration of each sprint (typically 1-2 weeks).
- **Sprint Goals:** Defining the sprint goals and the user stories to be completed.

4. Sprint Execution:

- **Development:** Implementation of the application's features and functionalities according to the user stories.
- **Daily Stand-ups:** Daily team meetings to review progress and address any impediments.

5. Review and Feedback:

- **Sprint Review:** Presentation of sprint results to stakeholders and collection of feedback.
- **Sprint Retrospective:** Review of the sprint process by the development team and identification of areas for improvement.

6. Continuous Integration and Delivery:

- **Code Integration:** Continuous integration of new code into the central repository and execution of automated tests.
- **Delivery:** Delivery of the completed functionality at the end of each sprint.

7. Evaluation and Improvement:

- **Metrics Analysis:** Evaluation of algorithm and application performance based on performance metrics.
- **Continuous Improvement:** Implementation of enhancements based on feedback and analysis results.

8. Final Release and Support

- **Release:** Creation of the final release of the application and delivery to end users.
- **Support:** Provision of support and maintenance for bug fixes and addition of new features.

Adaptation to Team Collaboration:

- **GitHub Usage:** Code management and team collaboration through GitHub, utilizing pull requests and code reviews.
- **Docker:** Use of Docker for application development and deployment, ensuring consistency in development and production environments.

Final Conclusion:

The use of the Agile model enables the team to remain flexible and respond quickly to changes in requirements and user feedback. Through continuous feedback and iterative development cycles, it ensures that the final product fully meets user needs and is of high quality.

Installation Instructions

Option 1: Docker Image Pull:

1. Pull the Docker Image

- Retrieve the Docker image from Docker Hub by executing the following command in your terminal: '*docker pull spyridonkokotos/brigade-01-sw:latest*'

2. Run the Docker Container

- Deploy the Docker container using the following command: '*docker run -d –name brigade-01-sw -p 8501:8501 brigade-01-sw:latest*

3. Access the Application

- Open your preferred web browser and navigate to *http://localhost:8501* to interact with the application.

Option 2: Build the local Dockerfile:

1. Clone the Repository

- Clone the repository from GitHub to your local machine using the following command: '*git clone https://github.com/Brigade-01/Software-Engineering.git*'

2. Navigate to the Project Directory:

- Change to the project directory where the source files are located: '*cd Software-Engineering/src/*'

3. Build the Docker Image:

- Construct the Docker image locally by executing: '*docker build -t brigade-01-sw:latest .*'

4. Run the Docker Container:

- Deploy the Docker container with the following command: '*docker run -d –name brigade-01-sw -p 8501:8501 brigade-01-sw:latest*'

5. Access the Application:

- Open your preferred web browser and navigate to http://localhost:8501 to interact with the application.

Basic Usage:

To efficiently utilize the application for data analysis and algorithm comparison, begin by uploading your desired file. Select either a .csv or .xls file by clicking the "Browse Files" button. Await a confirmation message indicating the successful upload of your file. Upon successful upload and validation of the file's content, you may proceed to explore the application's features.

Navigate through the various tabs using the menu on the left side of the interface. This allows you to access different functionalities and apply included algorithms to test their performance on your dataset. This structured approach ensures a seamless and effective user experience within the application.

Classification Results:

SVC performed better since the accuracy was slightly higher than for the random forest

UML Diagrams:

Use Case Diagram:

The Use Case Diagram presents a high-level view of the user's interaction with the web application, delineating the primary functionalities:

- File Upload: The user initiates interaction by uploading a file, which is central to subsequent processes.
- 2D Visualization: Once a file is uploaded, the user can generate 2D visualizations. This includes the generation of diagrams using dimensionality reduction algorithms and Exploratory Data Analysis (EDA).
- Dimensionality Reduction Algorithms Diagrams: These diagrams simplify data for visualization purposes.
- EDA Diagrams: These diagrams assist in initial data analysis by summarizing the main characteristics.
- Machine Learning Implementation: The user can implement machine learning algorithms, which include:

1. Clustering Algorithm Implementation: Algorithms such as hierarchical clustering and K-nearest neighbors (KNN) are utilized to group data points.
2. Classification Algorithm Implementation: Algorithms like Random Forest and Support Vector Classification (SVC) are used to classify data points .
3. Results & Comparison Printing: The results from clustering and classification implementations are compared and printed for analysis.

Class Diagram:

The Class Diagram provides a detailed structural view of the application by defining its classes, attributes, and relationships:

- FILE Class: Attributes: file size (up to 200MB) file type (supports .csv and .xlsx formats) This class is pivotal as it interfaces with other classes to provide the data required for processing.
- 2D Visualization Class: Methods: TSNE(file): Applies t-Distributed Stochastic Neighbor Embedding for visualization. PCA(file): Applies Principal Component Analysis for dimensionality reduction. EDAdiagrams(file): Generates EDA diagrams to summarize data characteristics.
- ML ALGORITHMS Class: Method: CompareResults(): Compares the results of different machine learning algorithms. This class acts as a parent class for clustering and classification algorithms.
- Clustering Algorithms Class (inherits from ML ALGORITHMS): Methods: HIER-CLUSTERING(file): Implements hierarchical clustering. KNN(file): Implements the K-nearest neighbors algorithm. CompareResults(results): Specific implementation of the result comparison.
- Classification Algorithms Class (inherits from ML ALGORITHMS): Methods: RND-FOREST(file): Implements the Random Forest algorithm. SVC-CLASS(file): Implements the Support Vector Classification algorithm. CompareResults(results): Specific implementation of the result comparison.

These diagrams collectively elucidate the application's functionality and structure, illustrating the user interactions and the modular design of classes facilitating various data analysis and machine learning tasks. This detailed representation ensures clarity in understanding the workflow and integration of different analytical methods within the application.

Reporting Bugs:

To report a discovered bug, it is imperative to open an issue on GitHub, as previously outlined. Accurate bug reporting is crucial for effective resolution and system improvement. When documenting a bug, please provide a detailed description, including steps to reproduce the issue, observed behavior, and expected behavior.

Additionally, including relevant system information and any error messages encountered will facilitate a more efficient troubleshooting process. This systematic approach ensures that developers can quickly identify and address the problem, thereby enhancing the overall functionality and reliability of the software.

License

Authors contributions statement

The above program is a creation of:

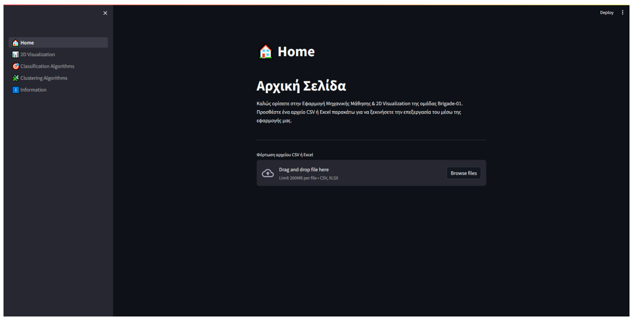
Florian Dima, Nikolaos Balatos & Spyridon Eftychios Kokotos

and you can use it for your personal projects or further develop it as long as you always give credit to the creators.

First Name	Last Name	Registry Number	Semester	Academic Email
Florian	Dima	inf2021044	6th	inf2021044@ionio.gr
Nikolaos	Balatos	inf2021151	6th	inf2021151@ionio.gr
Spyridon - Eftychios	Kokotos	inf2021098	6th	inf2021098@ionio.gr skokotos@ionio.gr

Table 1. Contact Details of Students

Figures



🏠 Home

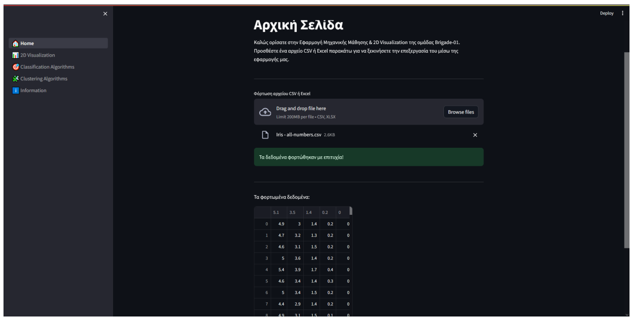
Αρχική Σελίδα

Καλώς ορίσατε στην Εφαρμογή Μηχανικής Μάθησης & 2D Visualization της ομάδας Brigade-01. Προσθέστε ένα αρχείο CSV ή Excel παρακάτω για να ξεκινήσετε την επεξεργασία του μέσω της εφαρμογής μας.

Φόρτωση αρχείου CSV ή Excel

Drag and drop file here
Limit 200MB per file • CSV, XLSX

Fig. 3. Welcome Screen of the Application



Αρχική Σελίδα

Καλώς ορίσατε στην Εφαρμογή Μηχανικής Μάθησης & 2D Visualization της ομάδας Brigade-01. Προσθέστε ένα αρχείο CSV ή Excel παρακάτω για να ξεκινήσετε την επεξεργασία του μέσω της εφαρμογής μας.

Φόρτωση αρχείου CSV ή Excel

Drag and drop file here
Limit 200MB per file • CSV, XLSX

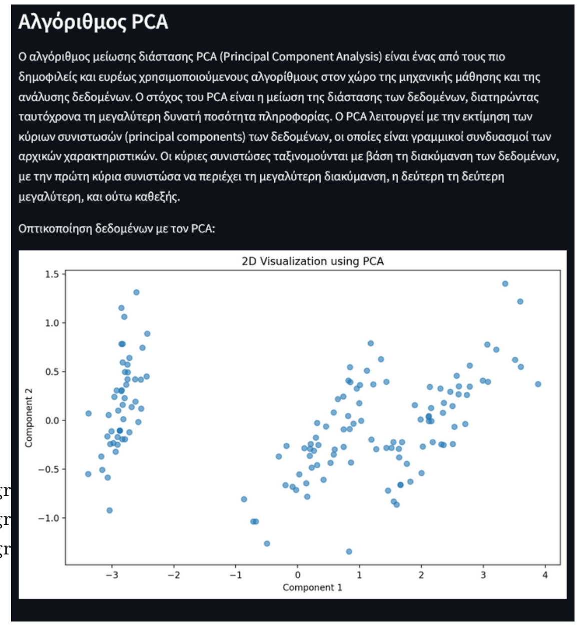
Iris - all-numbers.csv 2.6KB

Τα δεδομένα φορτώθηκαν με επιτυχία!

Τα φορτωμένα δεδομένα:

	5.1	3.5	1.4	0.2	0
0	4.9	3	1.4	0.2	0
1	4.7	3.2	1.3	0.2	0
2	4.6	3.1	1.5	0.2	0
3	5	3.6	1.4	0.2	0
4	5.4	3.9	1.7	0.4	0
5	4.6	3.4	1.4	0.3	0
6	5	3.4	1.5	0.2	0
7	4.4	2.9	1.4	0.2	0
8	4.9	3.1	1.5	0.1	0

Fig. 4. The Successful loading of the .csv dataset

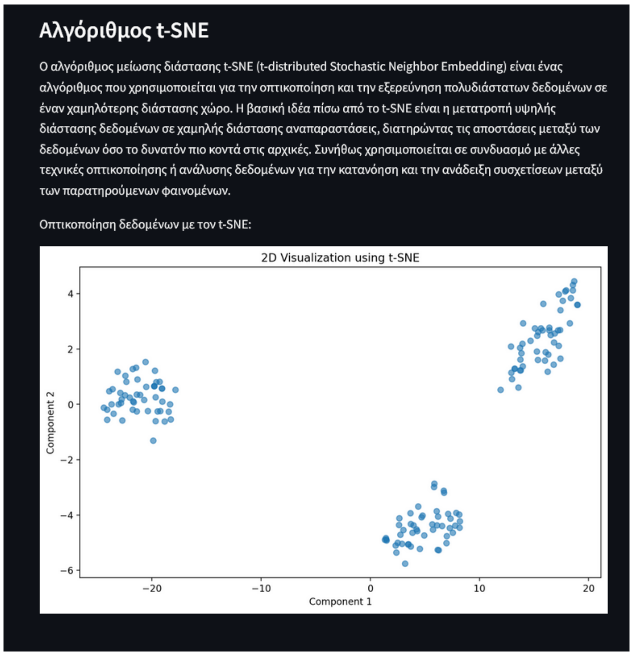


Αλγόριθμος PCA

Ο αλγόριθμος μείωσης διάστασης PCA (Principal Component Analysis) είναι ένας από τους πιο δημοφιλείς και ευρέως χρησιμοποιούμενους αλγορίθμους στον χώρο της μηχανικής μάθησης και της ανάλυσης δεδομένων. Ο στόχος του PCA είναι η μείωση της διάστασης των δεδομένων, διατηρώντας ταυτόχρονα τη μεγαλύτερη δυνατή ποσότητα πληροφορίας. Ο PCA λειτουργεί με την εκτίμηση των κύριων συνιστωσών (principal components) των δεδομένων, οι οποίες είναι γραμμικοί συνδυασμοί των αρχικών χαρακτηριστικών. Οι κύριες συνιστώσες ταξινομούνται με βάση τη διακύμανση των δεδομένων, με την πρώτη κύρια συνιστώσα να περιέχει τη μεγαλύτερη διακύμανση, η δεύτερη τη δεύτερη μεγαλύτερη, και ούτω καθεξής.

Οπτικοποίηση δεδομένων με τον PCA:

Fig. 5. The results of visualization after processing them through the PCA algorithm

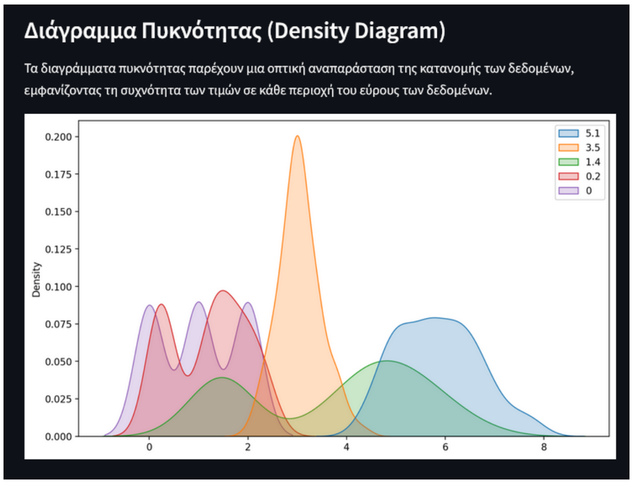


Αλγόριθμος t-SNE

Ο αλγόριθμος μείωσης διάστασης t-SNE (t-distributed Stochastic Neighbor Embedding) είναι ένας αλγόριθμος που χρησιμοποιείται για την οπτικοποίηση και την εξερεύνηση πολυδιάστατων δεδομένων σε έναν χαμηλότερης διάστασης χώρο. Η βασική ιδέα πίσω από το t-SNE είναι η μετατροπή υψηλής διάστασης δεδομένων σε χαμηλής διάστασης αναπαραστάσεις, διατηρώντας τις αποστάσεις μεταξύ των δεδομένων όσο το δυνατόν πιο κοντά στις αρχικές. Συνήθως χρησιμοποιείται σε συνδυασμό με άλλες τεχνικές οπτικοποίησης ή ανάλυσης δεδομένων για την κατανόηση και την ανάδειξη συσχετίσεων μεταξύ των παρατηρούμενων φαινομένων.

Οπτικοποίηση δεδομένων με τον t-SNE:

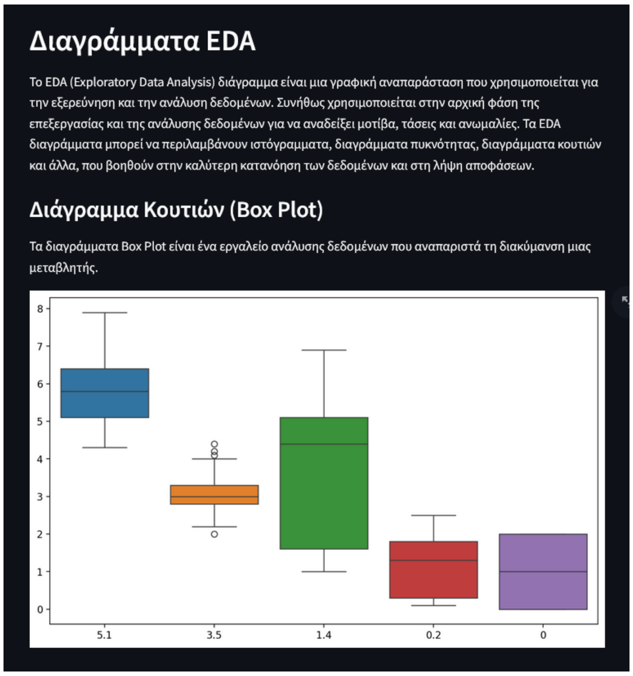
Fig. 6. The results of visualization after processing them through the t-SNE algorithm



Διάγραμμα Πυκνότητας (Density Diagram)

Τα διαγράμματα πυκνότητας παρέχουν μια οπτική αναπαράσταση της κατανομής των δεδομένων, εμφανίζοντας τη συχνότητα των τιμών σε κάθε περιοχή του εύρους των δεδομένων.

Fig. 8. Density Diagram



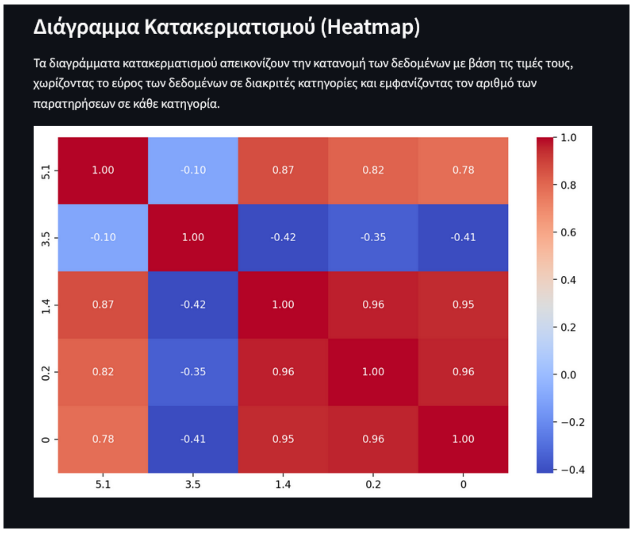
Διαγράμματα EDA

Το EDA (Exploratory Data Analysis) διάγραμμα είναι μια γραφική αναπαράσταση που χρησιμοποιείται για την εξερεύνηση και την ανάλυση δεδομένων. Συνήθως χρησιμοποιείται στην αρχική φάση της επεξεργασίας και της ανάλυσης δεδομένων για να αναδείξει μοτίβα, τάσεις και ανωμαλίες. Τα EDA διαγράμματα μπορεί να περιλαμβάνουν ιστόγραμματα, διαγράμματα πυκνότητας, διαγράμματα κουτιών και άλλα, που βοηθούν στην καλύτερη κατανόηση των δεδομένων και στη λήψη αποφάσεων.

Διάγραμμα Κουτιών (Box Plot)

Τα διαγράμματα Box Plot είναι ένα εργαλείο ανάλυσης δεδομένων που αναπαριστά τη διακύμανση μιας μεταβλητής.

Fig. 7. EDA-Boxplot Diagram

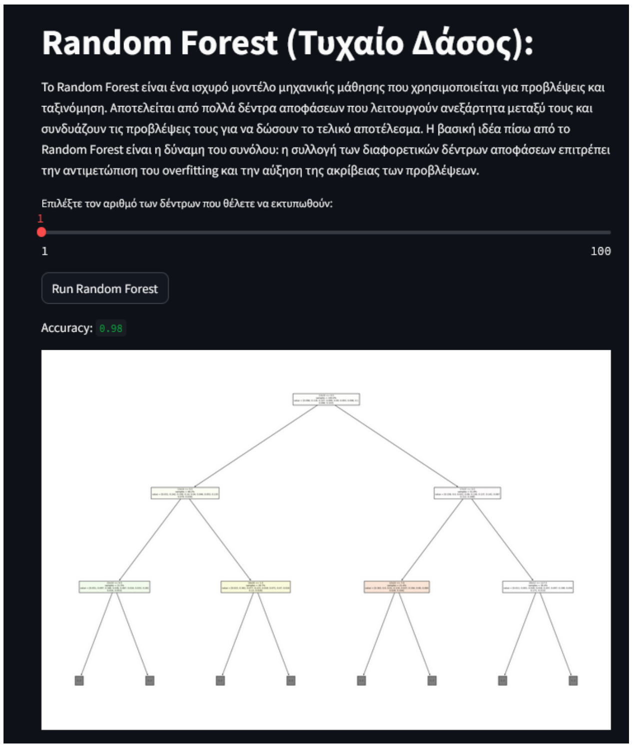


Διάγραμμα Κατακερματισμού (Heatmap)

Τα διαγράμματα κατακερματισμού απεικονίζουν την κατανομή των δεδομένων με βάση τις τιμές τους, χωρίζοντας το εύρος των δεδομένων σε διακριτές κατηγορίες και εμφανίζοντας τον αριθμό των παρατηρήσεων σε κάθε κατηγορία.

	5.1	3.5	1.4	0.2	0
5.1	1.00	-0.10	0.87	0.82	0.78
3.5	-0.10	1.00	-0.42	-0.35	-0.41
1.4	0.87	-0.42	1.00	0.96	0.95
0.2	0.82	-0.35	0.96	1.00	0.96
0	0.78	-0.41	0.95	0.96	1.00

Fig. 9. Heatmap Diagram



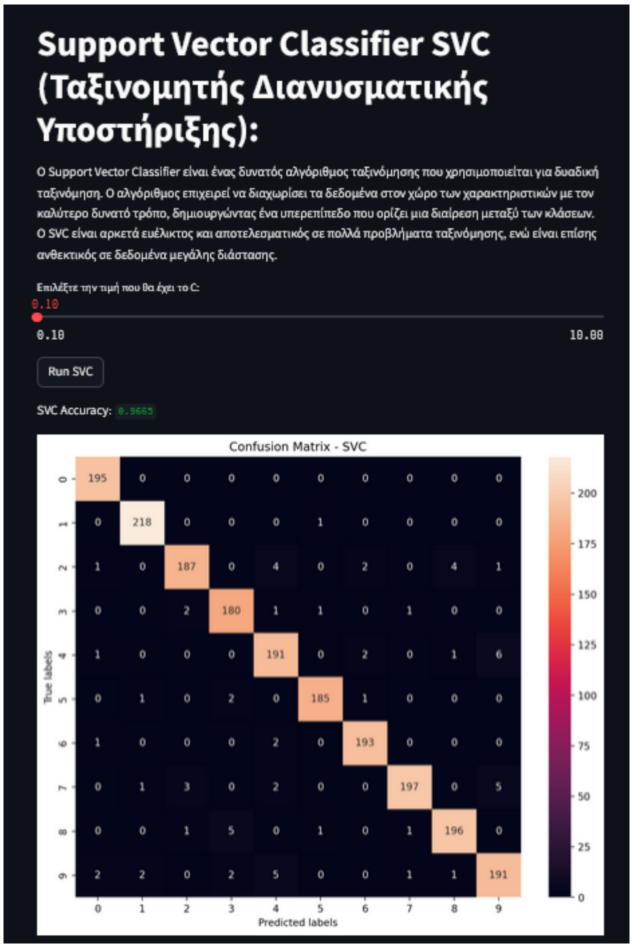
Random Forest (Τυχαίο Δάσος):

Το Random Forest είναι ένα ισχυρό μοντέλο μηχανικής μάθησης που χρησιμοποιείται για προβλέψεις και ταξινόμηση. Αποτελείται από πολλά δέντρα αποφάσεων που λειτουργούν ανεξάρτητα μεταξύ τους και συνδυάζουν τις προβλέψεις τους για να δώσουν το τελικό αποτέλεσμα. Η βασική ιδέα πίσω από το Random Forest είναι η δύναμη του συνόλου: η συλλογή των διαφορετικών δέντρων αποφάσεων επιτρέπει την αντιμετώπιση του overfitting και την αύξηση της ακρίβειας των προβλέψεων.

Επιλέξτε τον αριθμό των δέντρων που θέλετε να εκτυπωθούν: 1

Accuracy: 0.98

Fig. 10. Random Forest Diagram



Support Vector Classifier SVC (Ταξινομητής Διανυσματικής Υποστήριξης):

Ο Support Vector Classifier είναι ένας δυνατός αλγόριθμος ταξινόμησης που χρησιμοποιείται για δυαδική ταξινόμηση. Ο αλγόριθμος επιχειρεί να διαχωρίσει τα δεδομένα στον χώρο των χαρακτηριστικών με τον καλύτερο δυνατό τρόπο, δημιουργώντας ένα υπερεπίπεδο που ορίζει μια διαίρεση μεταξύ των κλάσεων. Ο SVC είναι αρκετά ευέλικτος και αποτελεσματικός σε πολλά προβλήματα ταξινόμησης, ενώ είναι επίσης ανθεκτικός σε δεδομένα μεγάλης διάστασης.

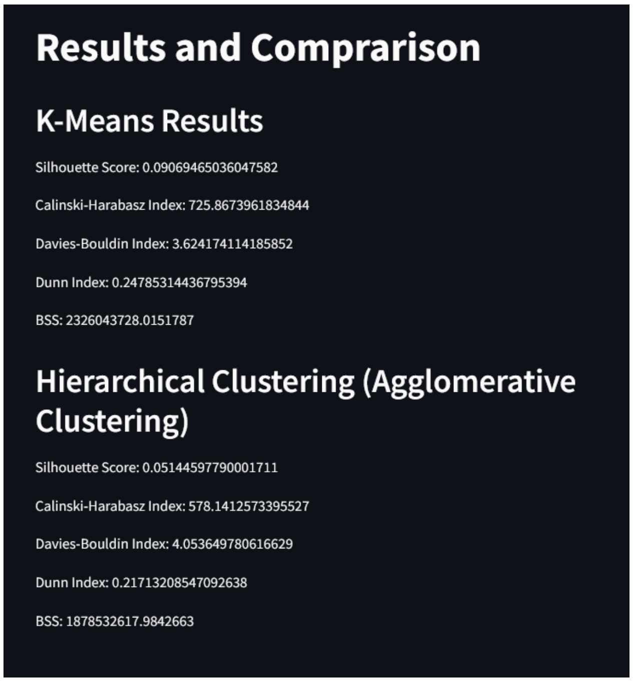
Επιλέξτε την τιμή που θα έχει το C: 0.10

SVC Accuracy: 0.9665

Confusion Matrix - SVC

True labels \ Predicted labels	0	1	2	3	4	5	6	7	8	9
0	195	0	0	0	0	0	0	0	0	0
1	0	218	0	0	0	1	0	0	0	0
2	1	0	187	0	4	0	2	0	4	1
3	0	0	2	180	1	1	0	1	0	0
4	1	0	0	0	191	0	2	0	1	6
5	0	1	0	2	0	185	1	0	0	0
6	1	0	0	0	2	0	193	0	0	0
7	0	1	3	0	2	0	0	197	0	5
8	0	0	1	5	0	1	0	1	196	0
9	2	2	0	2	5	0	0	1	1	191

Fig. 11. SVC Diagram



Results and Comprarison

K-Means Results

Silhouette Score: 0.09069465036047582

Calinski-Harabasz Index: 725.8673961834844

Davies-Bouldin Index: 3.624174114185852

Dunn Index: 0.24785314436795394

BSS: 2326043728.0151787

Hierarchical Clustering (Agglomerative Clustering)

Silhouette Score: 0.05144597790001711

Calinski-Harabasz Index: 578.1412573395527

Davies-Bouldin Index: 4.053649780616629

Dunn Index: 0.21713208547092638

BSS: 1878532617.9842663

Fig. 12. The results from the Clustering Tab

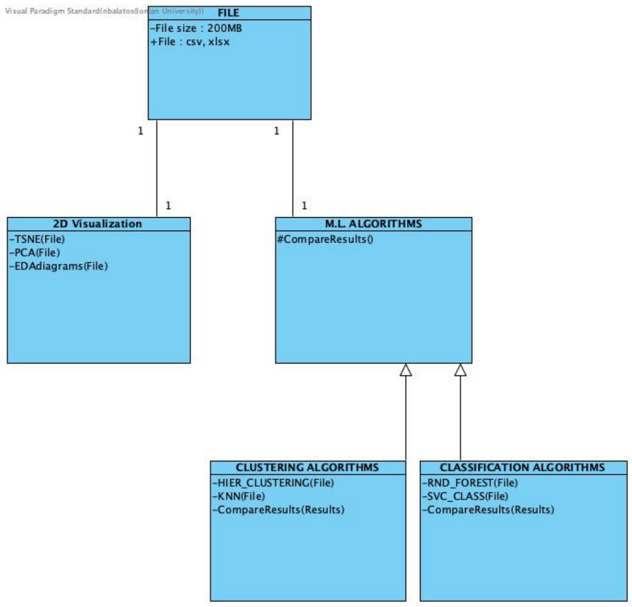


Fig. 13. The Class Diagram of our Application

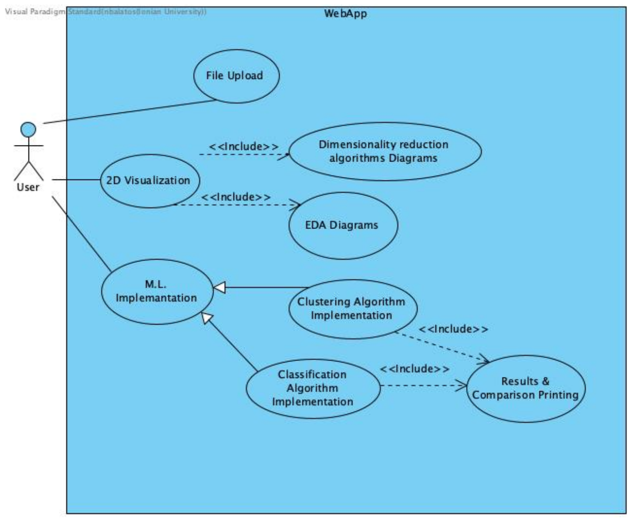

Visual Paradigm Standard(nbalatos(Ionian University))
WebApp
User
File Upload
2D Visualization
<<Include>>
Dimensionality reduction algorithms Diagrams
<<Include>>
EDA Diagrams
M.L. Implemantation
Clustering Algorithm Implementation
<<Include>>
Classification Algorithm Implementation
<<Include>>
Results & Comparison Printing

Fig. 14. The Use Case Diagram of our Application