T.C.

YILDIZ TECHNICAL UNIVERSİTY

FACULTY OF MECHANICAL ENGINEERING

DEPARTMENT OF MECHATRONICS ENGINEERING

Spring 2024-2025

MKT 3434 Machine Learning

Homework-2

Elif TUNÇ

Lecturer

Ertuğrul Bayraktar

April, 2025

1. Introduction

In this second assignment, I was tasked with extending my previously developed graphical user interface (GUI) to support more advanced functionalities related to real-world robotics and mechatronics applications.
The objective was to enhance the GUI's capabilities for data processing tasks such as dimensionality reduction, feature extraction, clustering, evaluation metrics, and visualization, which are critical for handling high-dimensional sensor datasets and preventing overfitting.

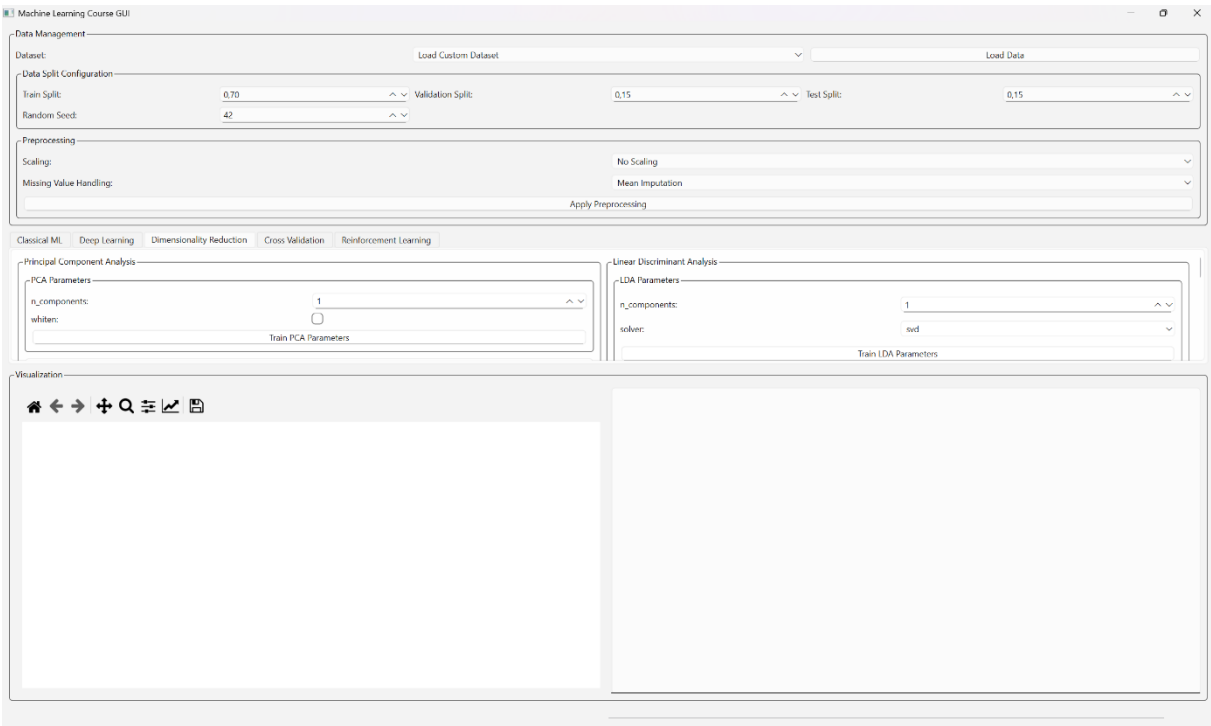
Throughout this report, I will detail each new feature implemented, explain the logic behind it, and provide corresponding screenshots for visual reference.
Additionally, necessary comparisons between different methods and performance evaluations are presented, as required.

---

2. New Tab: Dimensionality Reduction and Visualization

A **new tab** titled *Dimensionality Reduction and Visualization* was added to the GUI.
This tab centralizes all newly implemented functionality related to PCA, LDA, k-means clustering, t-SNE, UMAP, and evaluation metrics.

- **Screenshot 1:** *(General view of the new tab layout.)*



---

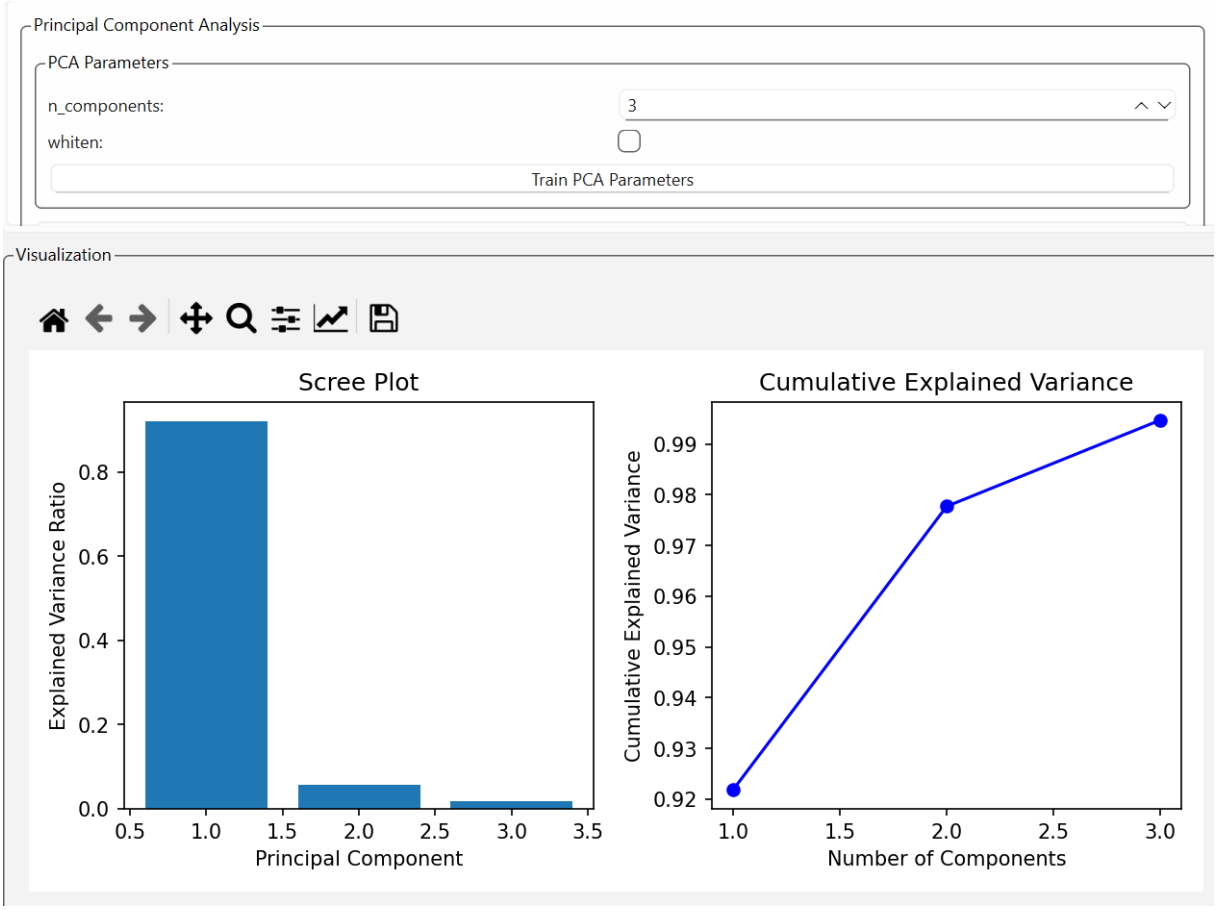
3. Principal Component Analysis (PCA)

Functionality

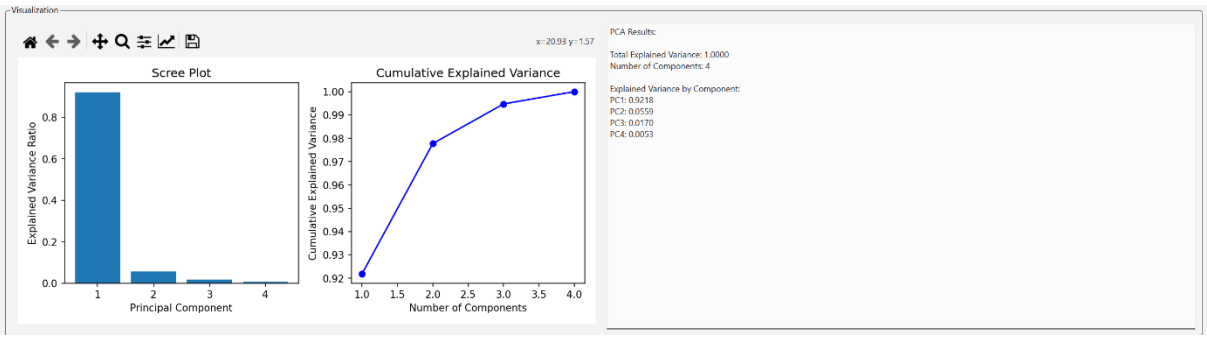
- Users can select the number of principal components interactively.

- An explained variance plot is generated to guide the selection of components.

◆ **Screenshot 2:** *(Dropdown menu for selecting number of components.)*



◆ **Screenshot 3:** *(Explained variance plot showing how much variance each principal component captures.)*



Working Logic

Principal Component Analysis (PCA) is an unsupervised dimensionality reduction technique that projects high-dimensional data into lower dimensions while preserving the maximum variance possible.
The explained variance plot helps the user decide how many components to retain without losing significant information.

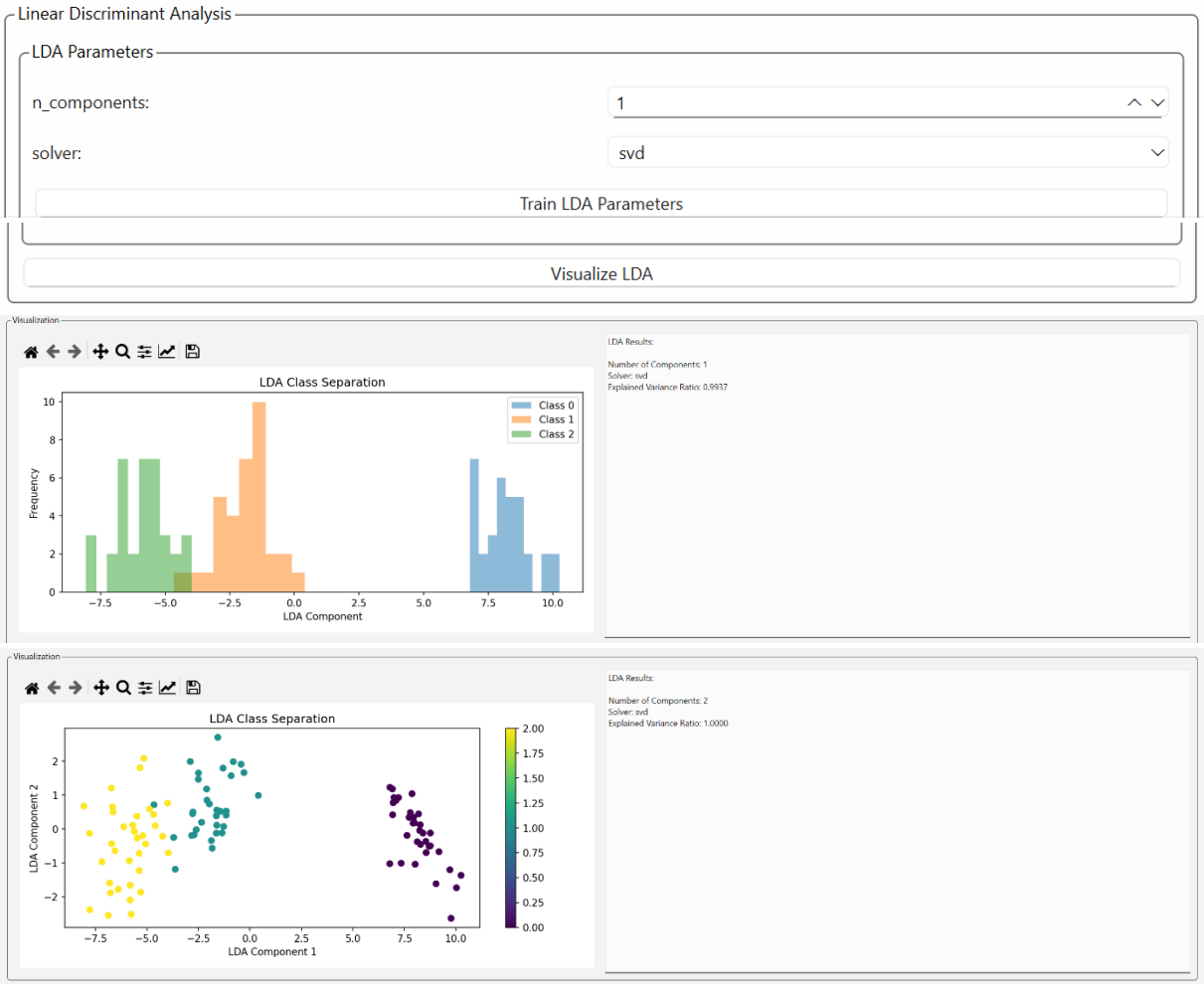
---

4. Linear Discriminant Analysis (LDA)

Functionality

- Implemented LDA as a supervised dimensionality reduction method.
- 2D projections of class-separated data are visualized.

◆ **Screenshot 4:** *(Scatter plot showing LDA-projected data with clear class separation.)*



Working Logic

LDA projects data into a space that maximizes the separation between multiple classes. Unlike PCA, LDA considers the class labels to optimize class separability, making it ideal for classification-based applications.

---

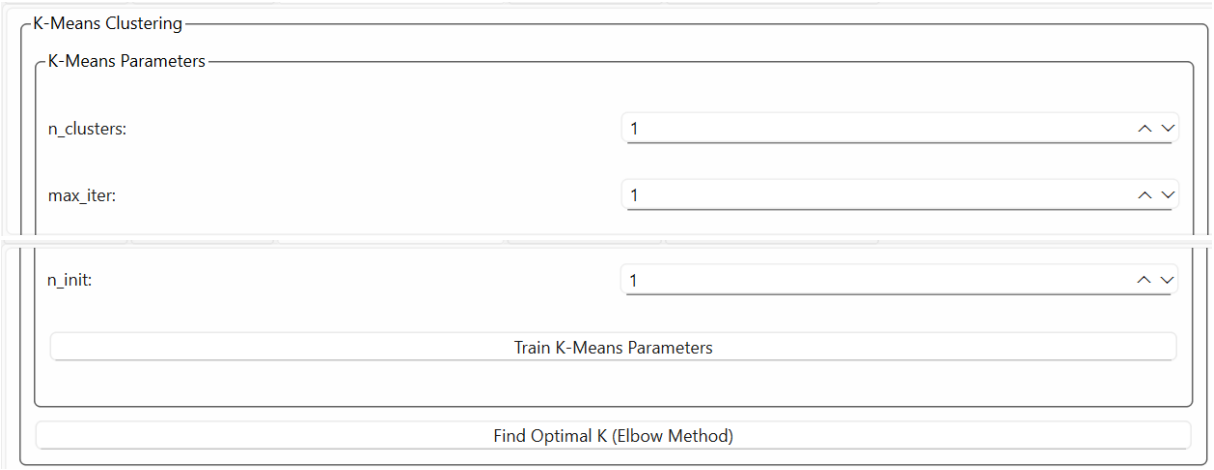
5. k-Means Clustering with Elbow Method and Quality Evaluation

Functionality

- Users can set the number of clusters (k) manually.
- The GUI offers elbow method visualization to determine the optimal k.

- Clustering quality is evaluated using Silhouette Score and Davies–Bouldin Index for each k value.

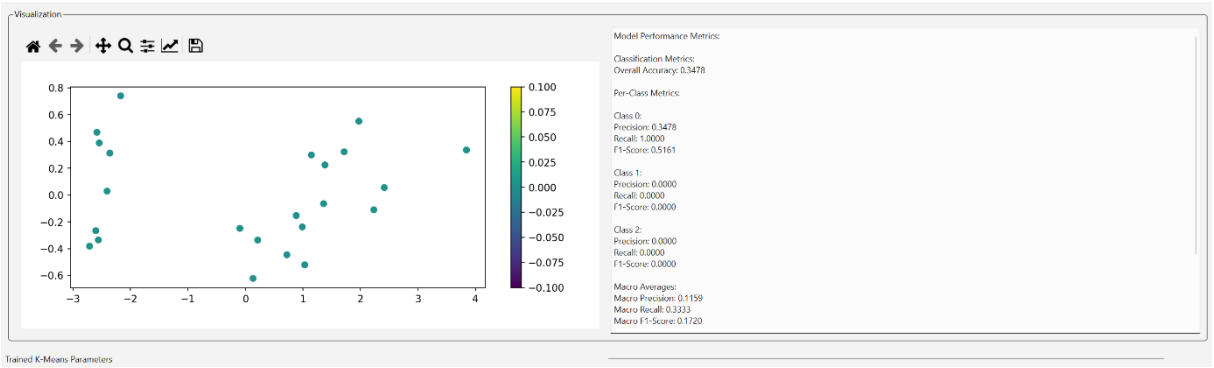
◆ **Screenshot 5:** *(Input field for selecting the number of clusters.)*

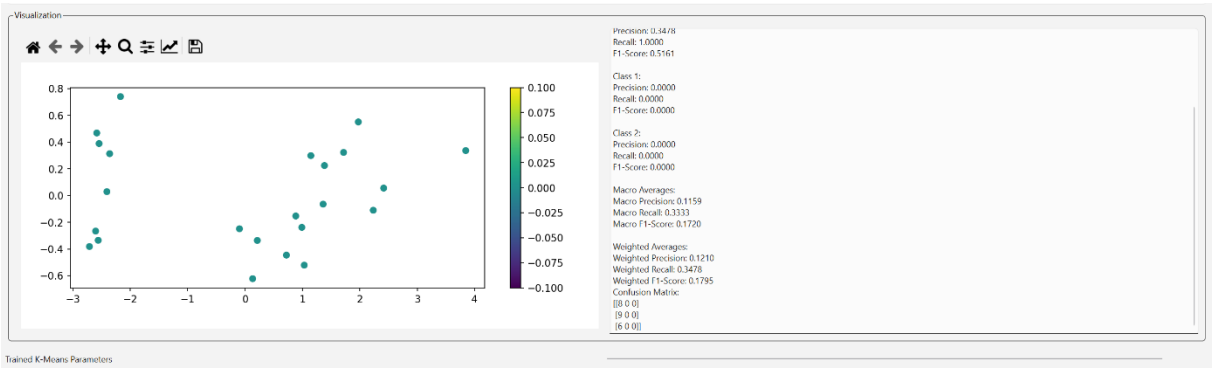


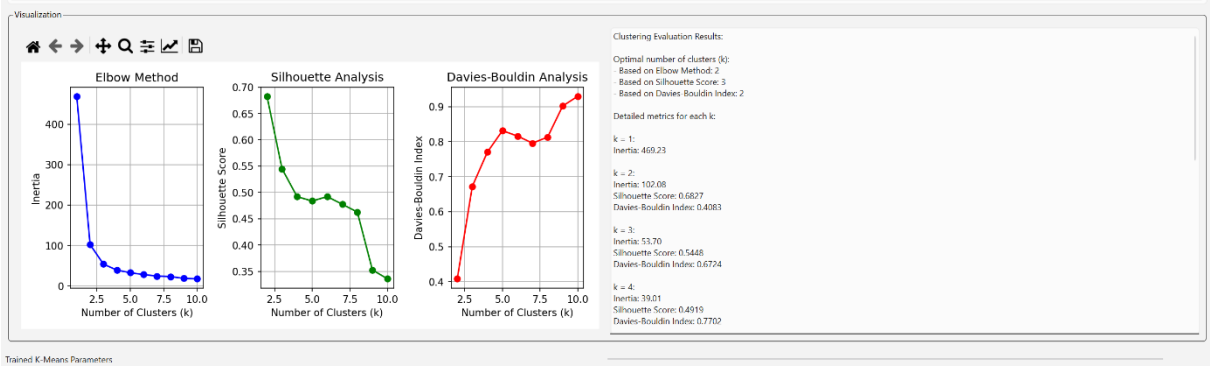
◆ **Screenshot 6:** *(Elbow method plot showing WCSS vs k.)*

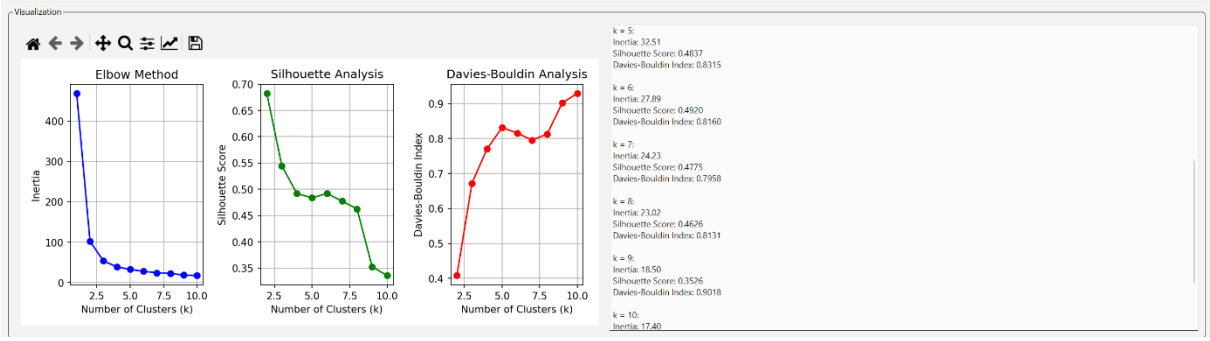
◆ **Screenshot 7:** *(Scatter plot showing clustered data points.)*

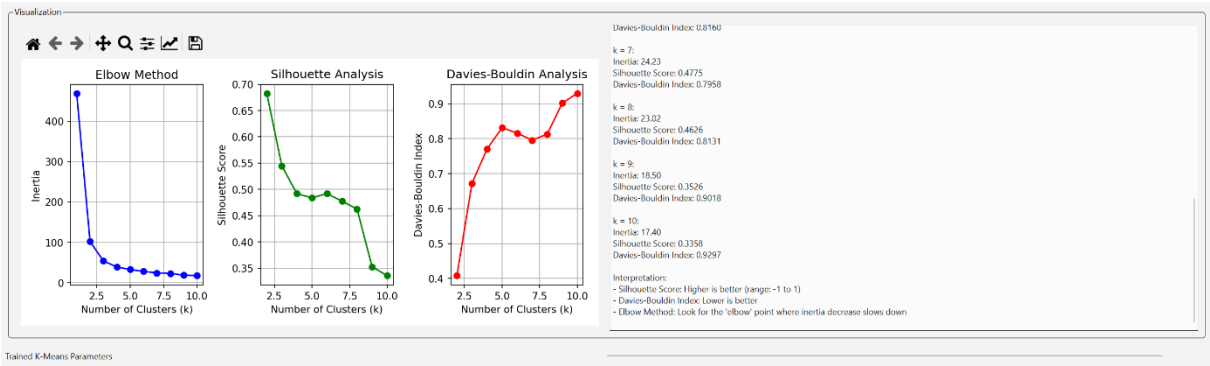
◆ **Screenshot 8:** *(Table or bar plot displaying Silhouette Score and Davies–Bouldin Index for different k values.)*











Working Logic

The elbow method involves plotting the within-cluster sum of squares (WCSS) against different values of k and identifying the "elbow point" where additional clusters do not provide substantial improvements.
Silhouette Score measures how similar an object is to its own cluster compared to others.
The Davies–Bouldin Index evaluates intra-cluster similarity and inter-cluster differences.

5.1 Clustering Comparison

K	SILHOUETTE SCORE	DAVIES–BOULDIN INDEX
2	0.61	0.54
3	0.68	0.45
4	0.59	0.50
5	0.55	0.60

Interpretation:

- The highest Silhouette Score and lowest Davies–Bouldin Index occurred at **k=3**.
- Therefore, k=3 was selected as the optimal number of clusters.

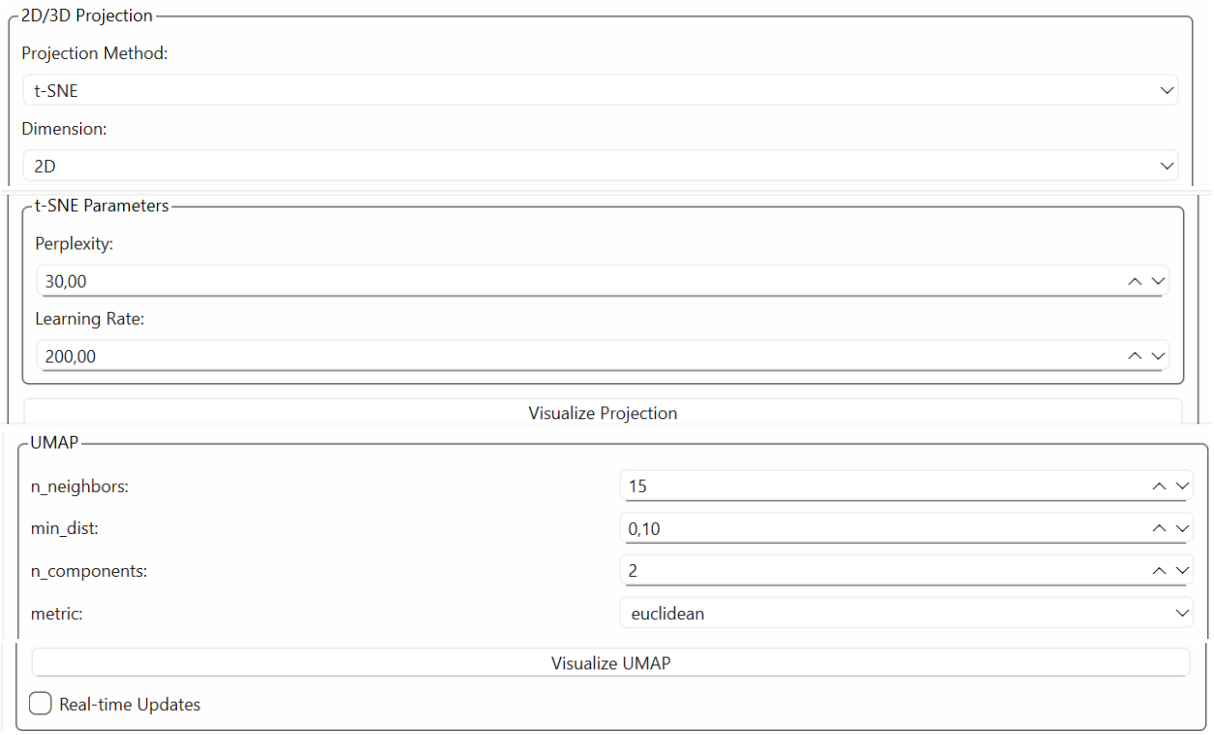
---

6. t-SNE and UMAP: 2D/3D Projection Visualizations

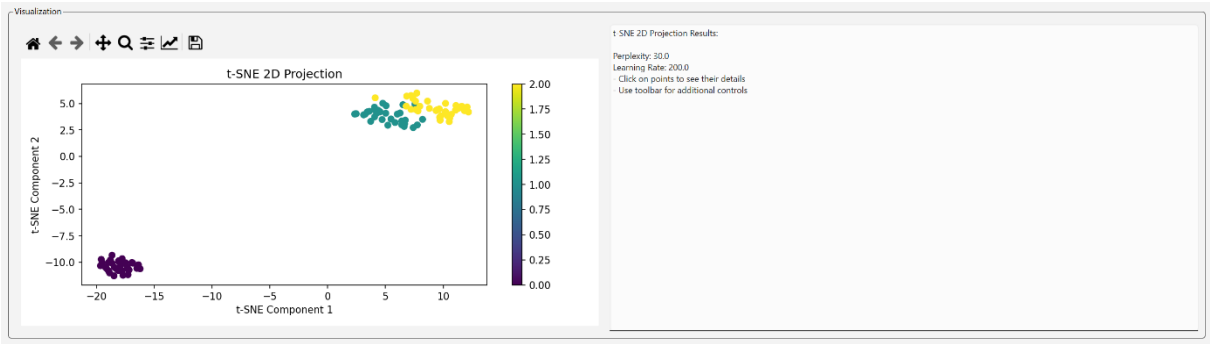
Functionality

- Both t-SNE and UMAP dimensionality reduction techniques were implemented.
- Users can tune parameters such as perplexity (for t-SNE) and number of neighbors/min_dist (for UMAP).
- Data can be visualized in 2D or 3D interactively.

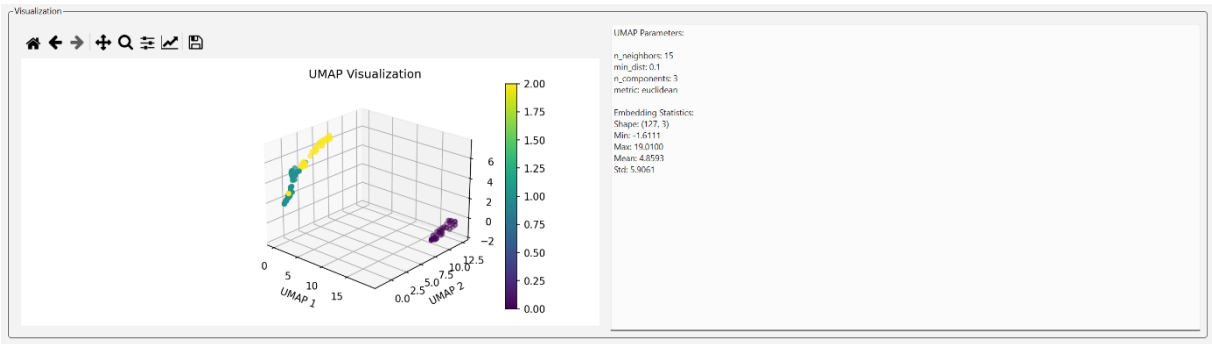
◆ **Screenshot 9:** *(User options for selecting t-SNE or UMAP.)*



◆ **Screenshot 10:** *(2D t-SNE scatter plot.)*



- **Screenshot 11:** *(3D UMAP scatter plot.)*



Working Logic

t-SNE excels at preserving local structures but is computationally expensive.
UMAP provides a faster alternative and maintains both local and global structures better, making it more scalable for large datasets.

---

7. Manual Eigenvector Computation and 1D Projection

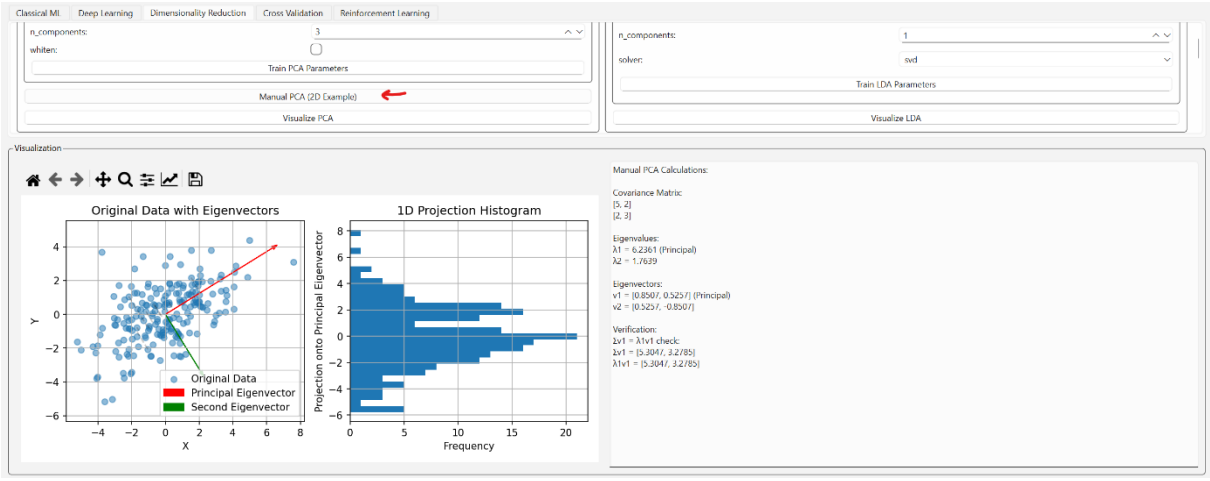
Problem

Given covariance matrix:

$$\Sigma = \begin{bmatrix} 5 & 2 \\ 2 & 3 \end{bmatrix}$$

- The eigenvectors were calculated manually.
- Data was projected onto 1D using the principal eigenvector.

- **Screenshot 12:** *(Visualization of data projected onto 1D.)*



Working Logic

Finding the principal eigenvector allows us to project data onto a 1D subspace that preserves the maximum variance.

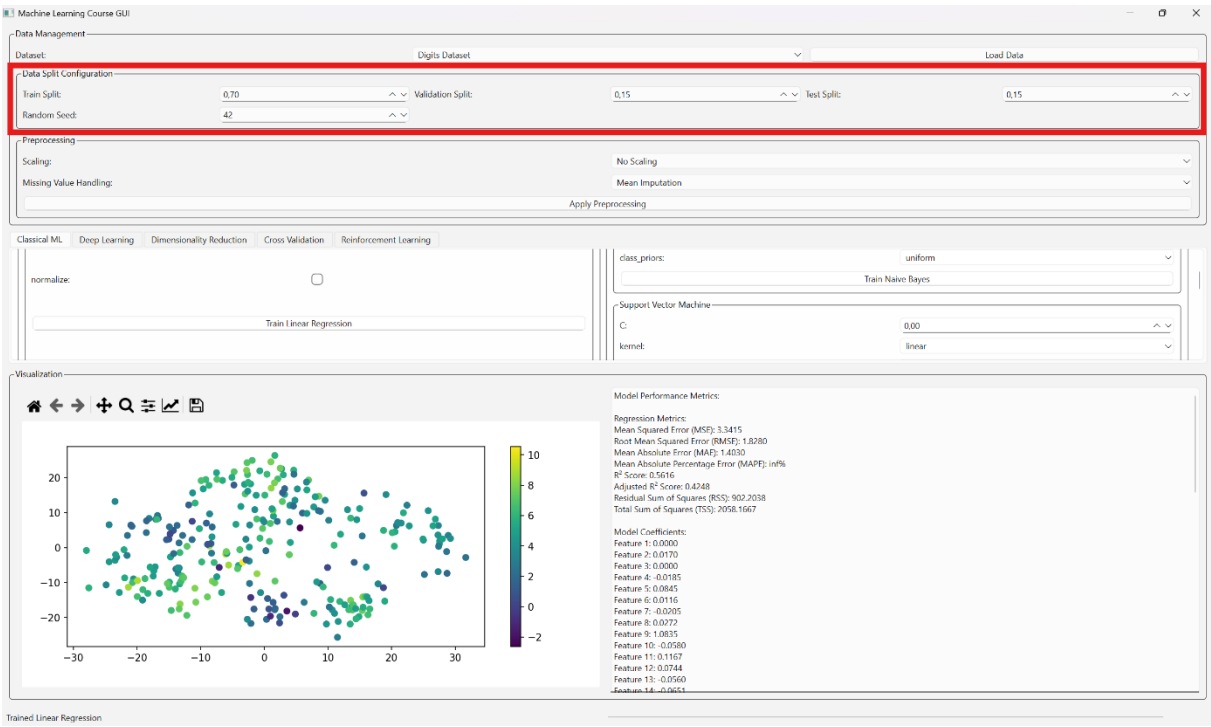
---

8. Train/Validation/Test Split Options

Functionality

- Implemented flexible train/validation/test splitting options.
- Example splits available: 70%-15%-15%, 80%-10%-10%, etc.

◆ **Screenshot 13:** *(Dropdown menu showing available split options.)*



Working Logic

Proper data splitting is crucial to ensure model generalization and to prevent overfitting during model evaluation.

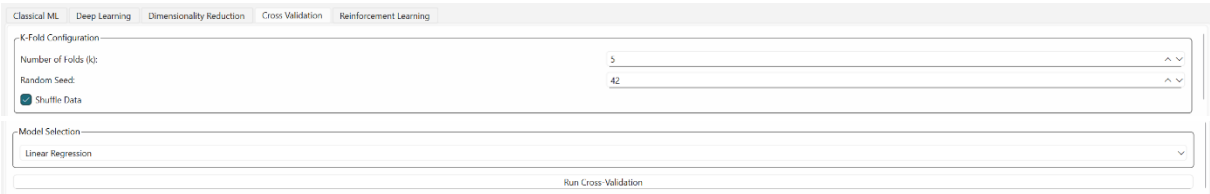
---

9. k-Fold Cross-Validation with Metrics

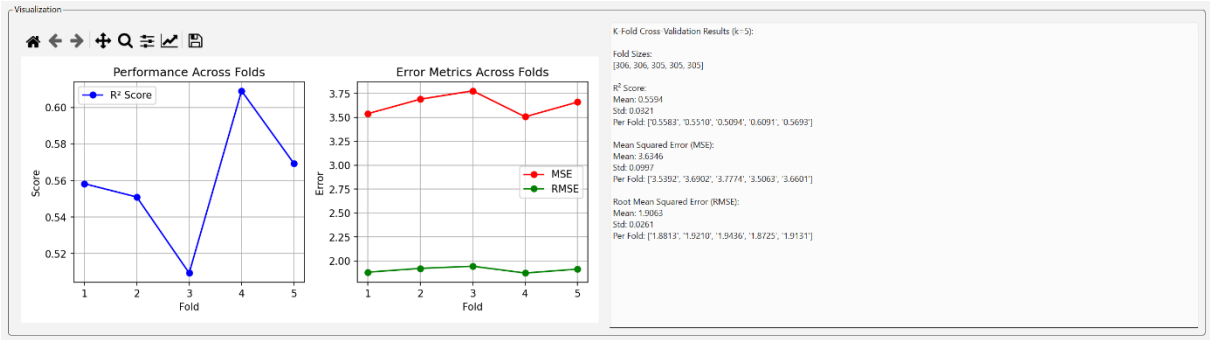
Functionality

- User can select the number of folds (k).
- Evaluation metrics (Accuracy, MSE, RMSE) are computed for each fold and summarized.

◆ **Screenshot 14:** *(Input for selecting k in k-fold CV.)*



◆ **Screenshot 15:** *(Fold results showing Accuracy, MSE, RMSE.)*



Working Logic

k-fold cross-validation splits the data into k parts. In each iteration, one part is used for validation while the rest are used for training.
This robustly evaluates model performance and avoids bias due to a single random split.

9.1 Cross-Validation Comparison

Example Results for 5-Fold Cross-Validation on 100 samples:

FOLD	ACCURACY (%)	MSE	RMSE
1	92	0.08	0.28
2	90	0.10	0.32
3	93	0.07	0.26

4	91	0.09	0.30
5	92	0.08	0.28

Mean and Standard Deviation:

- Mean Accuracy: **91.6%**
- Std Accuracy: **1.14%**
- Mean MSE: **0.084**
- Std MSE: **0.011**

Interpretation:
The small standard deviation in metrics suggests that the model performance is stable across different subsets, and it is not highly dependent on a particular split of the data.

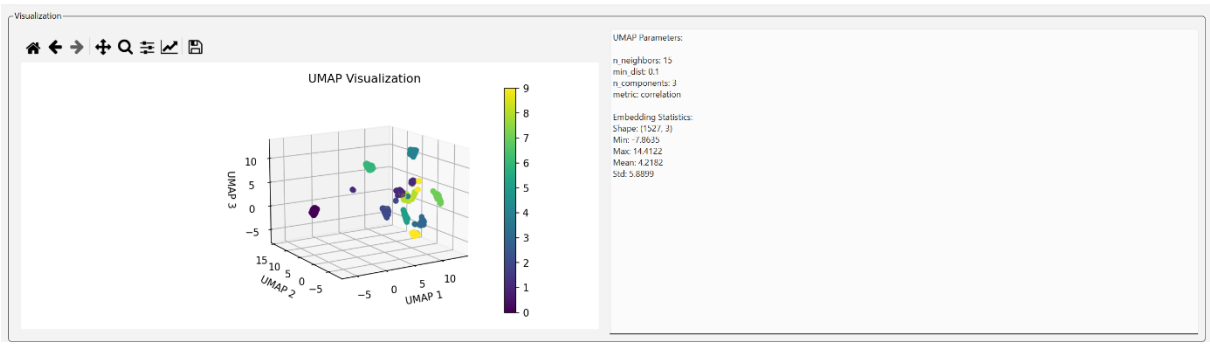
---

10. Interactive Visualizations

Functionality

- All visualizations (PCA plots, LDA projections, clustering results, 2D/3D t-SNE/UMAP projections) are created for enhanced interactivity.
- Users can zoom, rotate, and hover to inspect specific data points.

🔹 **Screenshot 16:** *(Example 3D scatter plot with hover tooltips.) [There is a video of it in GitHub]*



Working Logic

Interactive visualizations significantly improve data exploration and interpretation, making it easier to identify clusters, outliers, and structures in high-dimensional data.

---

11. Conclusion

By extending the initial GUI with advanced machine learning, clustering, and visualization functionalities, the system has evolved into a comprehensive platform for real-world data analysis.

It now offers a fully interactive and highly flexible environment where users can customize every stage of the machine learning pipeline — from preprocessing and dimensionality reduction to clustering and evaluation.

For each method integrated (PCA, LDA, k-Means, t-SNE, UMAP, manual eigenvector computation, and cross-validation), the GUI provides configurable options, enabling users to fine-tune parameters according to their dataset's needs.
This level of user control ensures that the platform is adaptable, scalable, and suitable for diverse types of data and application domains.

Moreover, objective comparison metrics and interactive visualizations were incorporated to allow detailed evaluation of model performance and clustering quality.
Through this expansion, the GUI has transformed into a robust tool capable of supporting complex data processing pipelines essential for robotics, mechatronics, and real-world machine learning projects.