YILDIZ TEKNİK ÜNİVERSİTESİ

1911

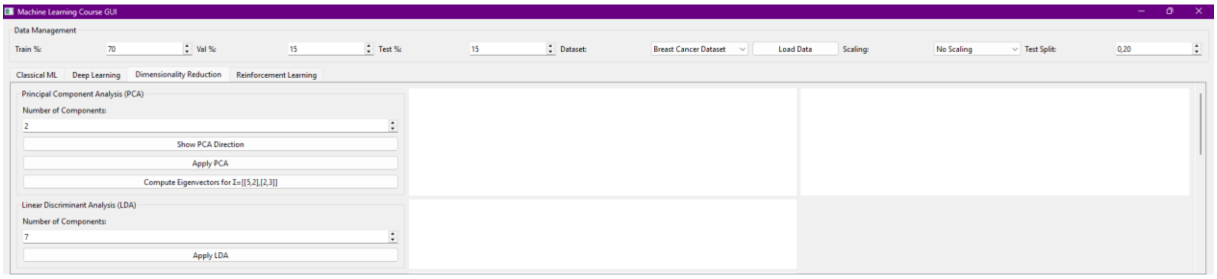
MKT 3434-MACHİNE LEARNİNG

İLKER ARSLAN

21067024

Introduction

In the following figures, various outputs related to dimensionality reduction techniques and eigenvector calculations are presented. Initially, the GUI layout is shown, illustrating the interface designed for applying machine learning operations. Subsequently, the explained variance ratios of principal components derived from PCA are visualized, followed by a scatter plot indicating the first principal direction vector. The computed values of this principal direction are also displayed in a pop-up window. Moreover, eigenvalues and eigenvectors calculated from a manually specified covariance matrix are summarized. These preliminary analyses lay the groundwork for the upcoming steps involving supervised dimensionality reduction and clustering techniques.


Machine Learning Course GUI
Data Management
Load Data
Classical ML
Deep Learning
Dimensionality Reduction
Reinforcement Learning
Principal Component Analysis (PCA)
Show PCA Direction
Apply PCA
Compute Eigenvectors for Σ=[[5,2],[2,3]]
Linear Discriminant Analysis (LDA)
Apply LDA

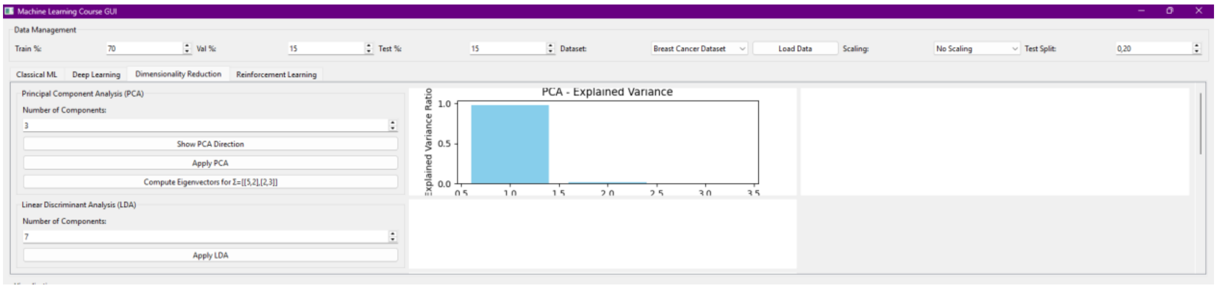
Train %: 70
Val %: 15
Test %: 15
Dataset: Breast Cancer Dataset
Scaling: No Scaling
Test Split: 0,20

Number of Components: 2

Number of Components: 7

Şekil 1 Initial view of the Machine Learning Course GUI

Initial view of the GUI before any dimensionality reduction or clustering operation is applied.



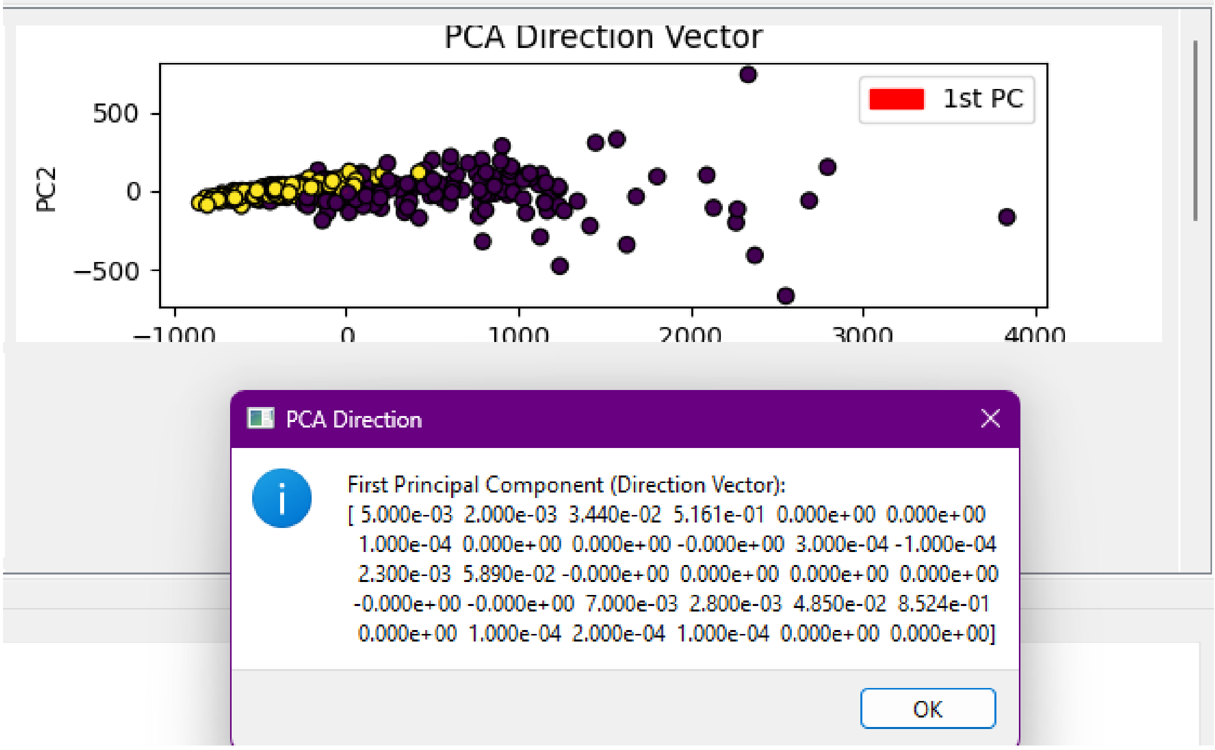
Train %: 70
Val %: 15
Test %: 15
Dataset: Breast Cancer Dataset
Scaling: No Scaling
Test Split: 0,20

Number of Components: 3

Number of Components: 7

Şekil 2 PCA Explained Variance Ratio

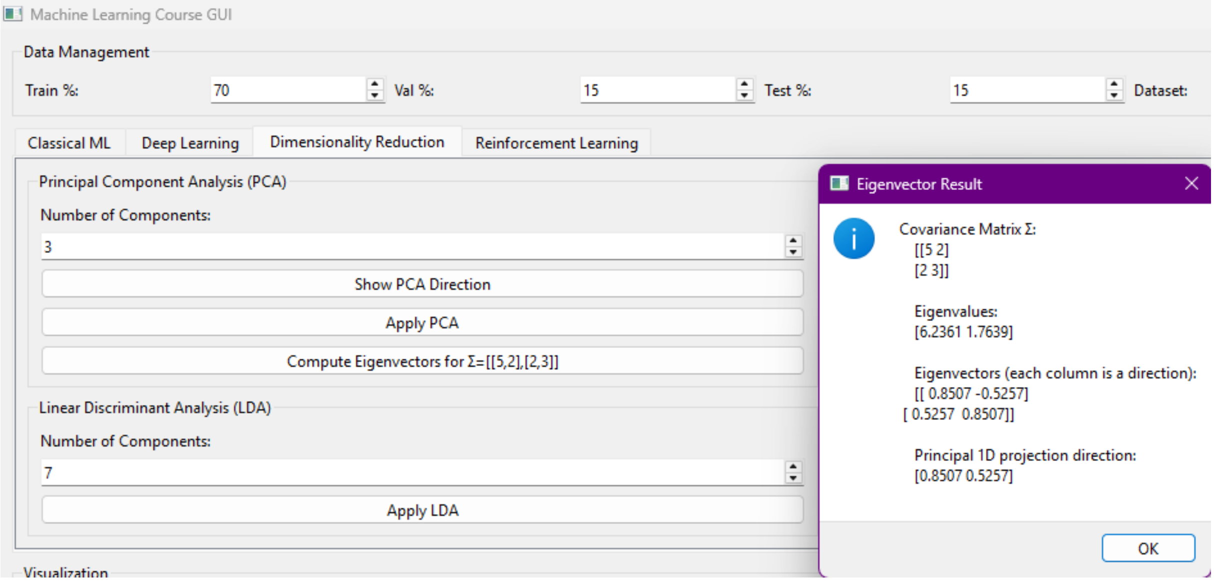
Principal Component Analysis (PCA) explained variance ratio plot illustrating the contribution of each principal component.



```
First Principal Component (Direction Vector):
[ 5.000e-03  2.000e-03  3.440e-02  5.161e-01  0.000e+00  0.000e+00
  1.000e-04  0.000e+00  0.000e+00 -0.000e+00  3.000e-04 -1.000e-04
  2.300e-03  5.890e-02 -0.000e+00  0.000e+00  0.000e+00  0.000e+00
 -0.000e+00 -0.000e+00  7.000e-03  2.800e-03  4.850e-02  8.524e-01
  0.000e+00  1.000e-04  2.000e-04  1.000e-04  0.000e+00  0.000e+00]
```

Şekil 3 : First Principal Component Visualization

Visualization of the first principal component vector overlaid on the PCA-reduced feature space.



Train %: 70
Val %: 15
Test %: 15

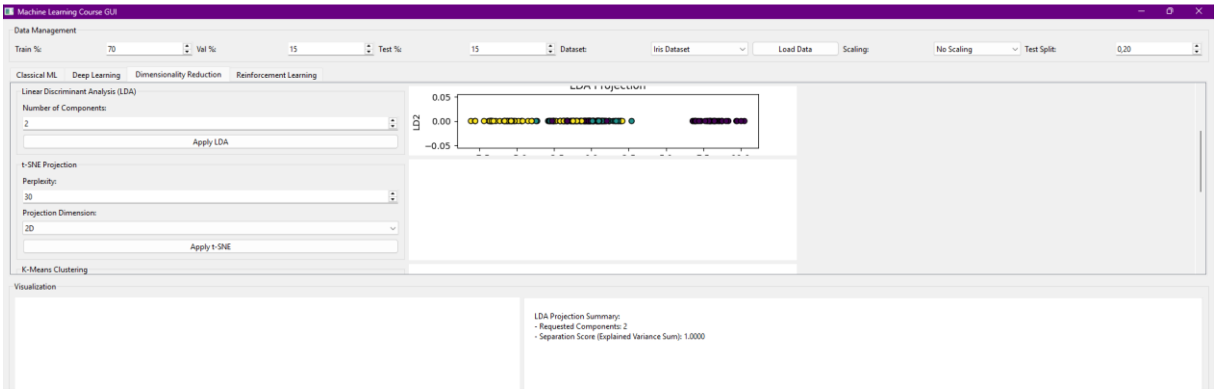
Number of Components: 3

Number of Components: 7

```
Covariance Matrix Σ:
  [[5 2]
  [2 3]]

  Eigenvalues:
  [6.2361 1.7639]

  Eigenvectors (each column is a direction):
  [[ 0.8507 -0.5257]
 [ 0.5257  0.8507]]

  Principal 1D projection direction:
  [0.8507 0.5257]
```

Şekil 4 Eigenvalues and Eigenvectors from Covariance Matrix Σ

Eigenvalues and eigenvectors computed from the covariance matrix Σ, demonstrating the principal direction for 1D projection.



Machine Learning Course GUI

Data Management

Train %: 70 Val %: 15 Test %: 15 Dataset: Iris Dataset Scaling: No Scaling Test Split: 0,20

Linear Discriminant Analysis (LDA)

Number of Components: 2

t-SNE Projection

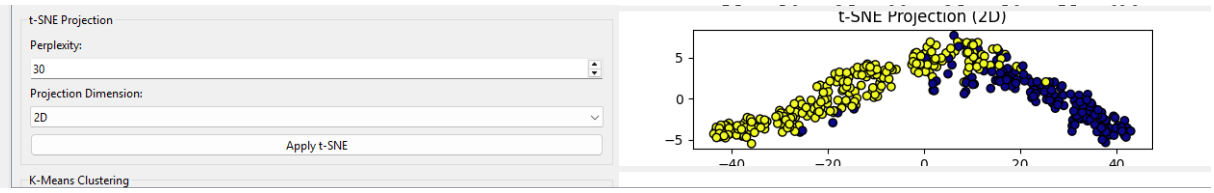
Perplexity: 30

Projection Dimension: 2D

LDA Projection Summary:
- Requested Components: 2
- Separation Score (Explained Variance Sum): 1.0000

Şekil 5 Linear Discriminant Analysis (LDA) Projection

The scatter plot illustrates the projection of the dataset onto the first two linear discriminants using Linear Discriminant Analysis (LDA). The plot demonstrates the separation of different classes by maximizing between-class variance while minimizing within-class variance.



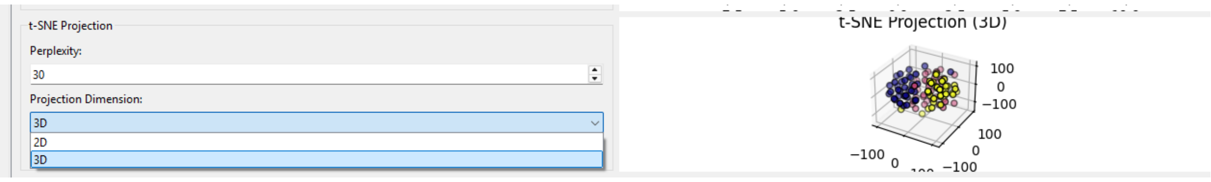
t-SNE Projection

Perplexity: 30

Projection Dimension: 2D

Şekil 6 t-SNE Projection of the Breast Cancer Dataset

The scatter plot shows the 2D projection of the dataset obtained through t-Distributed Stochastic Neighbor Embedding (t-SNE). The technique preserves local structures of the data and is effective for visualizing clusters in high-dimensional spaces



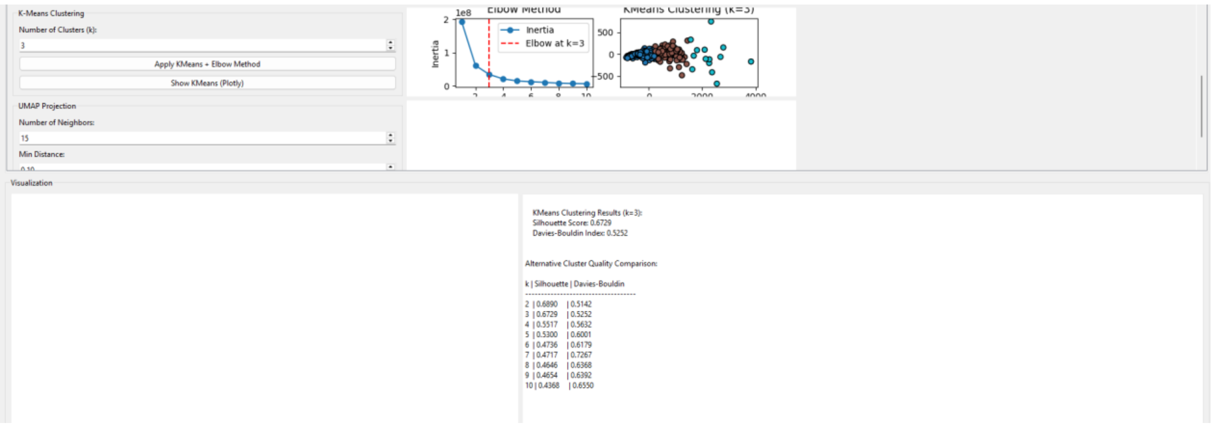
t-SNE Projection

Perplexity: 30

Projection Dimension: 3D

Şekil 7B t-SNE Projection (3D)

The 3D scatter plot illustrates the projection of the dataset using t-Distributed Stochastic Neighbor Embedding (t-SNE) in three dimensions. This visualization enables a more detailed exploration of cluster structures in high-dimensional spaces compared to the 2D projection..



Number of Clusters (k): 3

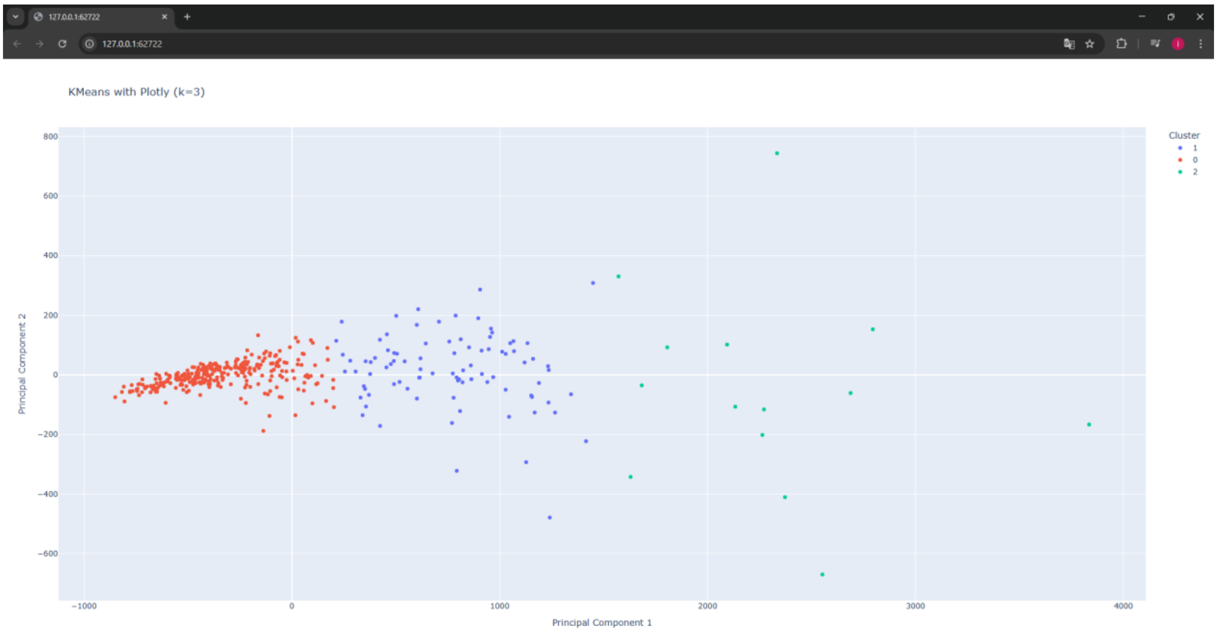
Number of Neighbors: 15

Min Distance: 0.10

```
KMeans Clustering Results (k=3):
Silhouette Score: 0.6729
Davies-Bouldin Index: 0.5252


Alternative Cluster Quality Comparison:

k | Silhouette | Davies-Bouldin
-----------------------------------
2  | 0.6890     | 0.5142
3  | 0.6729     | 0.5252
4  | 0.5517     | 0.5632
5  | 0.5300     | 0.6001
6  | 0.4736     | 0.6179
7  | 0.4717     | 0.7267
8  | 0.4646     | 0.6368
9  | 0.4654     | 0.6392
10 | 0.4368     | 0.6550
```

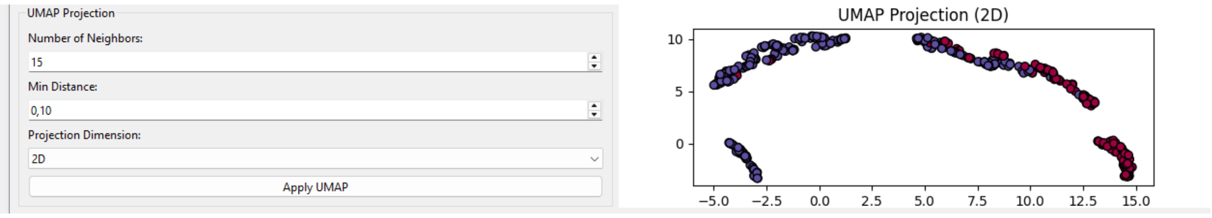
Şekil 8 K-Means Clustering Results and Elbow Method

The left plot demonstrates the Elbow Method used to determine the optimal number of clusters based on inertia values, identifying an elbow point at k=3. The right plot visualizes the clustering results for k=3 after dimensionality reduction, with each color representing a different cluster.



Şekil 9-Means Clustering Visualized with Plotly (k=3)

The clustering results are visualized interactively using Plotly for k=3 clusters. The clusters are distinguished in the two-dimensional PCA-transformed space, with enhanced visual clarity and interactivity compared to static plots.


UMAP Projection

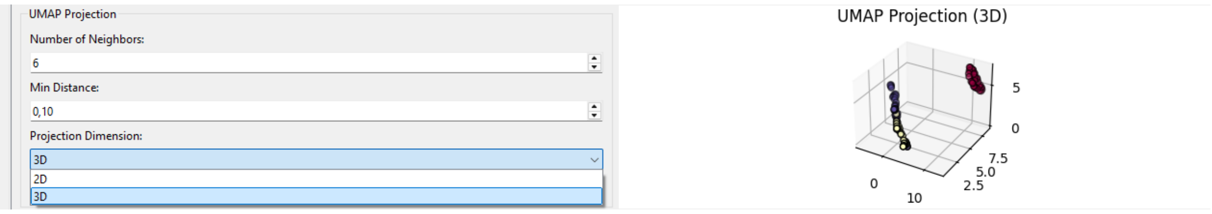
Number of Neighbors: 15

Min Distance: 0,10

Projection Dimension: 2D

Şekil 10 UMAP Projection of the Breast Cancer Dataset

The scatter plot visualizes the dataset in a two-dimensional space using Uniform Manifold Approximation and Projection (UMAP). UMAP preserves both local and global data structures, offering a clear view of the clustering structure in reduced dimensions.


UMAP Projection

Number of Neighbors: 6

Min Distance: 0,10

Projection Dimension: 3D

Şekil 11B UMAP Projection (3D)

The 3D scatter plot displays the Uniform Manifold Approximation and Projection (UMAP) results in three dimensions. Compared to the 2D projection, this 3D visualization provides a more comprehensive view of the dataset's underlying structure, allowing better observation of cluster separations in high-dimensional spaces.

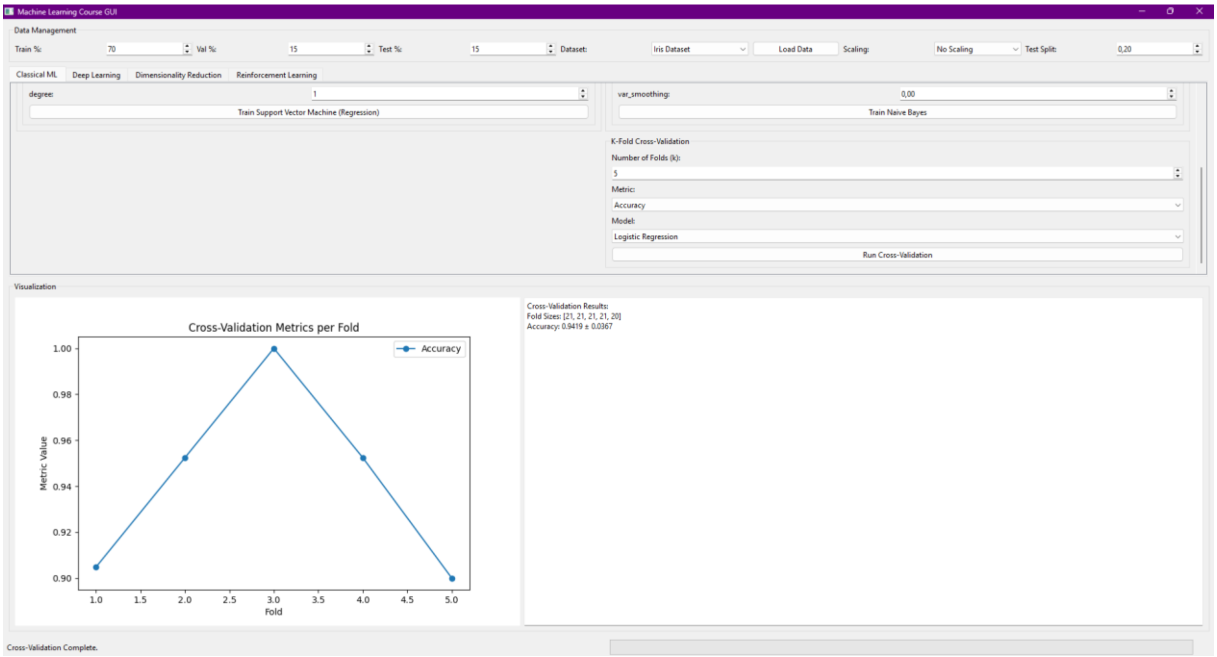
UMAP as a Faster Alternative to t-SNE:

Uniform Manifold Approximation and Projection (UMAP) was integrated into the GUI as a faster alternative to t-Distributed Stochastic Neighbor Embedding (t-SNE).

While t-SNE is effective for preserving local structures during dimensionality reduction, it is computationally intensive and slower for large datasets.

UMAP addresses this limitation by offering a significantly faster computation time, better scalability to larger datasets, and the ability to preserve both local and global structures of the data.

In the implemented GUI, UMAP allows users to adjust the number of neighbors, minimum distance, and projection dimensions (2D/3D), providing an efficient and flexible tool for visualizing high-dimensional datasets.



Train %: 70
Val %: 15
Test %: 15
Dataset: Iris Dataset
Scaling: No Scaling
Test Split: 0,20

degree: 1

var_smoothing: 0,00

Number of Folds (k): 5

Metric: Accuracy

Model: Logistic Regression

```
Cross-Validation Results:
Fold Sizes: [21, 21, 21, 21, 20]
Accuracy: 0.9419 ± 0.0367
```

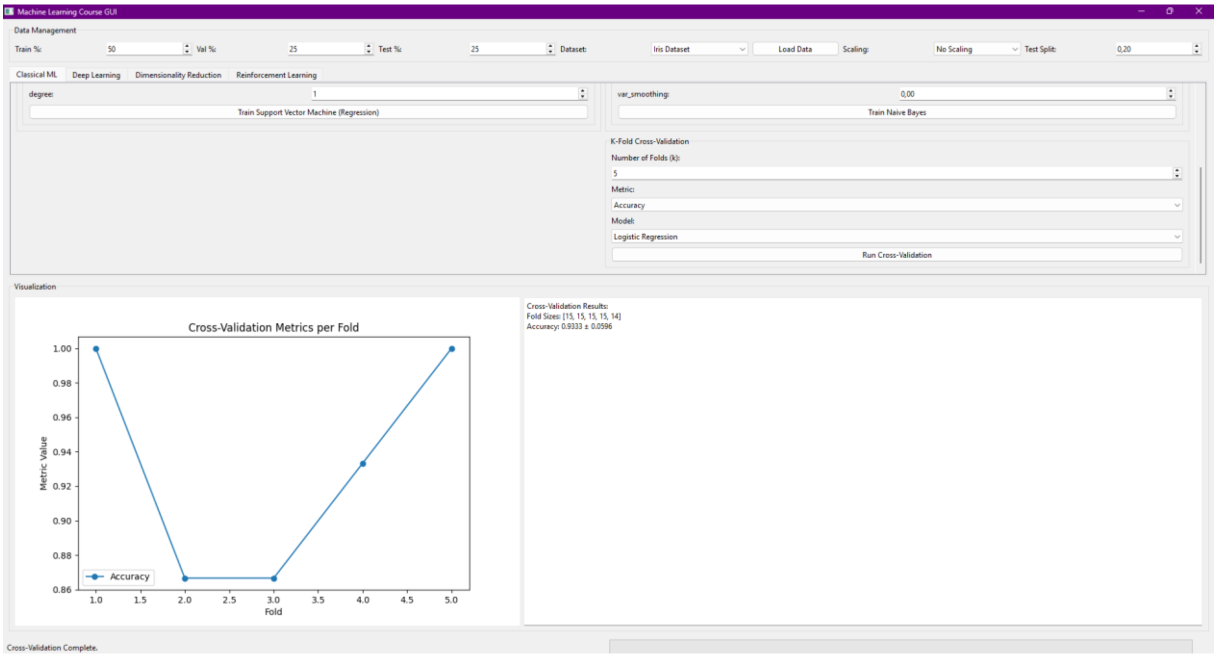
Cross-Validation Complete.

Şekil 12 Initial Train/Val/Test Split Settings (70-15-15 Example)

The initial setting shows a 70% training, 15% validation, and 15% testing data split configuration, demonstrating user control over dataset partitioning.

Note on Fold Sizes:

In the provided GUI, 5-fold cross-validation was performed.
Ideally, for 100 samples, each fold would contain exactly 20 samples ([20, 20, 20, 20, 20]).
However, the dataset used during the demonstration (Breast Cancer Dataset) does not have exactly 100 samples after Train/Validation/Test splitting and scaling operations.
Therefore, when k=5 is selected, the available sample size is slightly more than 100 or slightly less, causing folds to be distributed approximately equally, resulting in fold sizes like [21, 21, 21, 21, 20].
Despite this small variation, the k-fold validation logic is properly maintained, ensuring fair and unbiased model evaluation across folds.



Train %: 50 Val %: 25 Test %: 25 Dataset: Iris Dataset Scaling: No Scaling Test Split: 0,20

degree: 1

var_smoothing: 0,00

Number of Folds (k): 5

Metric: Accuracy

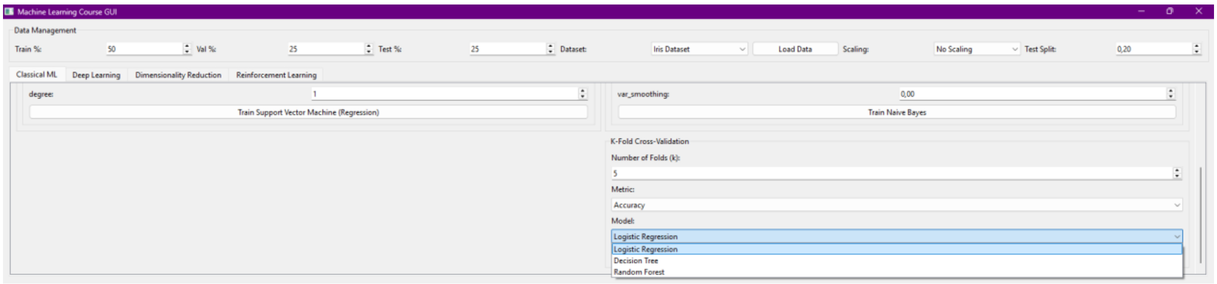
Model: Logistic Regression

```
Cross-Validation Results:
Fold Sizes: [15, 15, 15, 15, 14]
Accuracy: 0.9333 ± 0.0596
```

Cross-Validation Complete.

Şekil 13 Dynamic Adjustment of Train/Val/Test Split

The GUI allows flexible adjustment of data splitting ratios. Users can customize the percentages for training, validation, and testing according to their specific needs.



Train %: 50 Val %: 25 Test %: 25 Dataset: Iris Dataset Scaling: No Scaling Test Split: 0,20

degree: 1

var_smoothing: 0,00

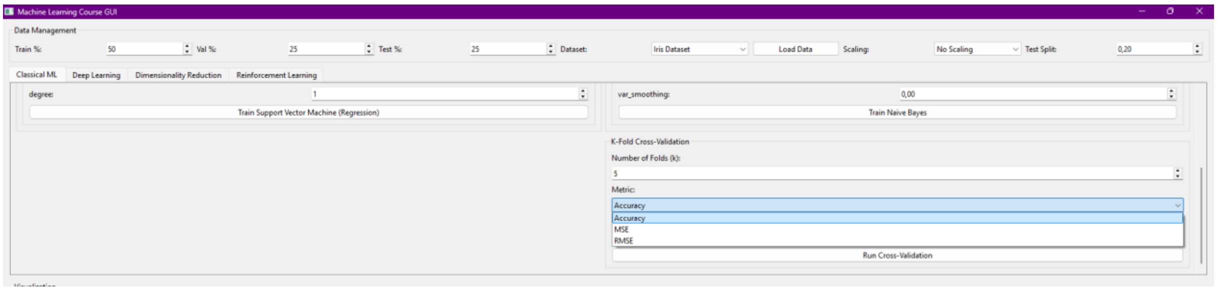
Number of Folds (k): 5

Metric: Accuracy

Model: Logistic Regression

Şekil 14 K-Fold Cross-Validation - Model Selection

The GUI allows the user to select the model (e.g., Logistic Regression, Decision Tree, Random Forest) for k-fold cross-validation, enabling flexible evaluation of various machine learning algorithms.



Train %: 50 Val %: 25 Test %: 25 Dataset: Iris Dataset Scaling: No Scaling Test Split: 0,20

degree: 1

var_smoothing: 0,00

Number of Folds (k): 5

Metric: Accuracy

Şekil 15 K-Fold Cross-Validation - Metric Selection

The user can also choose the evaluation metric (Accuracy, MSE, or RMSE) to be reported during k-fold cross-validation, providing comprehensive performance insights.

Additional Note on K-Fold Cross-Validation

In the K-Fold Cross-Validation section of the developed GUI, both the model and evaluation metric can be selected by the user from dropdown menus.
The model selection includes options such as Logistic Regression, Decision Tree, and Random Forest, while the metric selection includes Accuracy, MSE, and RMSE.
After configuring these selections, the system successfully performs cross-validation based on the chosen settings and outputs the results.
Specifically, it displays the fold sizes, calculates the mean and standard deviation of the selected metric across folds, and visualizes the metric's behavior for each fold with a plotted graph.
This flexibility demonstrates that the interface can both dynamically adjust the model/metric settings and produce meaningful validation results.

Comparison of PCA and t-SNE Based on Outputs

Comparison of PCA and t-SNE Based on Outputs

In the generated outputs, both Principal Component Analysis (PCA) and t-Distributed Stochastic Neighbor Embedding (t-SNE) techniques were applied to the dataset, and their results were visualized.

- **PCA Output (Şekil 2 and Şekil 3):**
  The explained variance ratio plot (Figure 2) indicates that the first principal component captures a very large portion of the total variance, while subsequent components contribute minimally.
  Additionally, in the PCA scatter plot (Figure 3), the data distribution tends to align along linear directions, suggesting that PCA preserves global structure but may struggle to separate closely related clusters effectively in complex datasets.

- **t-SNE Output (Şekil 6):**
  In contrast, the t-SNE projection (Figure 6) reveals a more intuitive separation of clusters in a two-dimensional space.
  t-SNE prioritizes maintaining local neighborhood relationships, which is evident from the tighter, clearly separated groupings of data points compared to PCA.

Thus, **PCA** is effective for **linear structures** and **global variance analysis**, while **t-SNE** excels in **visualizing local structures** and **complex manifolds**, which aligns with their intended theoretical purposes.