Experiment: Principal Component Analysis (PCA) vs Linear Discriminant Analysis (LDA)

Title:

Implementation of Linear Discriminant Analysis and Principal Component Analysis

Aim:

Comparing the results of PCA with LDA for better suitability

Objective:

Students will learn:

- The implementation of the principal component analysis and Linear Discriminant analysis on a dataset.
- Visualization and interpretation of results.

Problem Statement

Implement a Linear Discriminant Analysis LDA algorithm on any dataset of your choice and compare the results with PCA for better suitability

---

Explanation / Stepwise Procedure / Algorithm

1. **Figures/Diagrams**

- LDA and PCA 2D plots plotted for the dataset.
- Comparison between LDA and PCA.

LDA_Moodle[1]

February 10, 2025

0.1 Dimentionality Reduction (PCA, TSNE, LDA)

[30]:
```
import numpy as np
import pandas as pd
from sklearn.datasets import load_iris
from sklearn.decomposition import PCA
from sklearn.manifold import TSNE
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis as LDA
import matplotlib.pyplot as plt
import seaborn as sns
```

[31]:
```
# Load the Iris dataset
iris = load_iris()
X = iris.data
y = iris.target
target_names = iris.target_names
```

0.1.1 PCA

[32]:
```
# Apply PCA
pca = PCA(n_components=2)
X_pca = pca.fit_transform(X)
```

0.1.2 LDA

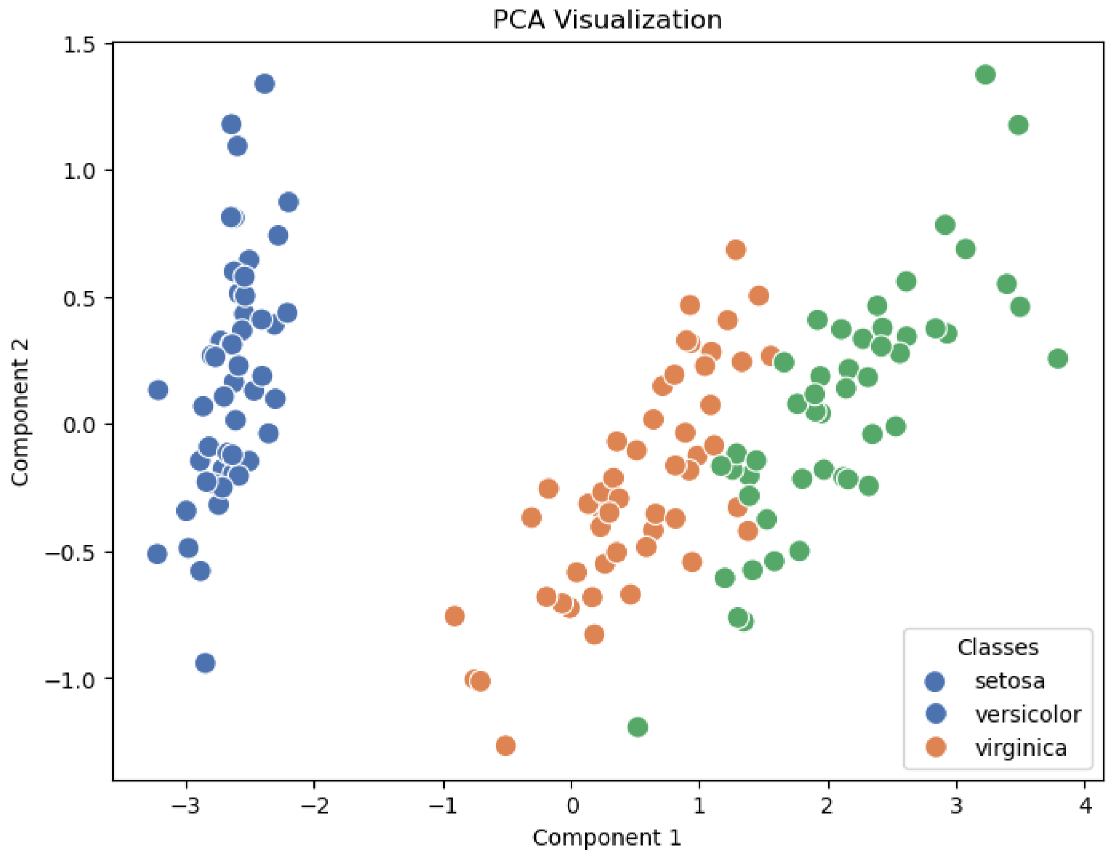
[33]:
```
# Apply LDA
lda = LDA(n_components=2)
X_lda = lda.fit_transform(X, y)
```

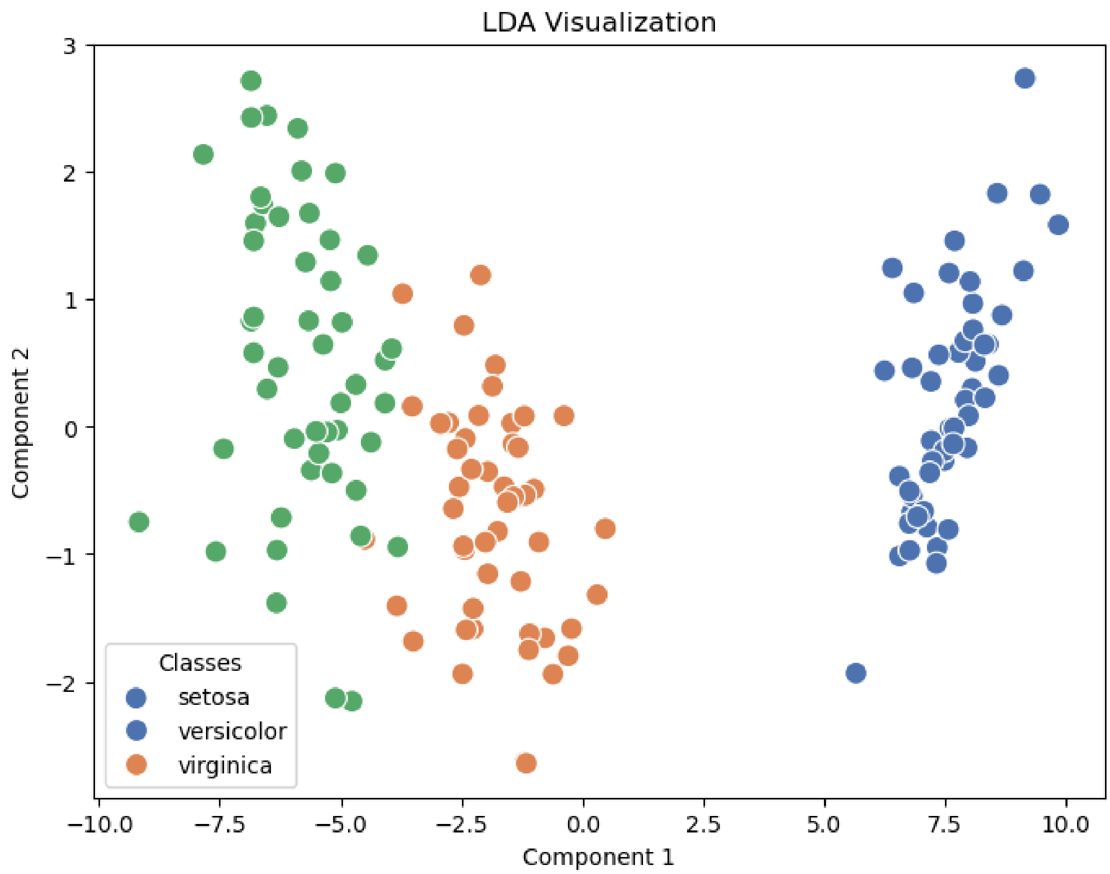
0.1.3 Visualisation

[34]:
```
# Plotting function
def plot_embedding(X_embedded, title, labels):
    plt.figure(figsize=(8, 6))
    sns.scatterplot(x=X_embedded[:, 0], y=X_embedded[:, 1], hue=labels,␣
 ↪palette="deep", s=100)
    plt.title(title)
    plt.xlabel("Component 1")
```

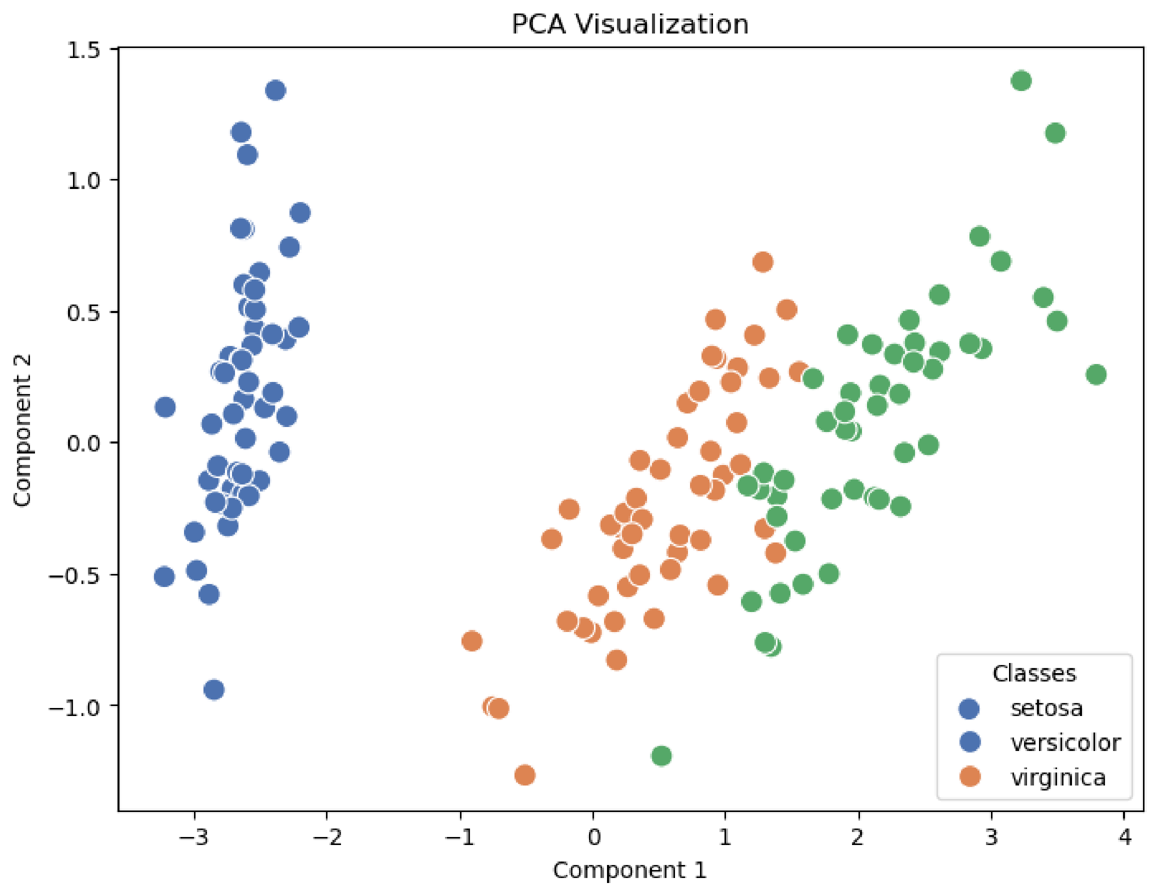
```python
    plt.ylabel("Component 2")
    plt.legend(title="Classes", loc="best", labels=target_names)
    plt.show()
```

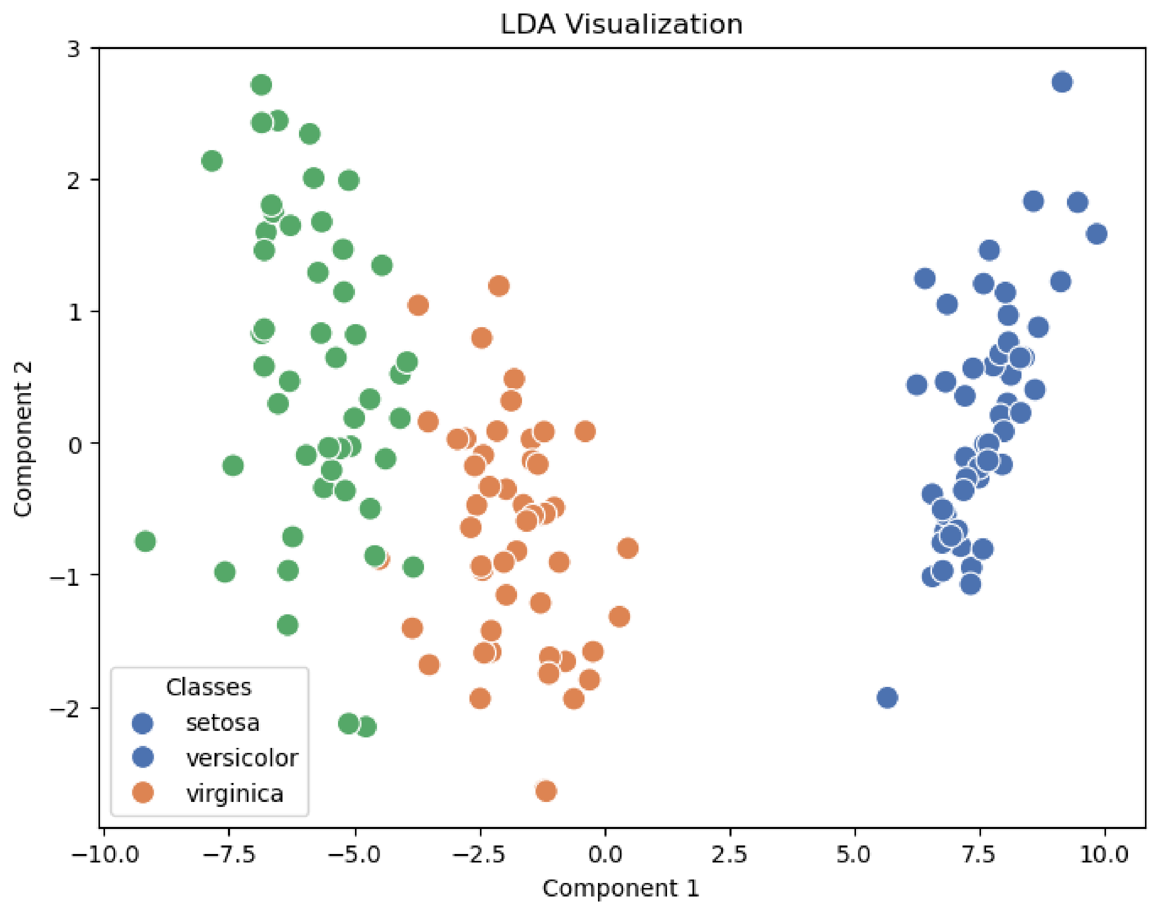
[35]:
```python
# Visualize PCA results
plot_embedding(X_pca, "PCA Visualization", y)

# Visualize LDA results
plot_embedding(X_lda, "LDA Visualization", y)
```





LDA is better than PCA when the goal is to maximize class separability, as it uses class label information to find the feature space that best distinguishes between classes. While PCA focuses on capturing the maximum variance in the data without considering class distinctions, LDA specifically reduces dimensionality by preserving the most discriminative features. This makes LDA more effective in supervised learning tasks like classification, where distinguishing between categories is crucial.


PCA Visualization
Component 2
Component 1
1.5
1.0
0.5
0.0
−0.5
−1.0
−3
−2
−1
0
1
2
3
4
Classes
setosa
versicolor
virginica


LDA Visualization
Component 2
Component 1
3
2
1
0
−1
−2
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0
Classes
setosa
versicolor
virginica

Challenges Encountered

1. LDA requires class labels, making it difficult to implement in unsupervised learning.
2. LDA limits the number of components to *classes - 1*, unlike the flexible PCA.
3. Eigen decomposition in both methods can lead to numerical instability or singular matrix errors.

Conclusion

- PCA is better suited for unsupervised learning as it doesn't rely on class labels.
- LDA outperforms PCA when class separability is the primary objective.
- PCA offers more flexibility in component selection compared to LDA.
- Both PCA and LDA can face computational challenges with small or poorly conditioned datasets.