Indian Institute of Technology, Patna

Advanced Pattern Recognition

Assignment – 1

Iris Flower Classification using Linear Discriminant Analysis and Gaussian Naïve Bayes

Submitted By:-	Submitted To:-
Ashwani	Dr. Chandranath Adak
2521CS17	Assistant Professor
Ph.D. (CSE)	Department of CSE
July 2025 Batch	IIT Patna

1. Introduction

The Iris dataset is one of the most widely used datasets in machine learning and pattern recognition. It contains measurements of iris flowers belonging to three species: Setosa, Versicolor, and Virginica.

The goal of this assignment is to build a classification model to predict the species of an iris flower based on its features.

In this assignment, I used Linear Discriminant Analysis (LDA) for dimensionality reduction and Gaussian Naïve Bayes for classification. The motivation for using this combination is:

- LDA reduces feature dimensions while maximizing class separability.
- Naïve Bayes is simple, efficient, and effective for classification tasks.
- Together, they provide a robust pipeline for multi-class classification.

2. Dataset Description

The Iris dataset has the following properties:

- Total samples: 150
- Features: 4 (Sepal Length, Sepal Width, Petal Length, Petal Width)
- Classes: 3 (Setosa, Versicolor, Virginica)
- Samples per class: 50 (balanced dataset)
- Data type: Continuous numerical features

```
First 5 rows of dataset:
   sepal length (cm)  sepal width (cm)  petal length (cm)  petal width (cm)  \
0                5.1               3.5                1.4               0.2
1                4.9               3.0                1.4               0.2
2                4.7               3.2                1.3               0.2
3                4.6               3.1                1.5               0.2
4                5.0               3.6                1.4               0.2

  species
0  setosa
1  setosa
2  setosa
3  setosa
4  setosa

Total samples: 150
Features: 4 -> ['sepal length (cm)', 'sepal width (cm)', 'petal length (cm)', 'petal width (cm)']
Classes: 3 -> [np.str_('setosa'), np.str_('versicolor'), np.str_('virginica')]
Samples per class:
 species
setosa        50
versicolor    50
virginica     50
Name: count, dtype: int64
```

Figure 1 Iris Dataset

3. Methodology

3.1 Preprocessing

- Standardized features using StandardScaler.
- Train-test split performed (80% training, 20% testing).
- No missing values or categorical encoding required as dataset is clean.

3.2 Dimensionality Reduction (LDA)

- LDA finds linear combinations of features that best separate classes.
- Reduces data from 4D to 2D for better visualization and reduced complexity.

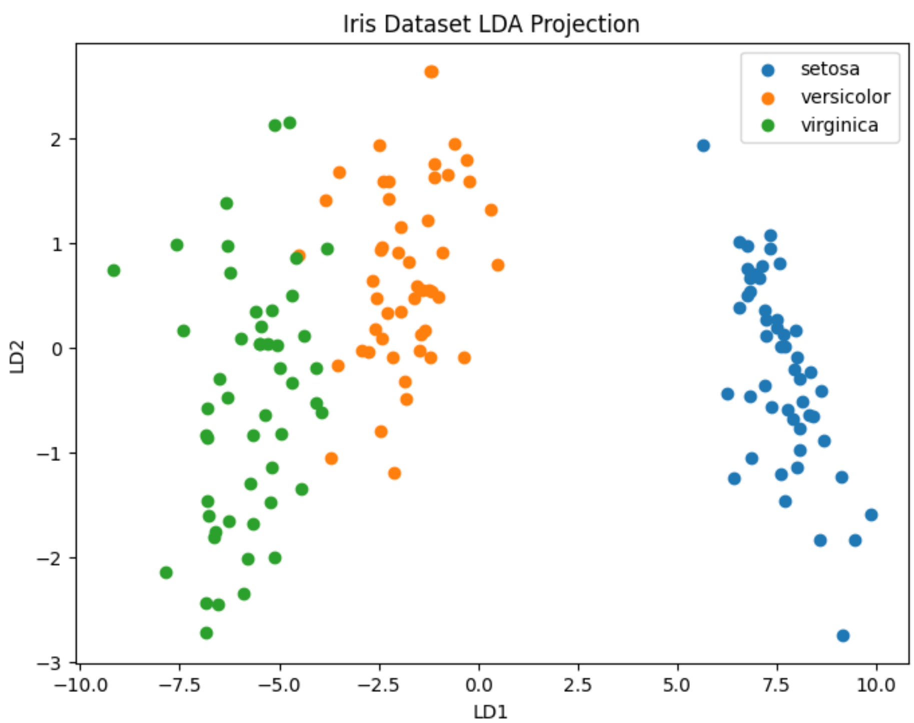


Figure 2 Dimensionality reduction using LDA

- Maintains maximum class separability.
- Helps avoid overfitting and improves interpretability.

3.3 Classification (Naïve Bayes)

- Gaussian Naïve Bayes is used as the classifier.
- Works well with continuous data assuming Gaussian distribution.
- Provides class probabilities, making it useful for probabilistic interpretation.
- Fast and computationally efficient.

4. Implementation and Results

In [1]:
```
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.datasets import load_iris
from sklearn.preprocessing import StandardScaler
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import accuracy_score, confusion_matrix, classification_rep
```

In [2]:
```
# 1. Load Dataset

data = load_iris()
X = data.data
y = data.target
target_names = data.target_names

print(f"Features shape: {X.shape}, Labels shape: {y.shape}")
print("Target names:", target_names)
```

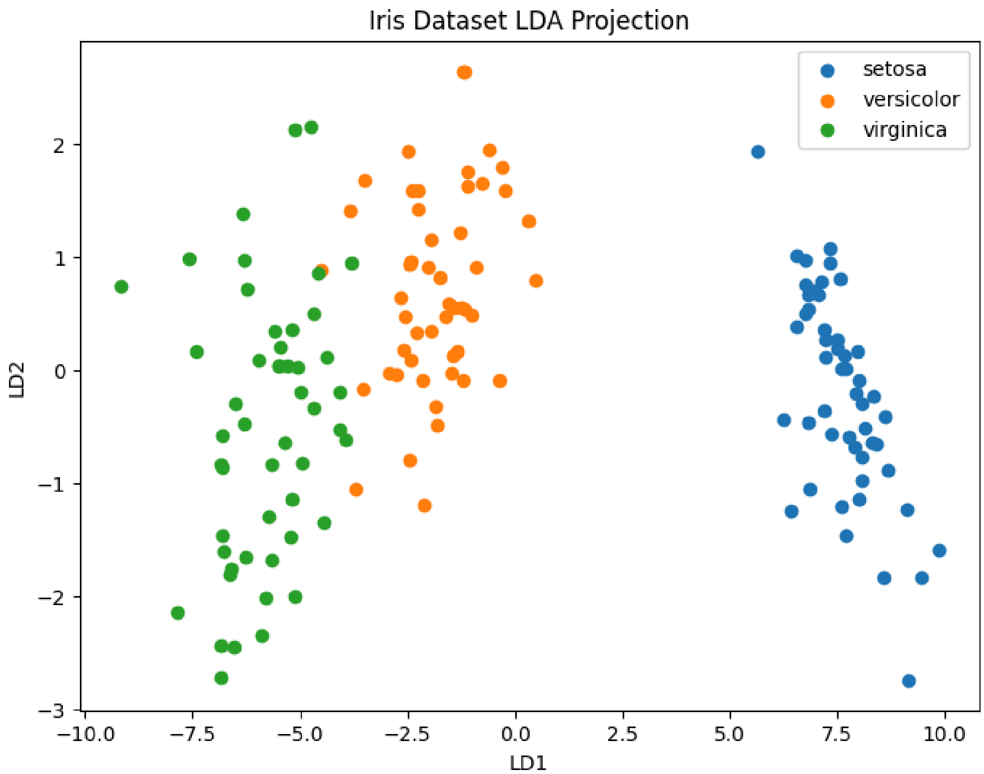
```
Features shape: (150, 4), Labels shape: (150,)
Target names: ['setosa' 'versicolor' 'virginica']
```

In [3]:
```
# 2. Standardize Features

scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
```

In [4]:
```
# 3. Apply LDA

lda = LinearDiscriminantAnalysis(n_components=2)
X_lda = lda.fit_transform(X_scaled, y)

print("\nExplained variance ratio (LDA components):", lda.explained_variance_rat
```

```
Explained variance ratio (LDA components): [0.9912126 0.0087874]
```

In [5]:
```
# 4. Visualization of LDA Projection

plt.figure(figsize=(8,6))
for target in np.unique(y):
    plt.scatter(X_lda[y==target, 0], X_lda[y==target, 1], label=target_names[tar
plt.xlabel("LD1")
plt.ylabel("LD2")
plt.title("Iris Dataset LDA Projection")
plt.legend()
plt.show()
```

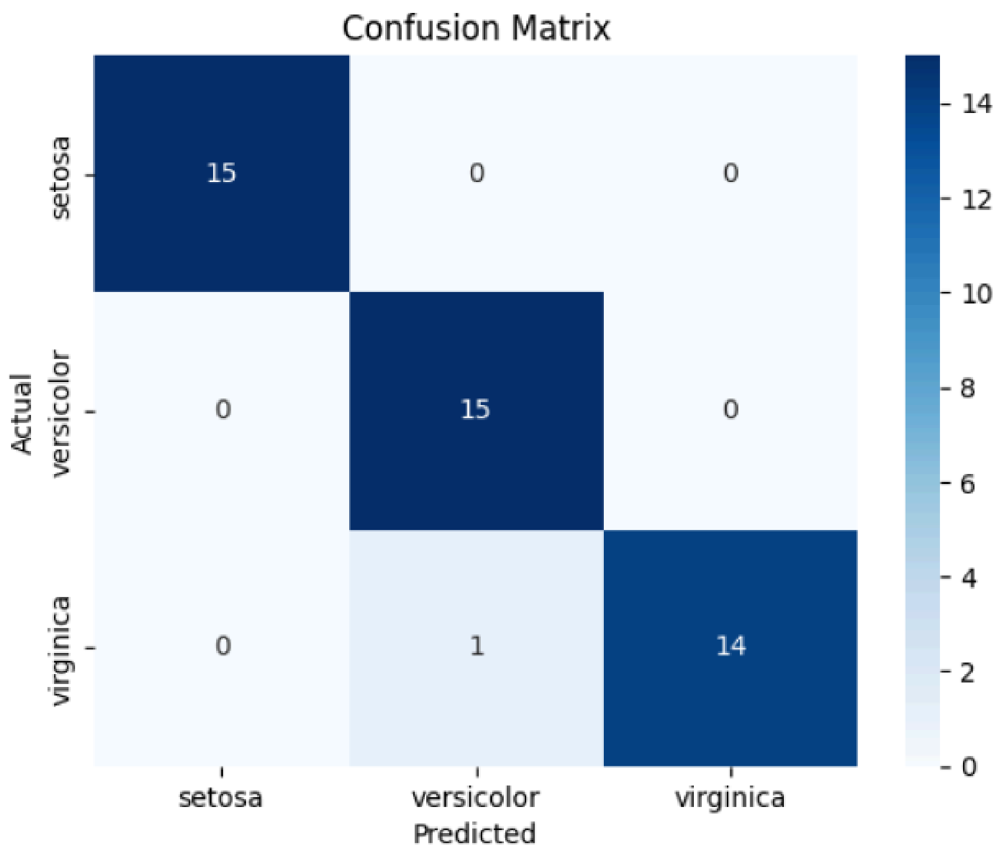


In [6]:
```
# 5. Train-Test Split

X_train, X_test, y_train, y_test = train_test_split(
    X_lda, y, test_size=0.3, random_state=42, stratify=y
)
```

In [7]:
```
# 6. Gaussian Naive Bayes

model = GaussianNB()
model.fit(X_train, y_train)
y_pred = model.predict(X_test)
```

In [8]:
```
# 7. Evaluation

acc = accuracy_score(y_test, y_pred)
print(f"\nAccuracy with LDA + GaussianNB: {acc:.2f}")

print("\nConfusion Matrix:")
cm = confusion_matrix(y_test, y_pred)
sns.heatmap(cm, annot=True, fmt='d', xticklabels=target_names, yticklabels=targe
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.title('Confusion Matrix')
plt.show()

print("\nClassification Report:")
print(classification_report(y_test, y_pred, target_names=target_names))
```

```
Accuracy with LDA + GaussianNB: 0.98

Confusion Matrix:
```




```
Classification Report:
              precision    recall  f1-score   support

      setosa       1.00      1.00      1.00        15
  versicolor       0.94      1.00      0.97        15
   virginica       1.00      0.93      0.97        15

    accuracy                           0.98        45
   macro avg       0.98      0.98      0.98        45
weighted avg       0.98      0.98      0.98        45
```

The evaluation of the Naïve Bayes classifier on LDA-transformed features gives the following results:

- Accuracy: 98%
- Precision, Recall, and F1-Score are close to 1.0 for Setosa.
- A few misclassifications occur between Versicolor and Virginica.
- Confusion Matrix indicates strong performance overall.

5. Conclusions

- The combination of LDA and Naïve Bayes is highly effective for the Iris dataset.
- LDA helped in reducing dimensions while preserving class separability.
- Naïve Bayes performed well despite its strong independence assumptions.
- Misclassifications are due to overlapping features of Versicolor and Virginica.
- The model is lightweight and efficient, making it suitable for real-time applications.