Introduction

This project seeks to evaluate the two popular classification methods – Linear Discriminant Analysis (LDA) and Quadratic Discriminant Analysis (QDA) on the performance aspect using a synthetic dataset. Both methods have their primary application in supervised classification in which data is classified into different groups depending on the characteristics of the data. LDA is restrictive in that it classifies the data into classes which are separated using linear boundaries, while QDA is not since it uses quadratic decision boundaries. This allows us to effectively determine the extent of variation in efficiency and effectiveness of these two classification techniques.

Objective

- Compare the accuracy and performance of LDA and QDA on a synthetic multi-class dataset.
- Visualize the decision boundaries of both models to understand how they separate the data.
- Assess the suitability of each model based on the complexity of the data (linear vs non-linear separability).

Data Management and Preparation:

The dataset applied in this project is new and is similar to a real-life multiclass classification task. The data comprises:

1000 samples, 2 features per sample, 3 classes, each with a separate cluster of points.

The data had be generated this data asdskjandfjad by using the make_classification function from sklearn.datasets. This make_classification dataset is pretty balanced so it serves well to evaluate linear and non-linear classification techniques.

Data Splitting:

Before training the models, the dataset was split into training and test portions. The ratio used was 80:20. This helps so as to train their models on one set of

data and test them on another which has not been seen before to prevent overfitting.

Model Training:

Linear Discriminant Analysis (LDA):

The LDA method assumes that the data classes can be separated with lines. The objective is to obtain minimum with in-class scatter and maximum between-class scatter. Hence LDA is used when the classes are separable by a straight line.

```
# Initialize and train the LDA model
lda = LinearDiscriminantAnalysis()
lda.fit(X_train, y_train)
y_pred_lda = lda.predict(X_test)

print("LDA Accuracy:", accuracy_score(y_test, y_pred_lda))
print("Confusion Matrix:\n", confusion_matrix(y_test, y_pred_lda))
print("Classification Report:\n", classification_report(y_test, y_pred_lda))
```

Quadratic Discriminant Analysis (QDA):

QDA makes use of quadratic boundaries allowing a more relaxed model structure. This is perfect for two dimensional data that is not linearly separable because it will model more abstract relations between the classes than LDA does.

Performance Evaluation:

The performance of both models was evaluated based on standard metrics of classification:

Visualizing Decision Boundaries:

To further illustrate the performance difference between LDA and QDA, the decision boundaries of both models were plotted. This visualization shows how each model divides the feature space into different regions representing the three classes.

```
print("LDA Accuracy:", accuracy_score(y_test, y_pred_lda))
print("Confusion Matrix:\n", confusion_matrix(y_test, y_pred_lda))
print("Classification Report:\n", classification_report(y_test, y_pred_lda))
```

```
LDA Accuracy: 0.8266666666666667
Confusion Matrix:
 [[ 75   4  22]
 [ 16  71   0]
 [  0  10 102]]
Classification Report:
              precision    recall  f1-score   support

           0       0.82      0.74      0.78       101
           1       0.84      0.82      0.83        87
           2       0.82      0.91      0.86       112

    accuracy                           0.83       300
   macro avg       0.83      0.82      0.82       300
weighted avg       0.83      0.83      0.83       300
```

LDA Decision Boundary:

Since LDA assumes linear separability, the decision boundaries are straight lines that divide the feature space. While this works well for simple, linearly separable data, it can fail when the data has more complex patterns.

QDA Decision Boundary:

QDA, on the other hand, allows for more flexible, curved decision boundaries. This is especially useful when the data cannot be separated by straight lines, making it a more powerful model for complex classification tasks.