Lab - Experiment 2

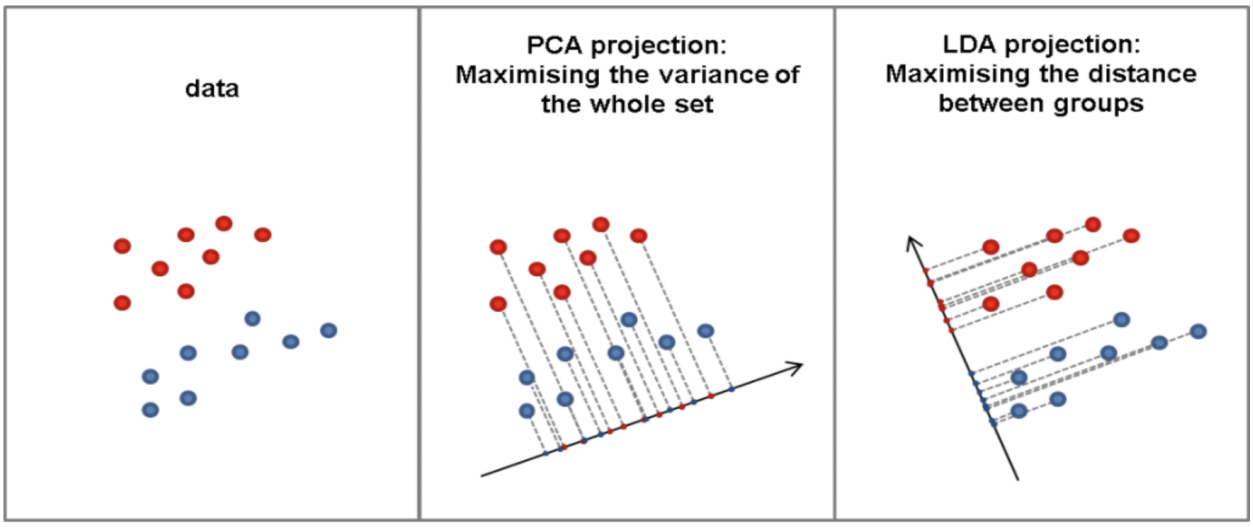
Linear Discriminant Analysis

It is a dimensionality reduction technique which is used to reduce dimension by preserving the important information.

Two criteria are used by LDA to create a new axis:

Maximize the distance between the means of the two classes.

Minimize the variation within each class.



PERFORMING ON IRIS DATASET

```
The Iris dataset represents 3 kinds of Iris flowers (Setosa,
Versicolour and Virginica) with 4 attributes: sepal length, s
epal width, petal length and petal width.

Linear Discriminant Analysis (LDA) tries to identify attribut
es that account for the most variance between classes or maxi
```

```
mize the separability between two classes. In particular, LD
A, in contrast to PCA, is a supervised method, using known cl
ass labels.
```

Applying Logistic Regression without LDA on IRIS DATASET

```
x_train , x_test , y_train , y_test = train_test_split(X,y,te
st_size = 0.3 , random_state = 42)

model = LogisticRegression()

model.fit(x_train,y_train)

y_pred_lr = model.predict(x_test)

accuracy = accuracy_score(y_test , y_pred_lr)
conf_matrix = confusion_matrix(y_test,y_pred_lr)
report = classification_report(y_test,y_pred_lr)

print(f'Accuracy: {accuracy:.2f}')
print('\nConfusion Matrix:')
print(conf_matrix)
print('\nClassification Report:')
print(report)
```

```
Accuracy: 1.00

Confusion Matrix:
[[19  0  0]
 [ 0 13  0]
 [ 0  0 13]]

Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        19
           1       1.00      1.00      1.00        13
           2       1.00      1.00      1.00        13

    accuracy                           1.00        45
   macro avg       1.00      1.00      1.00        45
weighted avg       1.00      1.00      1.00        45
```

Applying Linear Discriminant Analysis

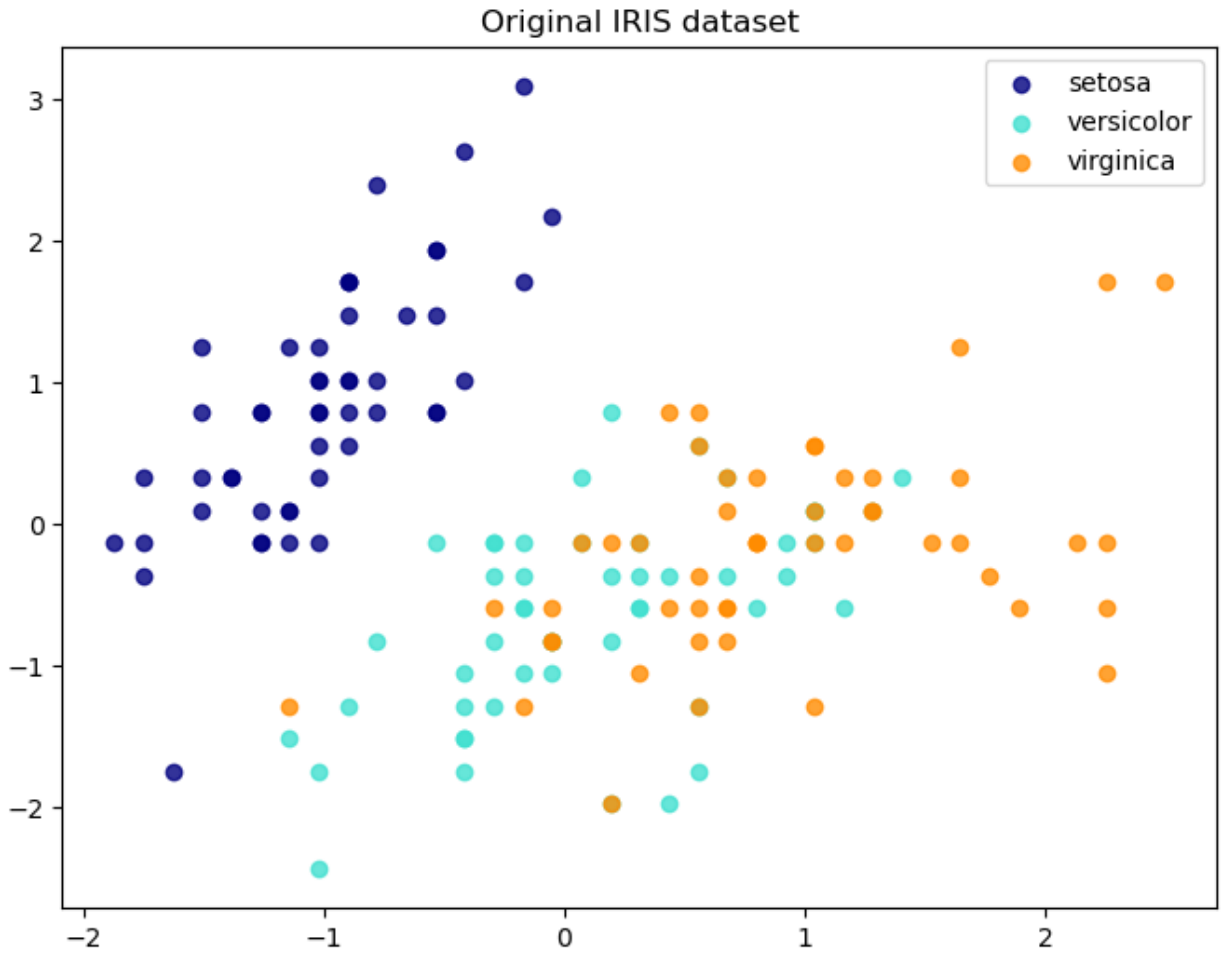
```
lda = LinearDiscriminantAnalysis()

lda = lda.fit(X,y).transform(X)
```

In LDA, the number of components is determined by the number of unique classes in the target variable. Specifically, for a classification problem with k unique classes, LDA will genera te at most k - 1 discriminant components. These components ar e derived to maximize the separation between the classes in t he transformed feature space.

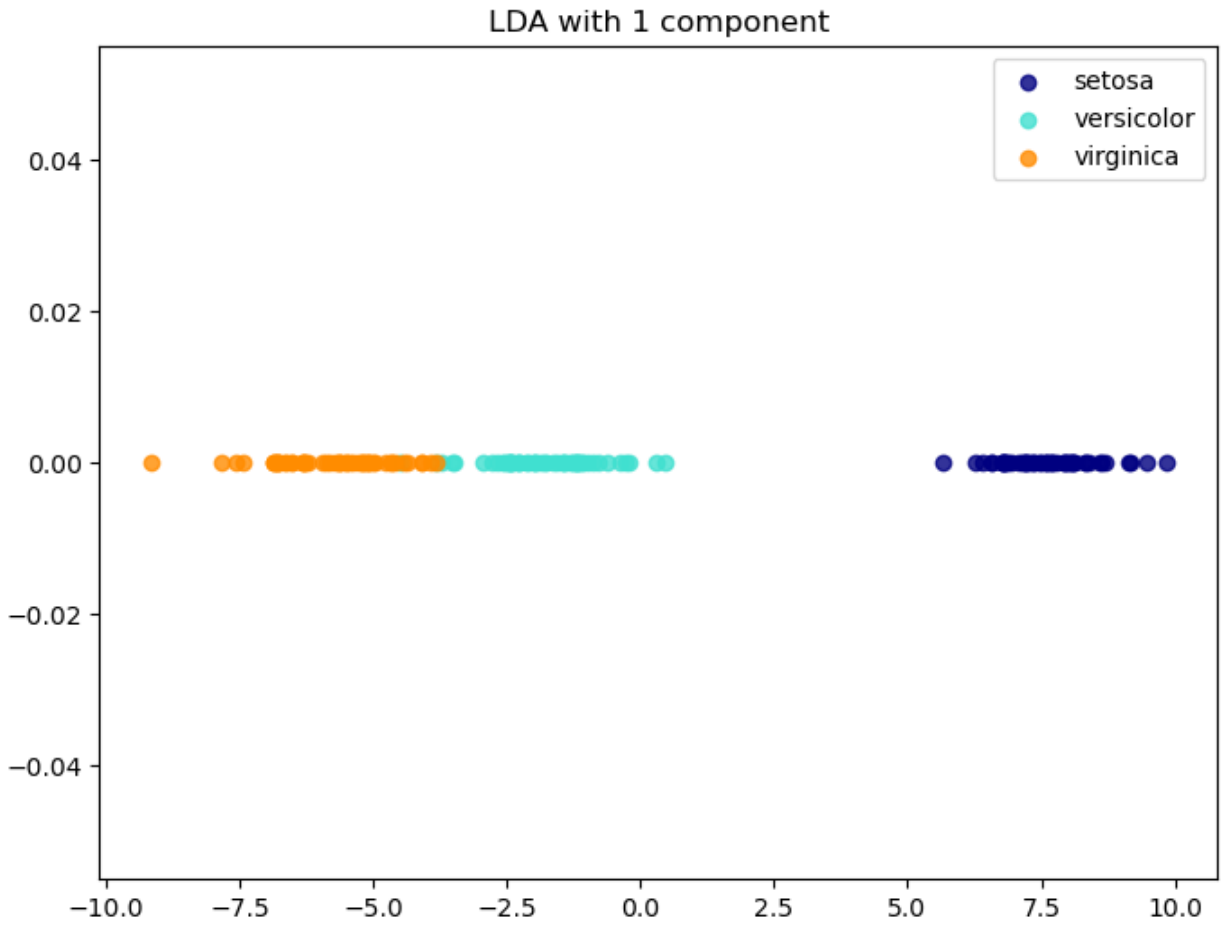
After fitting the LDA model, the transform method is called o n the original data X. This method transforms the data into a lower-dimensional space based on the learned discriminant com

```
ponents. The transformed data X_r2 will have fewer dimensions
than the original data, with the number of dimensions determi
ned by the number of unique classes in y.
```

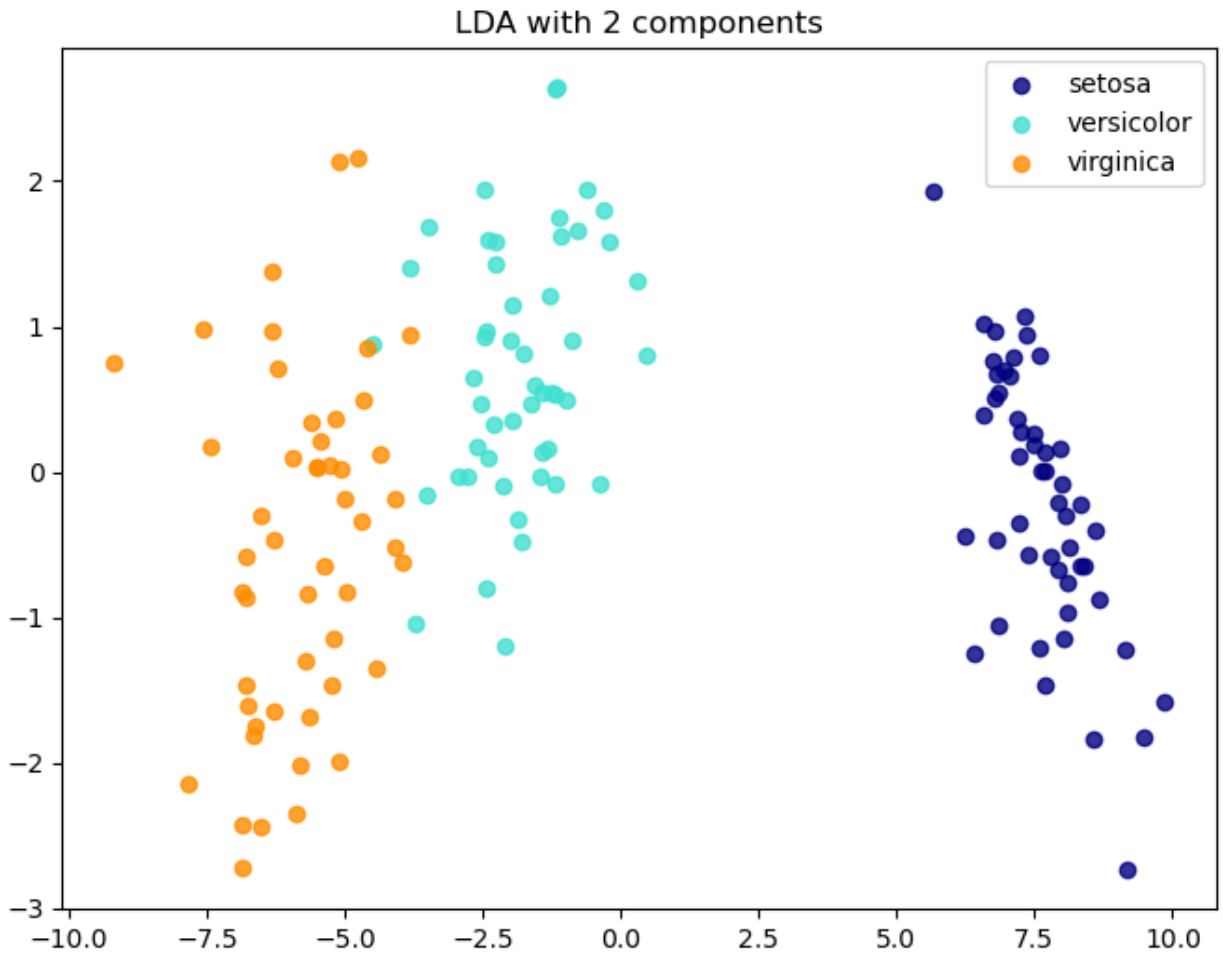
COMPARING SCATTER PLOTS ON ORIGINAL DATASET , LDA WITH COMPONENT 1 AND 2



cluster is not clear on original IRIS DATASET

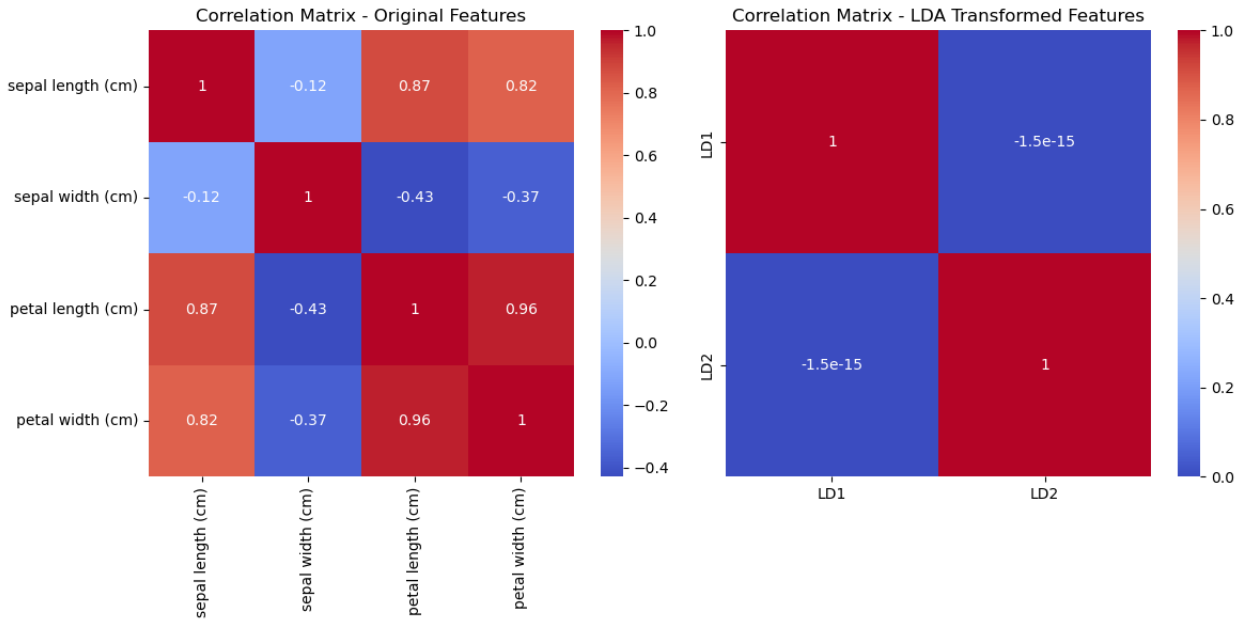


LDA WITH 1 COMPONENT MEANS POINTS ARE PLOTTED ON 1D AXIS WHERE CLUSTERS ARE VISIBLE FAIRLY



CLUSTERS ARE SEPARATED BUT STILL THERE ARE SOME OVERLAPPING

CORRELATION HEAT MAP BEFORE AND AFTER LDA




```
Applying LDA reduced the dimensionality of the dataset from 4
to 2 dimensions.
The classification report for the model trained without LDA s
hows good performance, with high accuracy and F1 scores for a
ll classes.
The classification report for the model trained with LDA show
s similar performance to the model without LDA, indicating th
at LDA was able to reduce the dimensionality without signific
antly affecting the model's performance.
```

MNIST DATASET

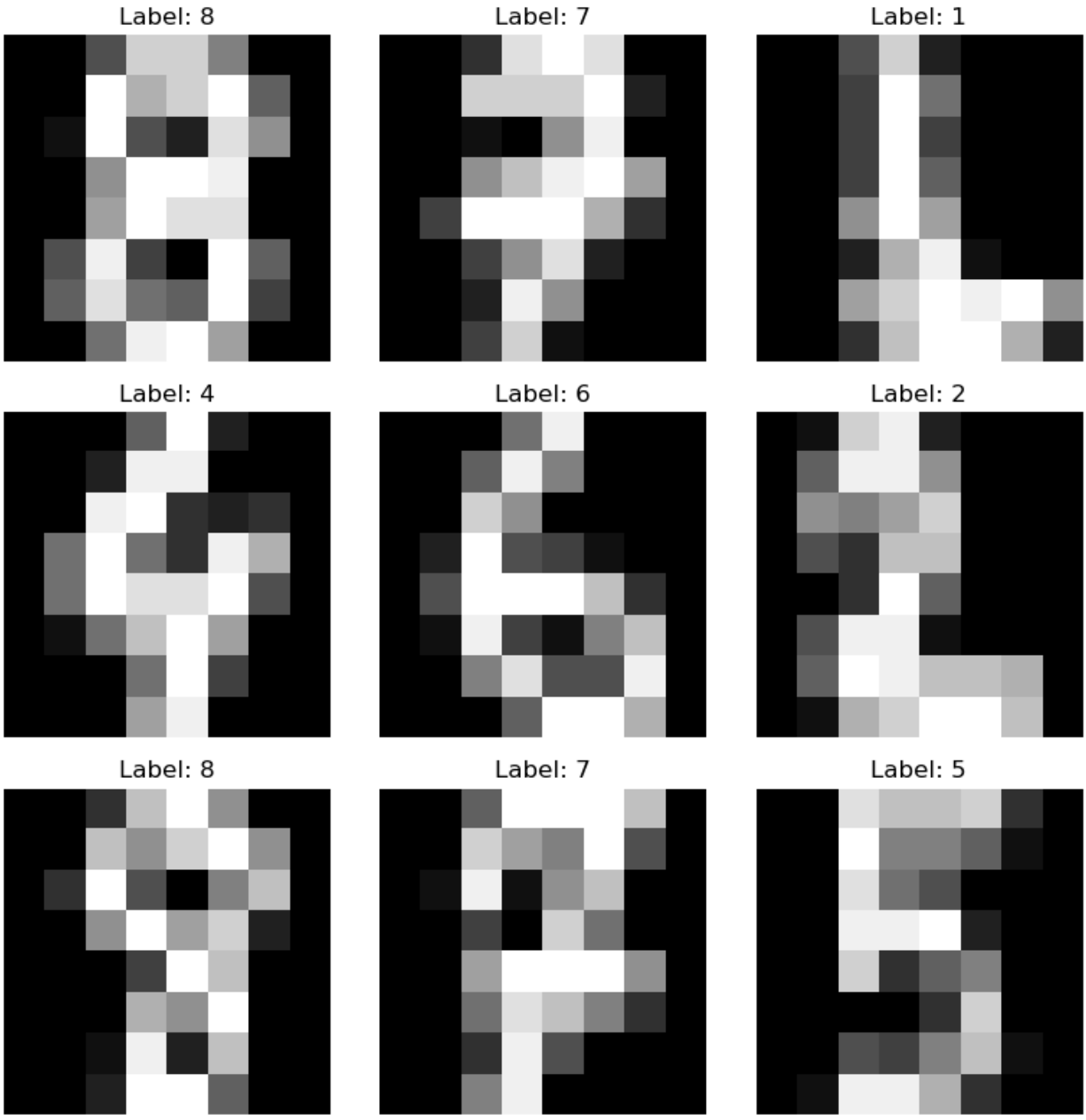
SYNOPSIS OF MNIST DATASET

```
The MNIST dataset is a widely used dataset in machine learnin
g and computer vision. It consists of a collection of 70,000
```

```
small images of handwritten digits (0-9), each 28x28 pixels i
n size. This results in a dataset with a high dimensionality
of 784 features (28x28). The dataset is split into a training
set of 60,000 examples and a test set of 10,000 examples.

The goal of the MNIST dataset is to classify these handwritte
n digits correctly. It is often used as a benchmark dataset f
or testing machine learning algorithms, particularly in the f
ield of image recognition. The simplicity and accessibility o
f the dataset make it a popular choice for beginners to learn
and practice machine learning techniques.
```

```
# Load the MNIST dataset
digits = datasets.load_digits()
X = digits.data
y = digits.target

# Split the dataset into training and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, tes
t_size=0.3, random_state=42)
```

VISUALIZATION OF DATASET



The image is a set of black and white labels with digits ranging from 1 to 8. It is a snippet of the MNIST handwritten digit dataset, which is a common dataset used for training image classification models.

APPLYING LOGISTIC REGRESSION

```
# Standardize the features
sc = StandardScaler()
X_train_std = sc.fit_transform(X_train)
```

```
X_test_std = sc.transform(X_test)

# Supervised machine learning model (Logistic Regression) wit
hout LDA
lr = LogisticRegression(max_iter=10000)
lr.fit(X_train_std, y_train)
y_pred = lr.predict(X_test_std)
print("Classification Report without LDA:")
print(classification_report(y_test, y_pred))
```

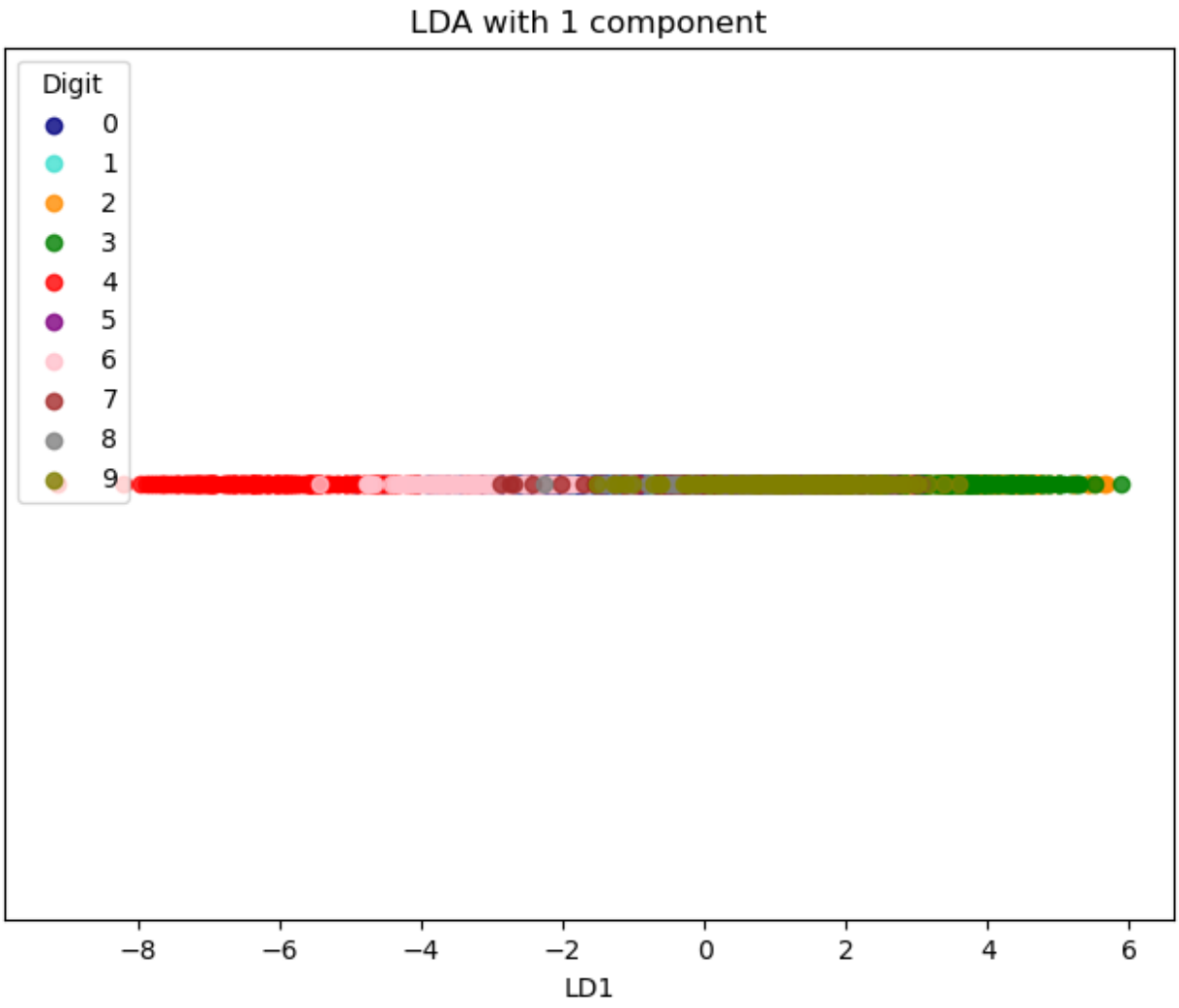
```
Classification Report without LDA:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        53
           1       0.98      0.94      0.96        50
           2       0.94      1.00      0.97        47
           3       1.00      0.93      0.96        54
           4       1.00      0.98      0.99        60
           5       0.95      0.95      0.95        66
           6       0.98      0.98      0.98        53
           7       1.00      0.98      0.99        55
           8       0.89      0.98      0.93        43
           9       0.95      0.97      0.96        59

    accuracy                           0.97       540
   macro avg       0.97      0.97      0.97       540
weighted avg       0.97      0.97      0.97       540
```

APPLYING LDA

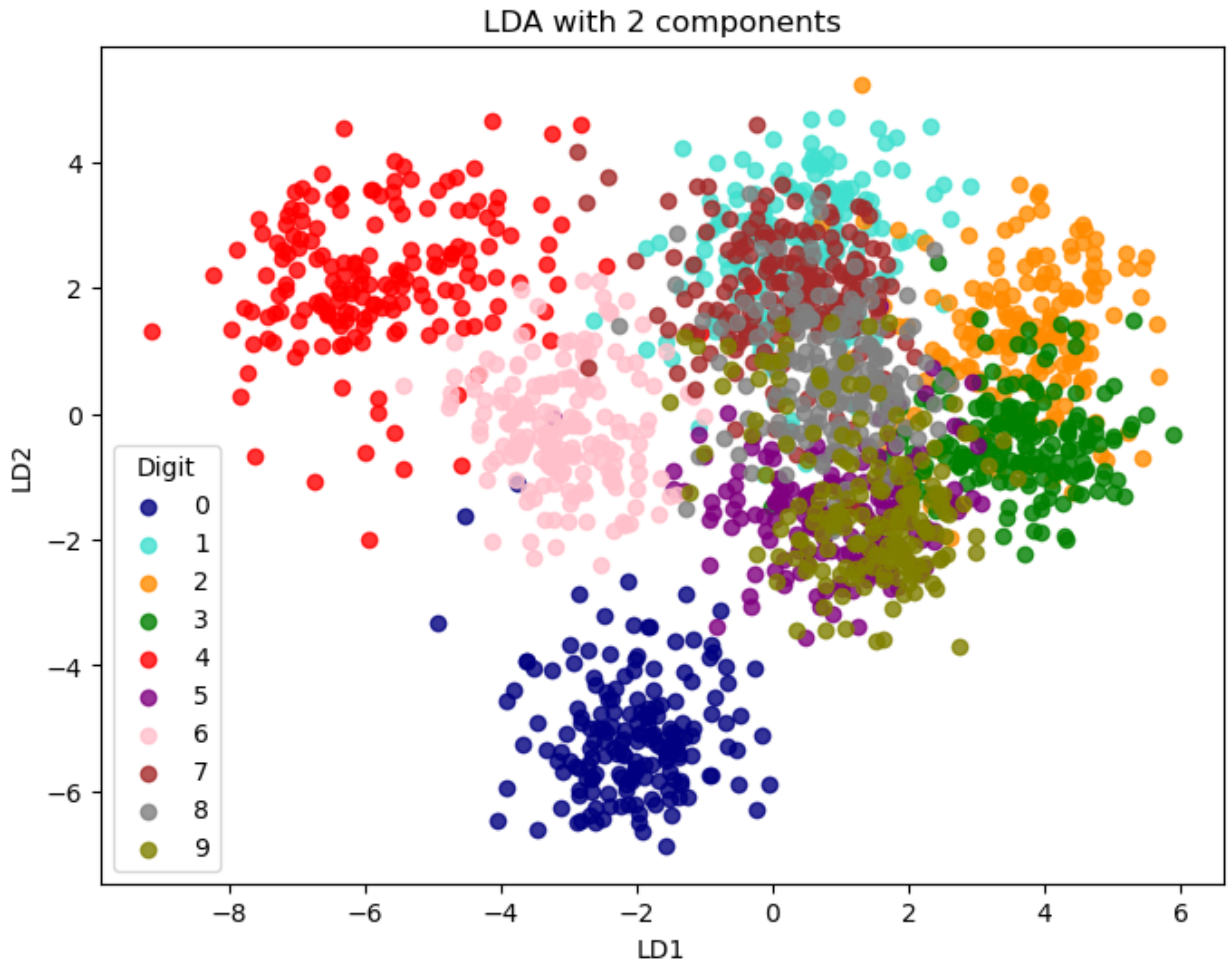
```
# Apply LDA with 1 component
lda1 = LDA(n_components=1).fit(X, y)
X_lda1 = lda1.transform(X)

# Plot LDA with 1 component
plt.figure(figsize=(8, 6))
```

```
for color, i in zip(['navy', 'turquoise', 'darkorange', 'gree
n', 'red', 'purple', 'pink', 'brown', 'gray', 'olive'], range
(10)):
    plt.scatter(
        X_lda1[y == i, 0], np.zeros_like(X_lda1[y == i, 0]),
alpha=0.8, color=color, label=str(i)
    )
plt.legend(loc="best", shadow=False, scatterpoints=1, title
='Digit')
plt.title("LDA with 1 component")
plt.xlabel('LD1')
plt.gca().axes.get_yaxis().set_visible(False)
plt.show()
```

```
# Apply LDA with 2 components
lda = LDA(n_components=2)
X_lda = lda.fit_transform(X, y)

# Plot LDA with 2 components
plt.figure(figsize=(8, 6))
for color, i in zip(['navy', 'turquoise', 'darkorange', 'gree
n', 'red', 'purple', 'pink', 'brown', 'gray', 'olive'], range
(10)):
    plt.scatter(
        X_lda[y == i, 0], X_lda[y == i, 1], alpha=0.8, color=
color, label=str(i)
    )
plt.legend(loc="best", shadow=False, scatterpoints=1, title
='Digit')
plt.title("LDA with 2 components")
plt.xlabel('LD1')
plt.ylabel('LD2')
plt.show()
```



Cluster is not visible no inference can be taken out clearly

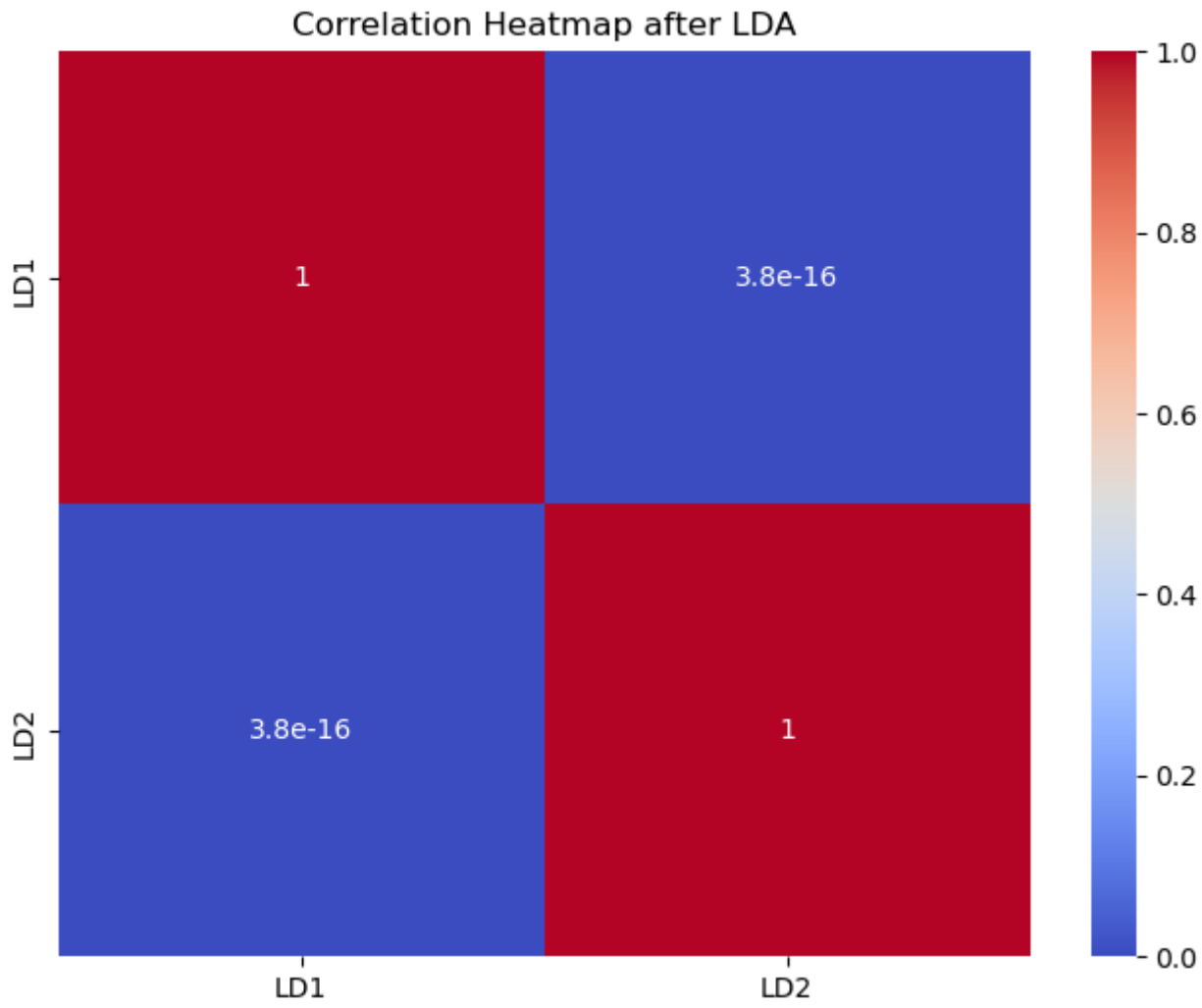


It can be seen that the clustering is not clear with LDA which gives us some inference that it may not give good results

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

           0       0.96      0.94      0.95        53
           1       0.73      0.72      0.73        50
           2       0.85      0.83      0.84        47
           3       0.76      0.87      0.81        54
           4       0.96      0.87      0.91        60
           5       0.59      0.53      0.56        66
           6       0.76      0.89      0.82        53
           7       0.56      0.45      0.50        55
           8       0.50      0.63      0.56        43
           9       0.49      0.47      0.48        59

    accuracy                           0.71       540
   macro avg       0.72      0.72      0.72       540
weighted avg       0.72      0.71      0.71       540
```

So the accuracy shows that the performance after applying LDA decreases so it is preferable to use this dataset without applying LDA

Correlation heat map


Correlation Heatmap after LDA
1
3.8e-16
3.8e-16
1
LD1
LD2
LD1
LD2
1.0
0.8
0.6
0.4
0.2
0.0