Assignment 2

Devishree Jothilingam

I have compared two dimensionality reduction techniques PCA (Principle Component Analysis) and LDA (Linear Discriminant Analysis).

Both these algorithms are Linear dimensionality techniques.

The Key idea is to preserve the most relevant data by reducing the volume of the dataset.

Exploring the dataset

Training data shape: (60000, 784)

Training labels shape: (60000,)

Testing data shape: (10000, 784)

Testing labels shape: (10000,)

Logic:

Used the scikit-learn library to implement the PCA and LDA techniques.

Before visualizing the result, converted the class labels to numeric labels using label encoder.

The first two components are plotted against each other for analysis in both PCA and LDA techniques.

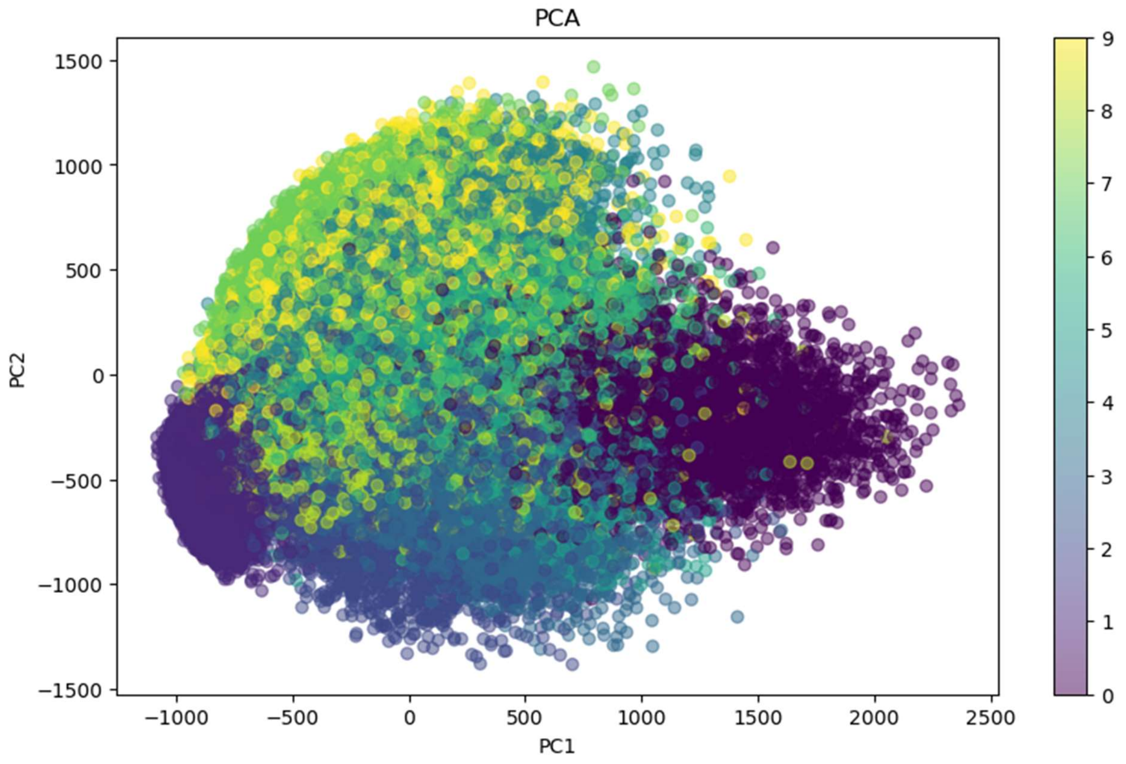
Comparison

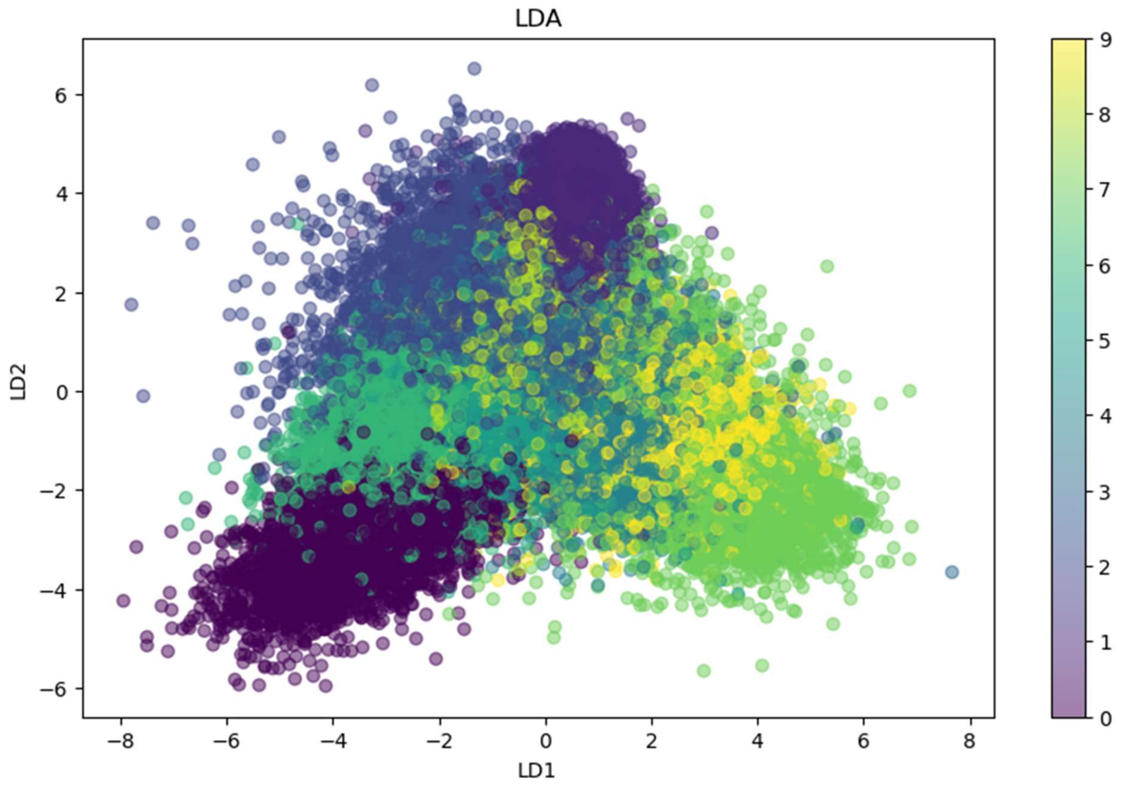
The first plot shows the two-dimensional representation of the MNIST dataset obtained by applying PCA. The data points are coloured based on their true labels, and we can see that there is some separation between the clusters of digits, but there is also some overlap. This suggests that PCA is able to capture some of the underlying structure of the data, but not all of it.

The second plot shows the two-dimensional representation of the MNIST dataset obtained by applying LDA. Again, the data points are coloured based on their true labels. We can see that there is more separation between the clusters of digits compared to the PCA plot. This suggests that LDA is able to capture more of the underlying structure of the data than PCA.

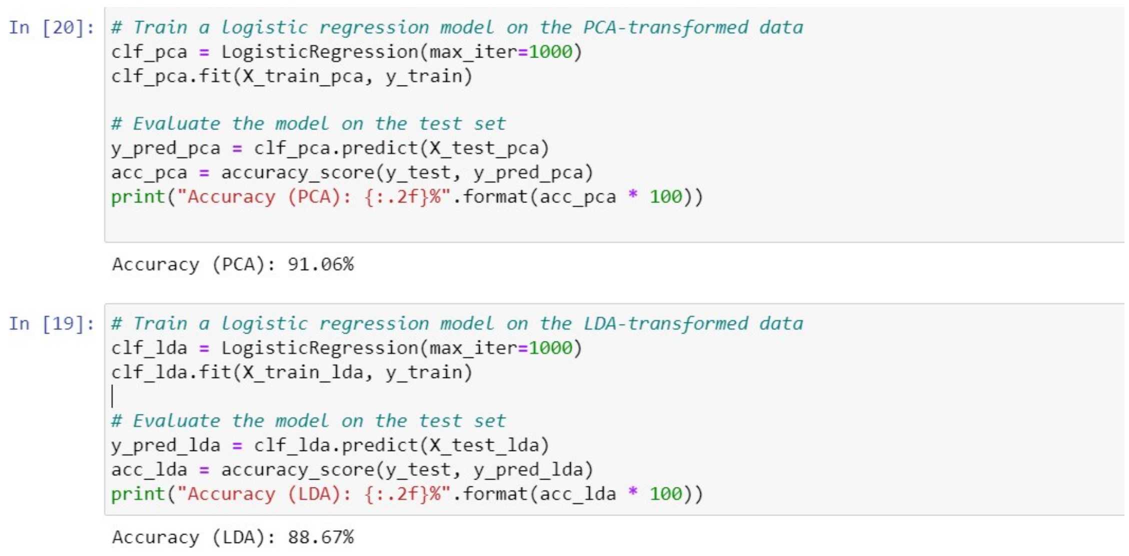
Overall, the LDA plot seems to be a better representation of the MNIST dataset than the PCA plot, as it shows clearer separation between the clusters of digits. However, it is worth noting that this is just a two-dimensional representation of a high-dimensional dataset, and it is possible that other visualizations or analysis methods may yield different insights.

Output


PCA
PC1
PC2
1500
1000
500
0
−500
−1000
−1500
−1000
−500
0
500
1000
1500
2000
2500
0
1
2
3
4
5
6
7
8
9


LDA
LD1
LD2
6
4
2
0
−2
−4
−6
−8
−6
−4
−2
0
2
4
6
8
0
1
2
3
4
5
6
7
8
9

Later Implemented Logistic Regression to the processed data.



```
In [20]: # Train a logistic regression model on the PCA-transformed data
         clf_pca = LogisticRegression(max_iter=1000)
         clf_pca.fit(X_train_pca, y_train)

         # Evaluate the model on the test set
         y_pred_pca = clf_pca.predict(X_test_pca)
         acc_pca = accuracy_score(y_test, y_pred_pca)
         print("Accuracy (PCA): {:.2f}%".format(acc_pca * 100))

         Accuracy (PCA): 91.06%

In [19]: # Train a logistic regression model on the LDA-transformed data
         clf_lda = LogisticRegression(max_iter=1000)
         clf_lda.fit(X_train_lda, y_train)

         # Evaluate the model on the test set
         y_pred_lda = clf_lda.predict(X_test_lda)
         acc_lda = accuracy_score(y_test, y_pred_lda)
         print("Accuracy (LDA): {:.2f}%".format(acc_lda * 100))

         Accuracy (LDA): 88.67%
```