GE461 – Programming Assignment 2
Muhammad Arham Khan
21701848
16 April, 2021

## Question 1

**1.1)** This is a plot of the eigenvalues from the matrix plotted in a decreasing order. As shown in the graph, the variance between the variables decreases after around the 70th component and hence, the eigenvalues converge after around 70 component values. Hence, I think I would chose around the first 70 components

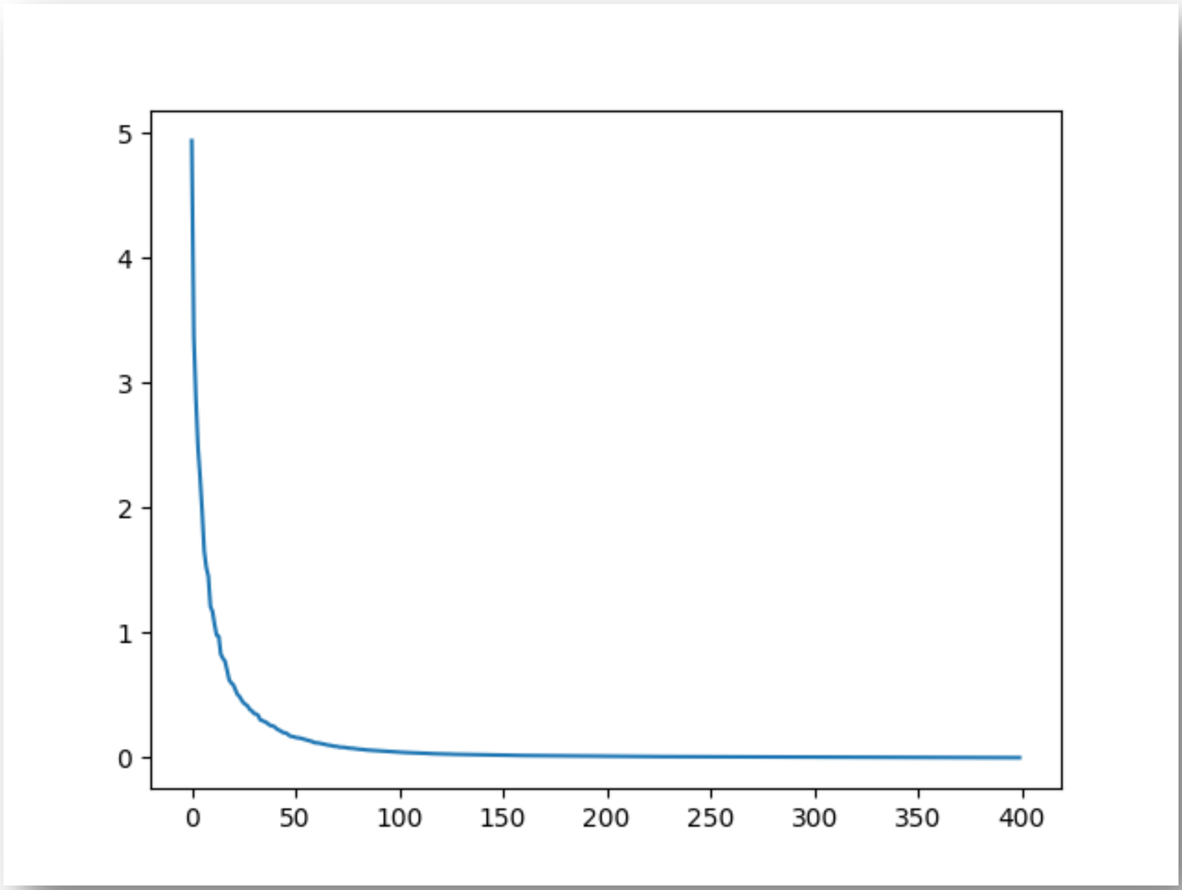

*Figure 1: the 400 eigenvalues*

**1.2)** Analyzing all the eigenvalues, it can be seen that the most amount of variance arises on the rounder shapes in the provided data matrix. So using this inference, we can classify the digits among types of circular and sharp-edged. For instance, values that solely comprise of circular shapes are 6, 8, 9 and shapes that have sharp edges are 1, 2, 3, 4, 5, 7 and so, a classification can be created here using the amount of variance of circular and non-circular shapes. As shown in the 70 eigenvalue and mean picture, the highest variance is achieved by circular shaped digits.

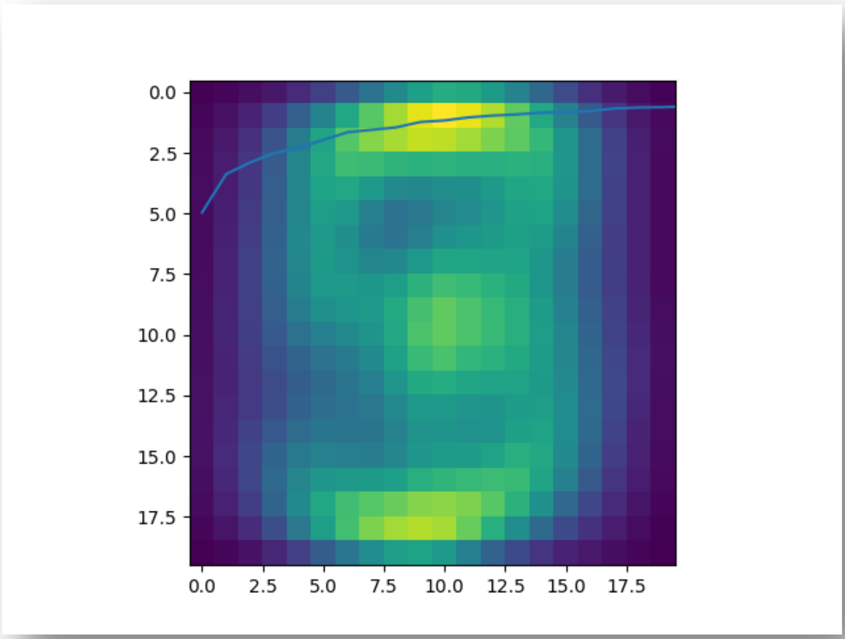

*Figure 2: Dataset's mean*

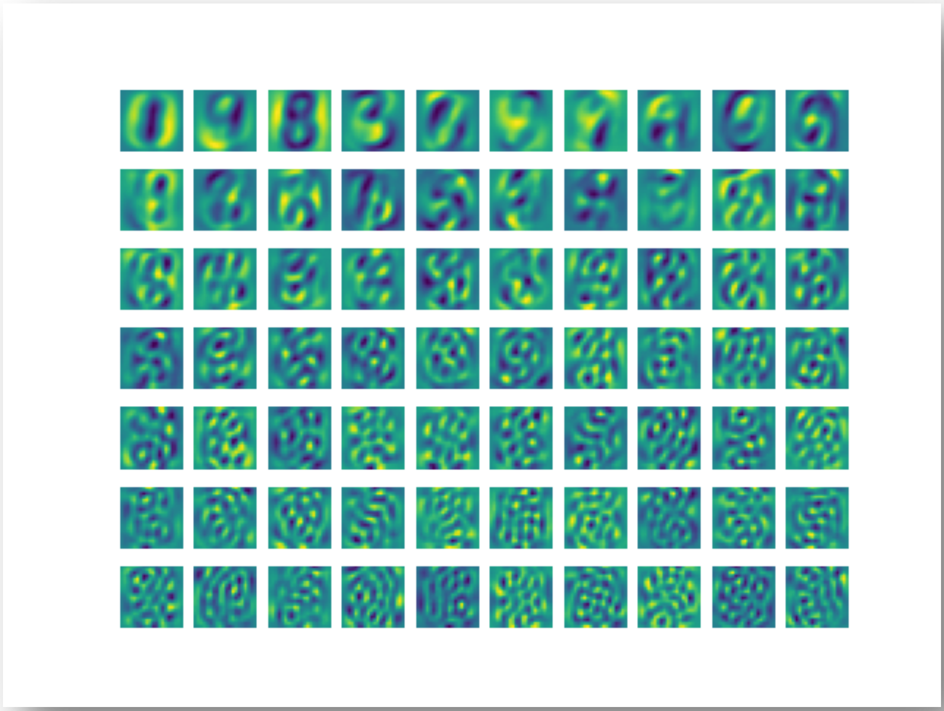

*Figure 3: 70 eigenvectors from the dataset*

**1.3)**

| SUBSPACE DIMENSION | PROJECTION OUTPUT |
|---|---|
| 10 | **0.80952381** |
| 20 | **0.83073229** |
| 30 | **0.82633053** |
| 40 | **0.81832733** |
| 50 | **0.80552221** |
| 60 | **0.79551821** |
| 70 | **0.77991196** |
| 80 | **0.767507** |
| 90 | **0.75630252** |
| 100 | **0.74469788** |
| 110 | **0.72589036** |
| 120 | **0.71068427** |
| 130 | **0.71228491** |
| 140 | **0.69427771** |
| 150 | **0.68467387** |
| 160 | **0.66426571** |
| 170 | **0.62785114** |
| 180 | **0.58903561** |
| 190 | **0.54221689** |
| 200 | **0.49859944** |

**Table 1: Test data Gaussian Prediction Accuracies**

| SUBSPACE DIMENSION | PROJECTION OUTPUT |
|---|---|
| 10 | **0.83126749** |
| 20 | **0.85645742** |
| 30 | **0.86645342** |
| 40 | **0.86285486** |
| 50 | **0.86005598** |
| 60 | **0.85645742** |
| 70 | **0.8472611** |
| 80 | **0.84086365** |
| 90 | **0.83086765** |
| 100 | **0.82606957** |
| 110 | **0.82247101** |
| 120 | **0.81407437** |
| 130 | **0.80327869** |
| 140 | **0.79808077** |
| 150 | **0.78528589** |
| 160 | **0.77608956** |
| 170 | **0.74730108** |
| 180 | **0.70571771** |
| 190 | **0.64134346** |
| 200 | **0.57337065** |

**Table 2: Training data Gaussian Prediction Accuracies**

**1.4)**
**Legend**
Test = Blue
Training = Orange

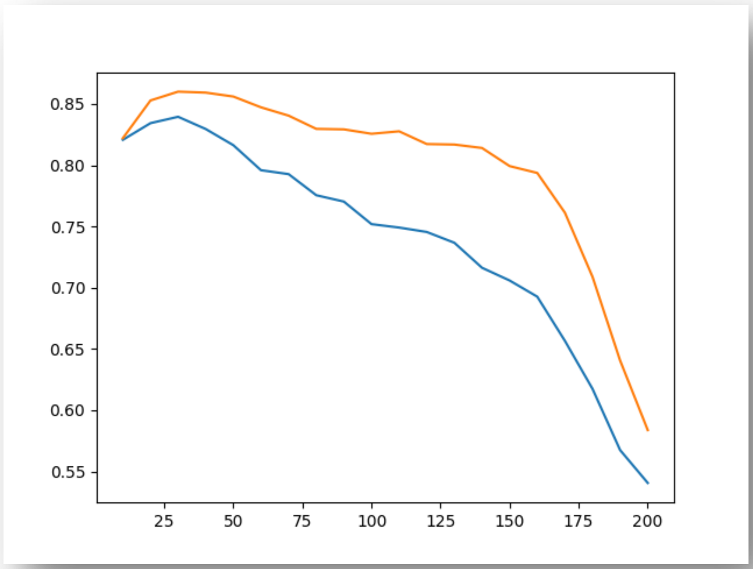

*Figure 4: PCA output (component at x-axis/ classification accuracy at y)*

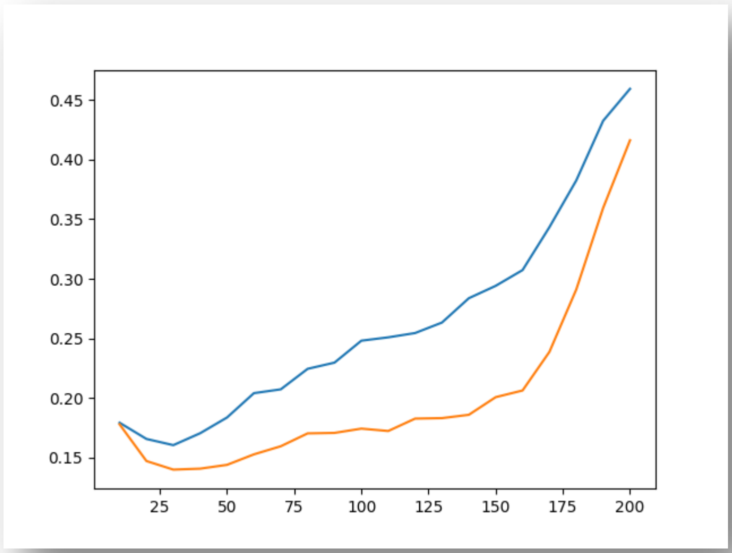

*Figure 5: PCA Output (component at x-axis/ classification error at y)*

Since the blue line is always higher than the orange one, the accuracies of the train data are consistently higher than the test data's accuracies. As far as the prediction of the optimum number of components goes, the highest accuracy is achieved near a value of 40, so the initial prediction (in part 1) might be slightly wrong.

Analyzing the second graph, it is evident that the error probability increases as the number of components increase. This agrees with the PCA theory as more components cause additional weightage given to less necessary features in the dataset. So a higher chance of errors.

## Question 2

### 2.1) Plotting the bases of LDA

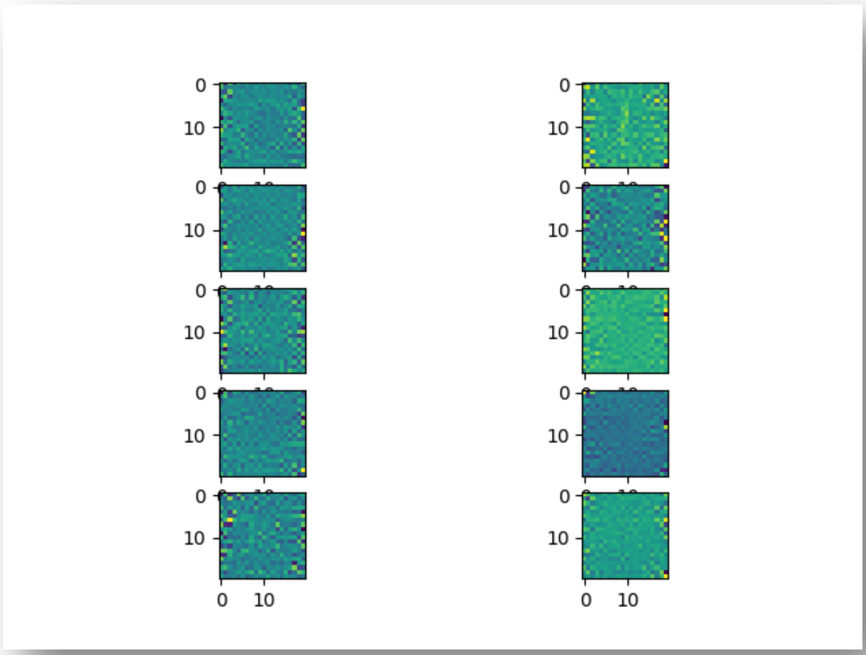

*Figure 6: 10 bases obtained using LDA*

This similarity in some bases and high difference in the others is expected according to the theory of LDA method. The similarity in some and the distinction in others represents specific features of the bases and helps in differentiating between the different digits. This happens because LDA gives more weight to classes as compared to other methods like PCA.

### 2.2)

| Subspsace dimension | Accuracies |
|---|---|
| 1 | 0.36814726 |
| 2 | 0.58863545 |
| 3 | 0.70828331 |
| 4 | 0.71588635 |
| 5 | 0.77430972 |
| 6 | 0.81712685 |
| 7 | 0.82793117 |
| 8 | 0.82873149 |
| 9 | 0.81592637 |

Table 3: Accuracies against subspace dimensions on test data

| Subspsace dimension | Accuracies |
|---|---|
| 1 | 0.39984006 |
| 2 | 0.67413035 |
| 3 | 0.80047981 |
| 4 | 0.81767293 |
| 5 | 0.8672531 |
| 6 | 0.91523391 |
| 7 | 0.92962815 |
| 8 | 0.93682527 |
| 9 | 0.93882447 |
| | |

Table 4: Accuracies against subspace dimensions on train data

**2.3)**

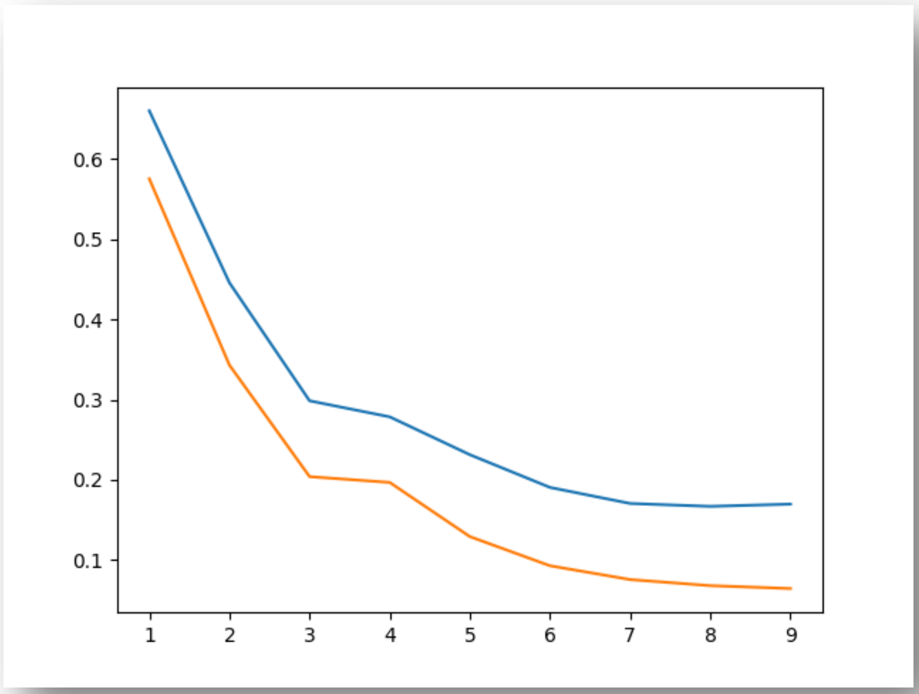

*Figure 7: LDA ouput (subspace dimensino on x-axis/ class .error on y)*

Referring to the graph plotted, the classification error decreases using the LDA approach as compared to the PCA approach. Since LDA clutters features together, in a higher number of dimentions, the clusters predict the accuracy better with higher data points hence reducing the error percentage.

**REFERENCES:**

- Loading mat files: https://towardsdatascience.com/how-to-load-matlab-mat-files-in-python-1f200e1287b5
- Scipy.io
- Numpy
- sklearn