GE 461 Introduction to Data Science

Project 2 Report

Fahad Waseem Butt

21801356

Spring 2022

TABLE OF CONTENTS

INTRODUCTION:

Digit recognition is an important problem in the modern data landscape, and the MNIST database of handwritten digits [1] was used to explore problems regarding Dimensionality Reduction and Visualization. In this project, the digits dataset of a total of 5000 patterns was randomly split into training and test sets which had 2500 patterns each. The training and test sets were stored in a separate file "data.mat", which is given in the zip file, along with the code for doing so which is named "Split_Data.m". It may be noted that each time "Split_Data.m" file is run, random rearrangement of data takes place [2]. The purpose of storing training and test data and corresponding labels in "data.mat" was to use the same data for our analysis in all questions to follow.

QUESTION 1:

In this question, the requirement was to make use of Principal Components Analysis (PCA) to project the 400-dimensional data onto lower dimensional subspaces to observe the effect of reduced dimensionality on the performance of the Gaussian classier. To accomplish this, PCA was used to find a number of eigenvectors which are made use of in mapping the data onto lower dimensional subspaces. A built-in function "pca" available in Matlab was used to carry out PCA on raw data [3].

Q1.1 SOLUTION:

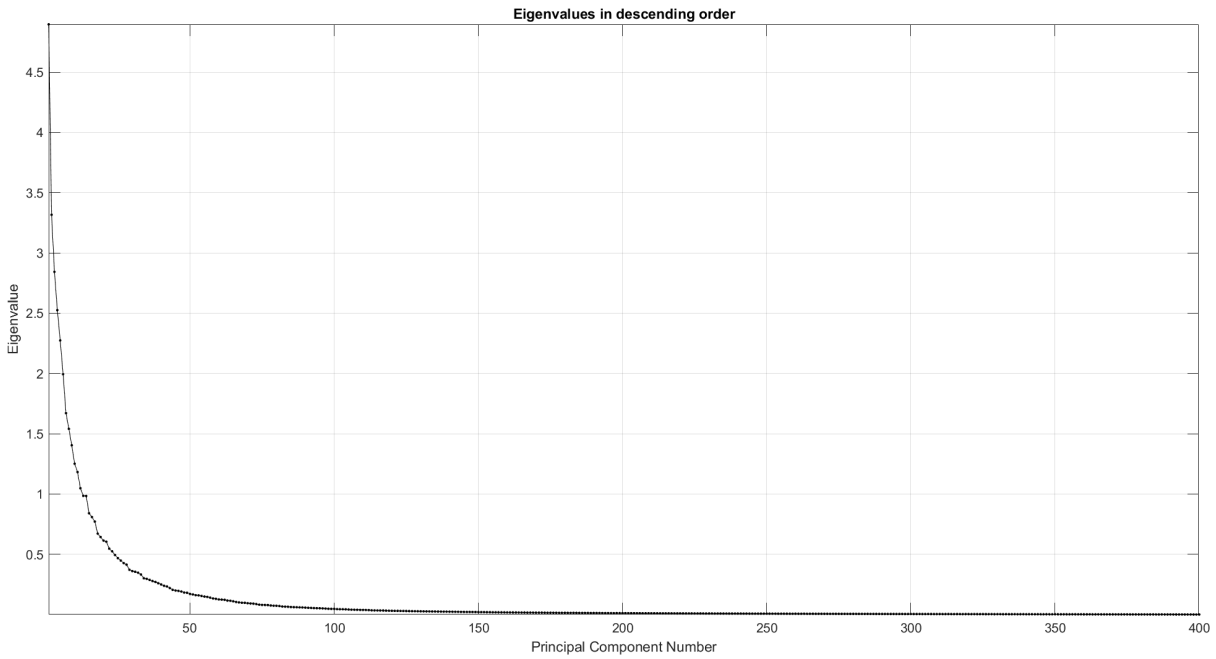

Figure 1: Eigenvalues vs. Number of Components Plot [4]

When the normalized eigenvalues (y-axis) are arranged in descending order, it is noted that as the number of components (x-axis) [4] increases, the variance decreases, with it being the highest at the first principal component. It can be seen in the plot that after crossing 250 components, there is no significant decrease in the variance. After crossing 100 components and upto 250 components, there is a noticeable decrease in the variance, but not significant enough to be noteworthy. Therefore, using just 20 to 40 components would be enough to allow for reliably accurate predictions, and so the number of features to consider can be significantly reduced, with further components not necessarily being required.

Q1.2 SOLUTION:

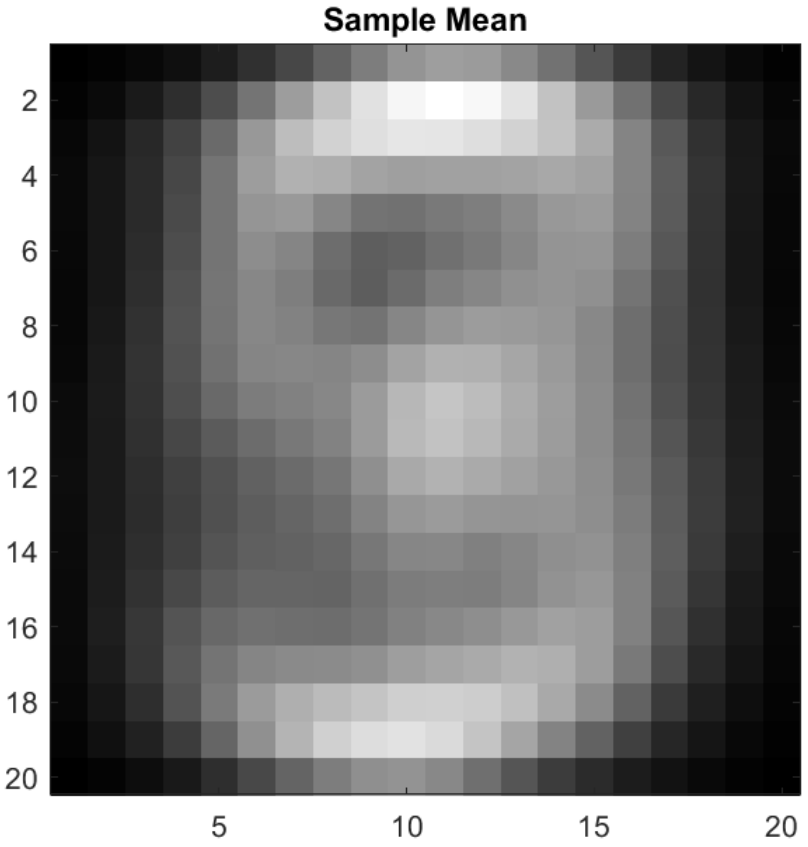

Figure 2: Sample Mean for Whole Training Dataset as an Image [5]

As could be expected, the sample mean [5] of the whole training dataset looks quite similar to the digits “8” or “9”. When considering the digits for numbers 0 to 9, the digit 8 shares features from each of the numbers, with 9 also doing the same to an extent.

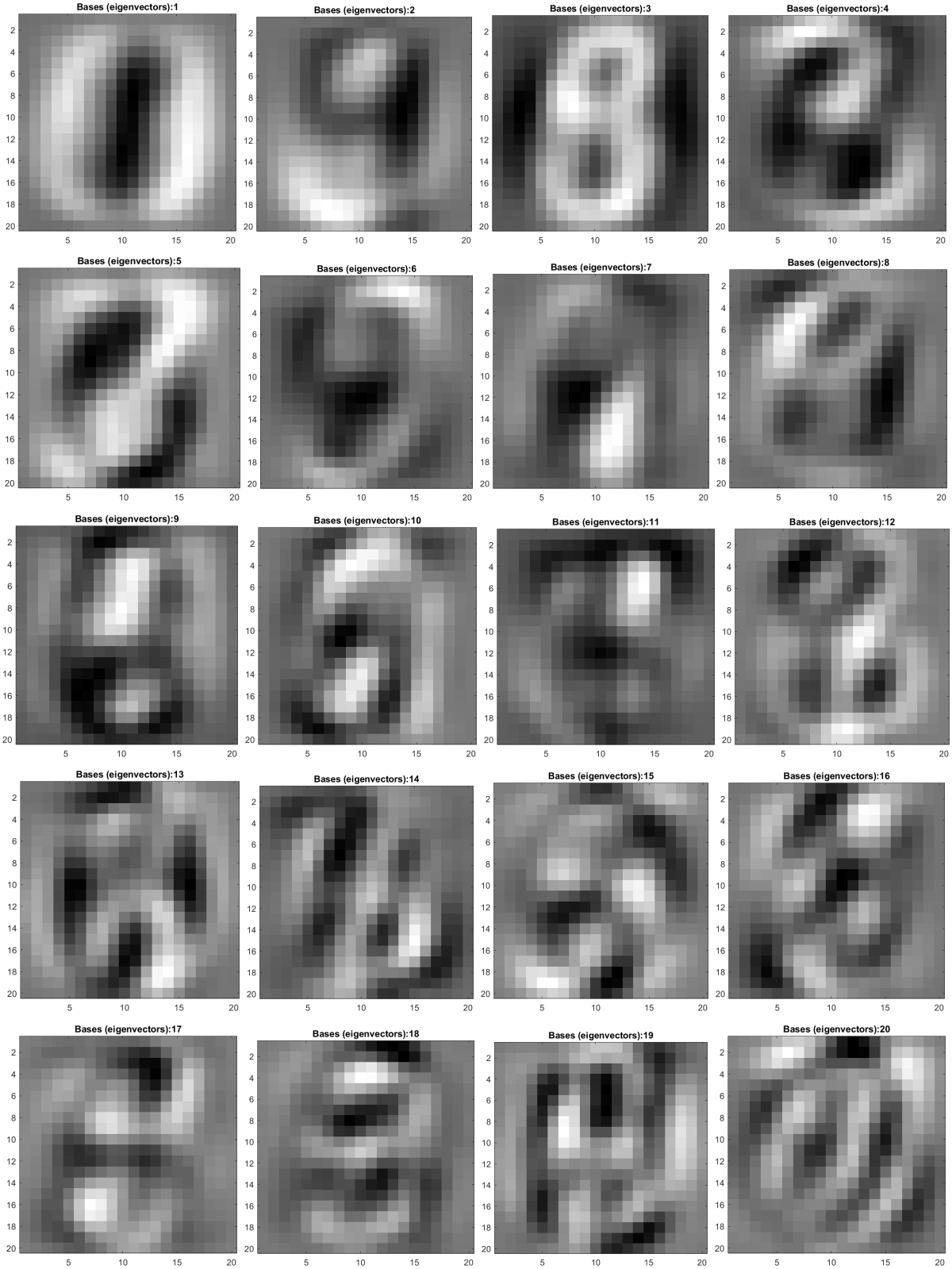

Figure 3: Chosen Bases (Eigenvectors) as Images [5] [6]

When looking at the chosen bases, it can be noted that the images [5] [6] depict quite similar to either the digits from 0 to 9, or similar to some combination of them. This was also

expected as the components selected to make predictions were supposed to have information regarding the digits in the bases.

Q1.3 SOLUTION:

Matlab toolbox for "pca" [3] was used in which 200 subspace dimensions were chosen for the Training Dataset. Later to meet the requirement of selecting at least 20 different subspace dimensions was met. This was done by selecting subspace dimensions 1 ~ 50, 100 ~ 145 and 150 ~ 195 with a step of 5 so as to select a total of 30 different subspace dimensions. In this way the effect of selecting subspace dimensions over a larger domain can be evaluated/ visualized easily. A Gaussian classifier was trained using data in each subspace [7], with half being used for training and the other half for testing.

Q1.4 SOLUTION:

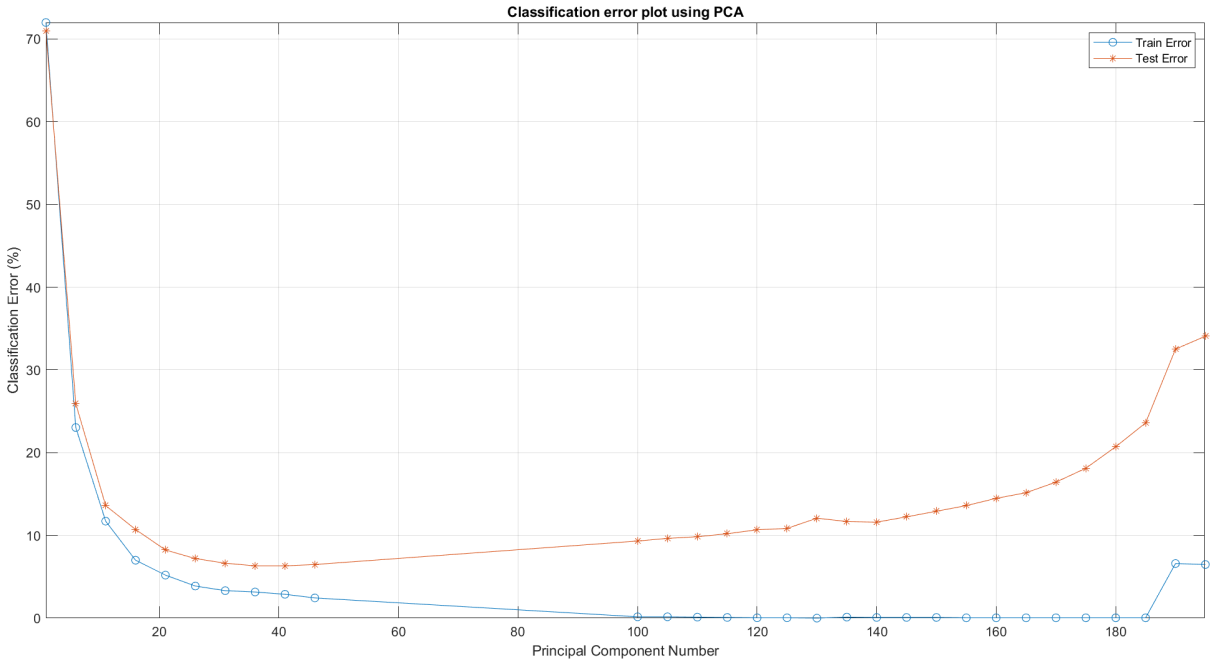

Figure 4: PCA Classification Error vs. Number of Components Plot (for train and test sets)

In a previous part, it was determined that using just 20 to 40 components would be enough to allow for reliably accurate predictions. This number of dimensions estimated prior, seems to be the number of components where the model performs at its best when considering both the train and test errors. The training error decreases with an increase in the number of components. The error on the testing set decreases up until about 40 principal components, but begins to increase again after that. This increase in error after 40 components is likely due to the model overfitting with the training data.

QUESTION 2:

In this question, the requirement was to use Fisher linear discriminant analysis (LDA) to project the 400-dimensional data onto lower dimensional subspaces. To accomplish this, LDA was used to find a number of bases which are made use of in mapping the data onto lower dimensional subspaces. Matlab toolbox for dimensionality reduction was used where code for LDA is available [8].

Q2.1 SOLUTION:

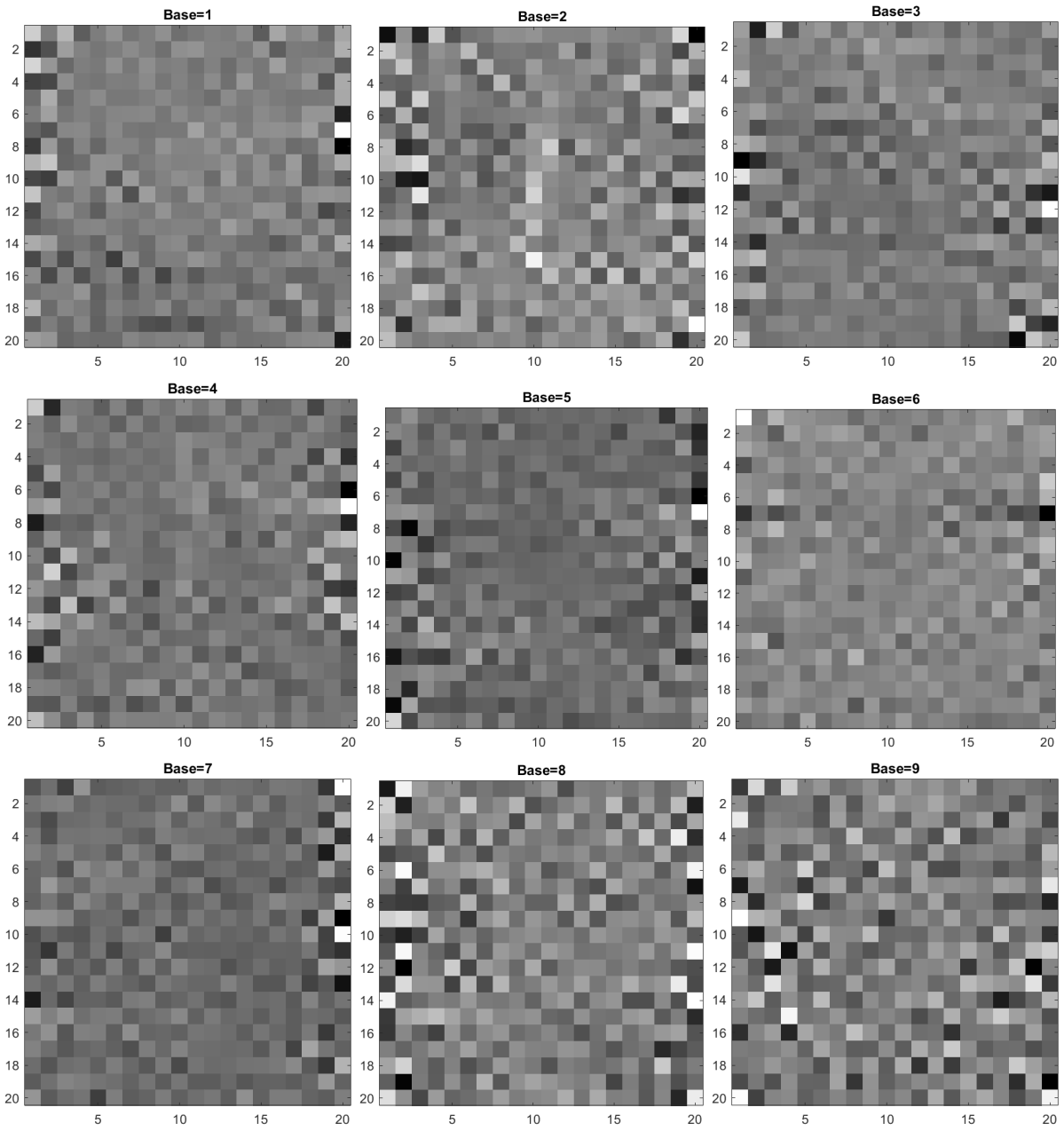

Figure 5: Set of Bases as Images

There are 10 classes in the dataset, and consequently there are 9 bases found. When considering the PCA case from before, the bases seemed similar to the digits in the dataset because PCA uses the data to make them while ignoring the class labels [9]. On the other hand, LDA is aware of the class labels and so uses the labels to differentiate between the data by maximizing the separation between the classes [9], and as a result, the bases found in LDA are very different from a digit, and seem to be undecipherable and illegible in comparison.

Q2.2 SOLUTION:

Matlab toolbox for dimensionality reduction was used where code for LDA is available [8], where 9 subspace dimensions (as it is the largest possible for LDA with a dataset containing 10 classes) were chosen, with the entire data (both training data and test data using the transformation matrix) being projected onto these subspaces. A Gaussian classifier was trained using data in each subspace, with half being used for training and the other half for testing. The working is very similar to Question 1 [7].

Q2.3 SOLUTION:

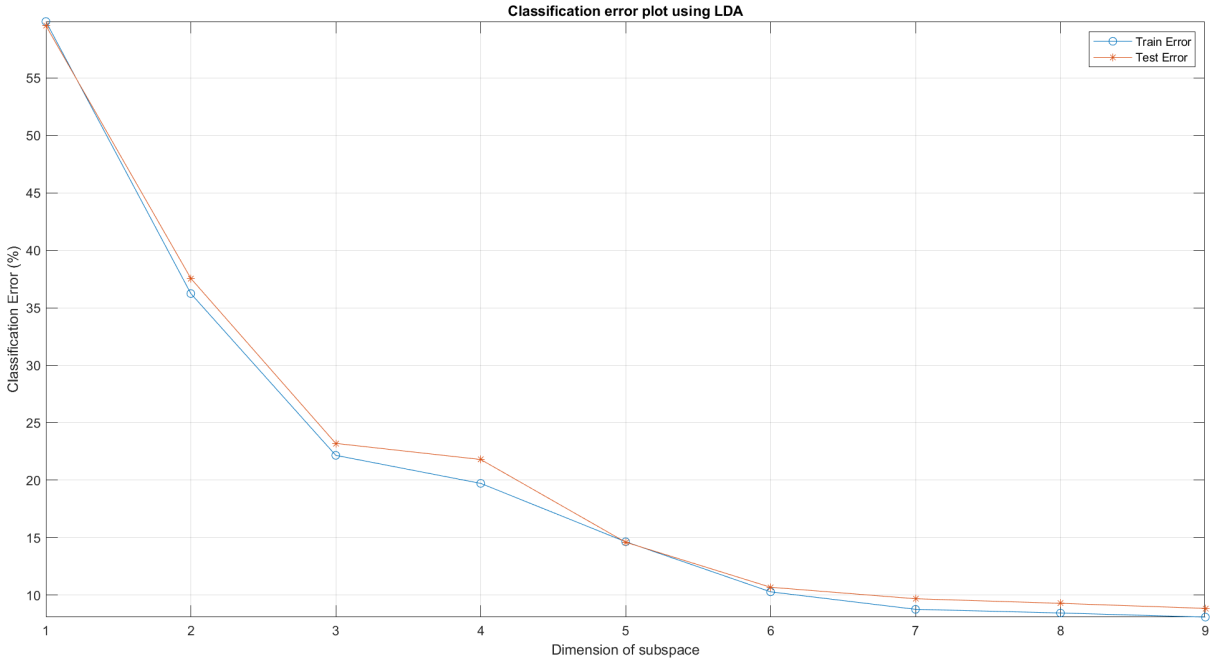

Figure 6: LDA Classification Error vs. Dimension of Subspace Plot (for train and test sets)

As it is the largest possible for LDA with a dataset containing 10 classes, only 9 subspace dimensions were chosen for this model. This time, as the number of dimensions of subspace increases, both the train and test errors decrease in a similar fashion (there is no increase in the error after a decrease this time). In the Gaussian model for the PCA section, the model was most likely overfitting with the training data. However, in the Gaussian model for the LDA, the largest possible number subspace dimensions is 9, and so the model cannot overfit.

As such, when compared with the PCA case, a better result is observed in the LDA case as the dimensionality has been reduced.

QUESTION 3:

Q3.1 SOLUTION:

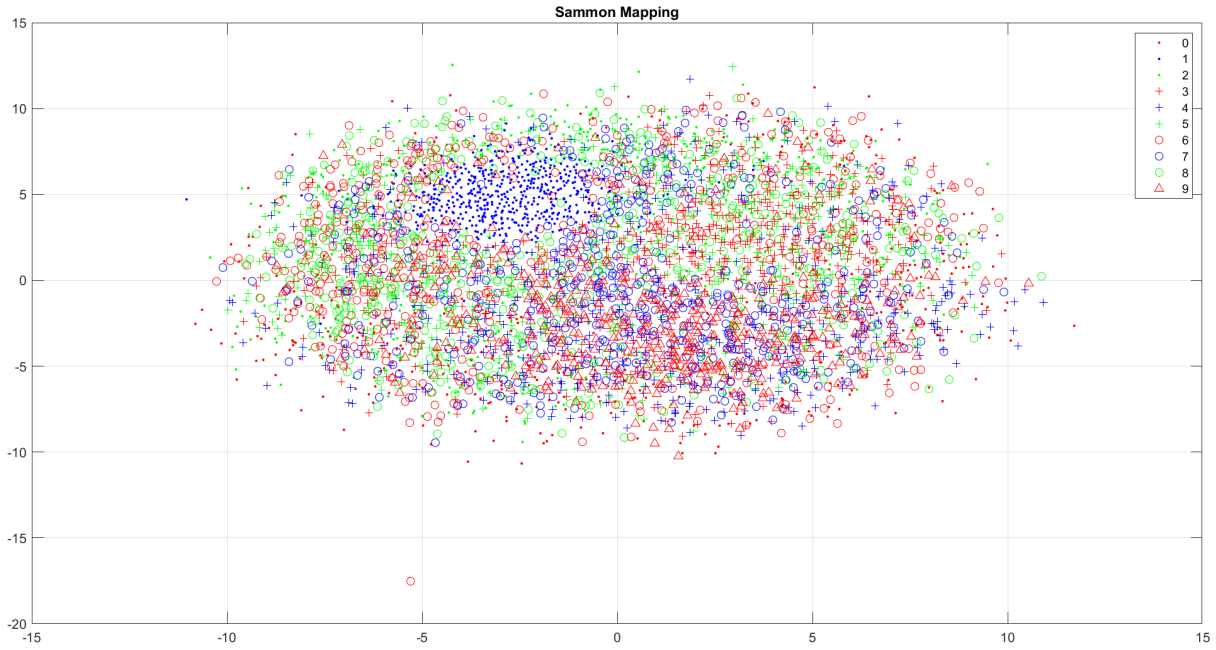

Figure 7: Sammon's Mapping Scatter Plot

Matlab toolbox for dimensionality reduction was used where code for Sammon mapping is available [10] and further reference was taken from Miscellaneous MATLAB Software - UEA Computational Biology Laboratory [11]. In this implementation, in the Sammons options, the MaxIter was set to 150 and TolFun to 10^-5, with the other options being set to defaults. This was chosen so as to be able to successfully run the implementation with the least epochs while achieving the smallest error in the limited number of epochs. After 100 epochs, the error was 10.29643518% and since there were diminishing returns after further increase, the program options were set so it terminated early, giving an error at the end of 7.33905997%, which is a good result. As Sammon mapping aims to find structures in higher dimensional data, where there are geometric relations in the subsets of the data [12], the scatter plot shows limited dimensions so it may not be easily interpreted by human eyes.

Q3.2 SOLUTION:

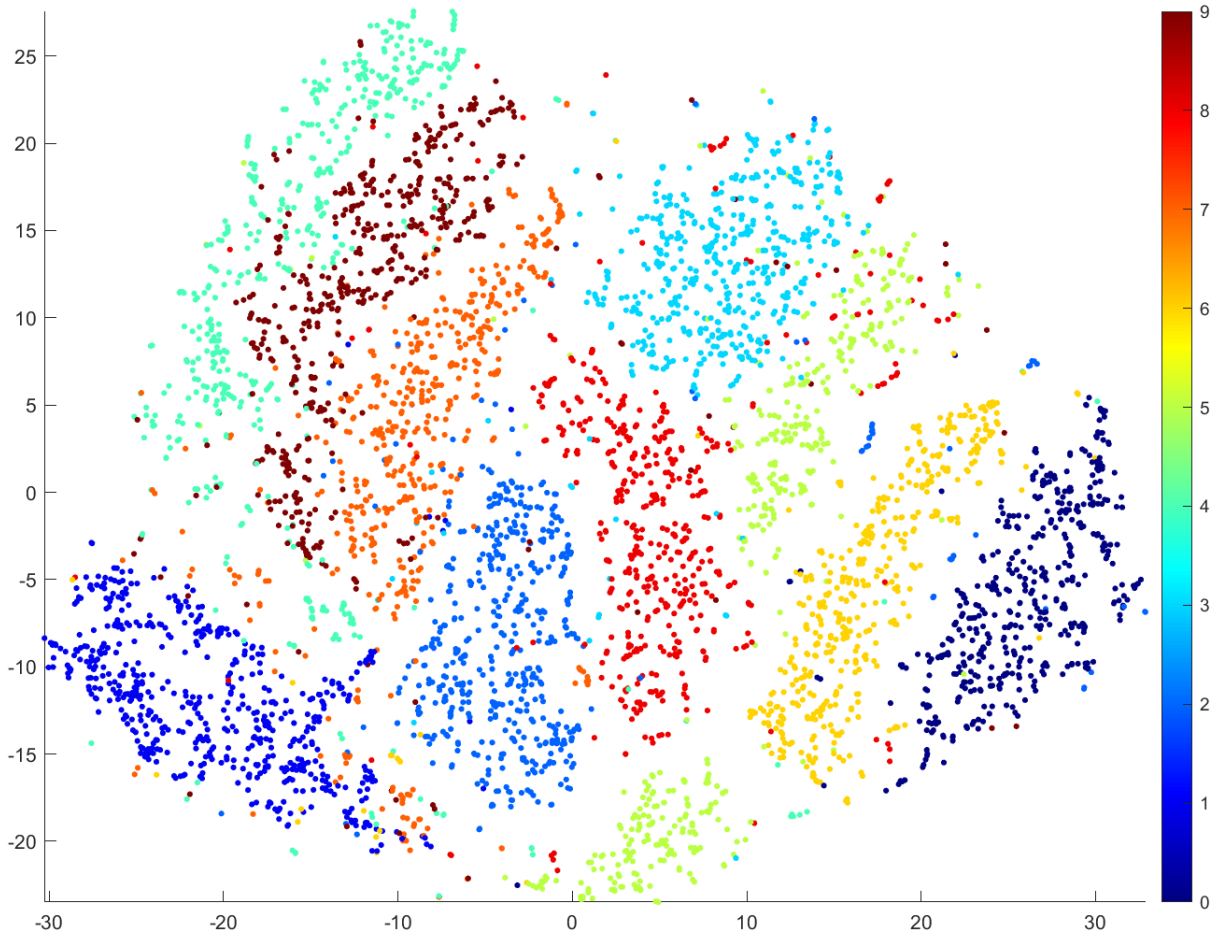

Figure 8: t-SNE Scatter Plot

Matlab toolbox for dimensionality reduction was used where code for t-SNE is available [13][14][15][16]. In this implementation, in the t-SNE options, the MaxIter was set to 250 and Perplexity to 50, with the other options being set to defaults. The number of iteration and perplexity (number of points to whom the distance to preserve) are the most important parameters of t-SNE [17], and it was noted that finding good balance to these settings, a good result was achieved. Because of diminishing returns at higher iterations, there was an error of 1.6164% at iteration 250, and after checking further values, it was most efficient to stop at 250 iterations. As t-SNE views the dimensions of the data as clusters [17], with the spread of the data in this scatter plot, it can be noted that the human observer can tell the difference between the classes, so the data can be interpreted from the figure.

REFERENCES

[1] Y. LeCun, C. Cortes, C. J.C. Burges. “MNIST handwritten digit database, Yann LeCun, Corinna Cortes and Chris Burges”. http://yann.lecun.com/exdb/mnist/

[2] “how can I separate data randomly?”. Mathworks. April 2015. https://www.mathworks.com/matlabcentral/answers/214251-how-can-i-separate-data-randomly

[3] “Principal component analysis of raw data - MATLAB pca”. Mathworks. https://www.mathworks.com/help/stats/pca.html

[4] “Applying PCA to Handwritten Digits Dataset in MATLAB - PCA in Python and MATLAB”. YouTube. January 2020. https://www.youtube.com/watch?v=KhogS53oDFc

[5] B. Milanko. “Solving MNIST with Principle Component Analysis”. BenMilanko https://benmilanko.com/projects/mnist_with_pca/

[6] J. T. VanderPlas. “In Depth: Principal Component Analysis | Python Data Science Handbook”. Python Data Science Handbook. Beijing: O'Reilly, 2016. https://jakevdp.github.io/PythonDataScienceHandbook/05.09-principal-component-analysis.html

[7] A. C. Kapourani “Classification with Gaussians”. The University of Edinburgh. https://www.inf.ed.ac.uk/teaching/courses/inf2b/labs/learn-lab7.pdf

[8] “Matlab-Toolbox-for-Dimensionality-Reduction”. Github. https://github.com/UMD-ISL/Matlab-Toolbox-for-Dimensionality-Reduction/blob/master/techniques/lda.m

[9] P. Sarkar. “What is LDA: Linear Discriminant Analysis for Machine Learning”. Knowledgehut. March 2022. https://www.knowledgehut.com/blog/data-science/linear-discriminant-analysis-for-machine-learning

[10] “Matlab-Toolbox-for-Dimensionality-Reduction”. Github. https://github.com/UMD-ISL/Matlab-Toolbox-for-Dimensionality-Reduction/blob/master/techniques/sammon.m

[11] “Miscellaneous MATLAB Software”. UEA Computational Biology Laboratory. http://theoval.cmp.uea.ac.uk/matlab/default.html

[12] J. W. Sammon, “A Nonlinear Mapping for Data Structure Analysis," IEEE Transactions on Computers, vol. C-18, no. 5, pp:401-409, May 1969

[13] “Matlab-Toolbox-for-Dimensionality-Reduction”. Github. https://github.com/UMD-ISL/Matlab-Toolbox-for-Dimensionality-Reduction/blob/master/techniques/tsne.m

[14] “Matlab-Toolbox-for-Dimensionality-Reduction”. Github. https://github.com/UMD-ISL/Matlab-Toolbox-for-Dimensionality-Reduction/blob/master/techniques/tsne_d.m

[15] “Matlab-Toolbox-for-Dimensionality-Reduction”. Github. https://github.com/UMD-ISL/Matlab-Toolbox-for-Dimensionality-Reduction/blob/master/techniques/tsne_p.m

[16] “Matlab-Toolbox-for-Dimensionality-Reduction”. Github. https://github.com/UMD-ISL/Matlab-Toolbox-for-Dimensionality-Reduction/blob/master/techniques/d2p.m

[17] Z. Zhong, N. Verma. “t-Distributed Stochastic Neighbor Embedding”. Columbia University in the City of New York. June 2018. http://www.cs.columbia.edu/~verma/classes/uml/lec/uml_lec8_tsne.pdf