# Introduction

We’ll be analysing a MNIST dataset containing images of digits and will be trying to classify them into their respective correct buckets of digits. We’ll be implementing QDA for classification, and shall be using PCA for reducing and eventually reconstructing the data.

# Data visualisation

A number in a row

Description automatically generated with medium confidence

5 samples of each type have been displayed here from the dataset provided.

# QDA (Quadratic Discriminant Analysis)

For QDA, we first estimated the parameters of a multivariate normal distribution. Then, we computed the respective discriminant functions which determine the respective classes’ likelihoods. This is done using the training data and now, when testing data is provided, all the above steps are re-done for the given case and then we get the max of the discriminant function value for all classes of the given case, to which it is then assigned.

Upon running QDA on the given dataset, the following observations were recorded :

Overall Accuracy: 0.857

Class 0 Accuracy: 0.9336734693877551

Class 1 Accuracy: 0.6740088105726872

Class 2 Accuracy: 0.935077519379845

Class 3 Accuracy: 0.8772277227722772

Class 4 Accuracy: 0.9083503054989817

Class 5 Accuracy: 0.7993273542600897

Class 6 Accuracy: 0.8903966597077244

Class 7 Accuracy: 0.8618677042801557

Class 8 Accuracy: 0.8870636550308009

Class 9 Accuracy: 0.8235877106045589

# Principal Component Analysis

This is used for reducing and then reconstructing the data while preserving essential data. For this, we first standardize the features grid to a mean of 0 and standard deviation of 1. Then, covariance matrix is calculated of the standardized data, whose eigenvalues and eigenvectors are then calculated. Next, we sort the eigenvectors based on their corresponding eigenvalues. Now, let p be the required dimensionality of the reduced feature space. We select the top p eigenvectors to get the principal components of the data. Now, the original data is projected onto the new feature space.

As for the mean square error, I got the value of

MSE: 1.0478608808473839e-20, which is pretty good.

# Reconstruction with PCA

The following formula was used for reconstruction of data minimized by PCA :

Reconstructed Data = (Up) Y + X¯

Where Up is the matrix of the top p components acquired during PCA

Y is the reduced data matrix (after PCA)

X¯ is the vector of means of the input data

A number in a row

Description automatically generated with medium confidence A number in squares

Description automatically generated with medium confidence

As can be seen, the reconstructed images are far more recognizable when p = 30 than when p = 5, hence showing that as p increases, the quality preservation is greater.

# Class wise accuracies in PCA :

p : 5

accuracies :

Class 0: 0.8908163265306123

Class 1: 0.8572687224669604

Class 2: 0.811046511627907

Class 3: 0.7326732673267327

Class 4: 0.5651731160896131

Class 5: 0.6132286995515696

Class 6: 0.7640918580375783

Class 7: 0.561284046692607

Class 8: 0.3295687885010267

Class 9: 0.6531219028741329

Overall accuracy : 0.6814

p : 30

accuracies :

Class 0: 0.960204081632653

Class 1: 0.48105726872246696

Class 2: 0.9786821705426356

Class 3: 0.904950495049505

Class 4: 0.929735234215886

Class 5: 0.9192825112107623

Class 6: 0.8622129436325678

Class 7: 0.7344357976653697

Class 8: 0.8613963039014374

Class 9: 0.8761149653121902

Overall accuracy : 0.8448

# Conclusion

This project reveals the effectiveness of QDA and PCA in classification and reduction of data dimensions clearly, and hence are proven to be valuable tools for tasks involving pattern recognition. Having made this, it is probable that more advanced libraries such as pyTorch must be utilizing something similar on its end to produce the good results it’s known to give in this field.