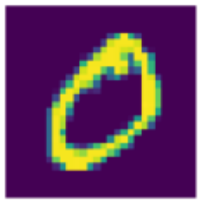
SML ASSIGNMENT - 02 REPORT

Q1:

1st of all, i have imported the dataset. Then the 5 images of each of the different classes present in the dataset.

- The dataset has 10 different classes [0, 9] integral values image dataset.

Similarly for each class Present in dataset, count Of 5 samples each



- After plotting the dataset sample i have vectorized the x_train and x_test to 784-dimensional vector space.

```
Shape of Train matrix:  (60000, 784)
Shape of Test matrix:  (10000, 784)
```

- Calculated the prior probabilities for each class from the train dataset displayed respectively in order.

```
[0.09871666666666666, 0.11236666666666667, 0.0993, 0.10218333333333333, 0.09736666666666667, 0.09035, 0.09863333333333334, 0.10441666666666667, 0.09751666666666667, 0.09915]
```

- Made a function named qda to calculate the discriminant without using any inbuilt library.
- Determined the covariance, inverse covariance, determinant of covariance, and mean for every class from the training dataset. X test sample.

  Qda = $-(x-\mu)^T\Sigma^{-1}(x-\mu) * 0.5 + log(prior\ probab.) - 0.5 * log(det(\Sigma))$

  The prediction goes to class having max for the above expression
- Using all of the above resources determine the max discriminant for every class that belongs to which class and stored in a prediction array.
- Using the prediction array storing the class values for every element in the test dataset.
- Then matched that with every element in the in y_tests_label data.
- Using that determined the matched and unmatched cases and calculated the accuracy for every class and the overall accuracy. Achieved an overall accuracy of 85.73 %. The classwise accuracy displayed below.

```
Accuracy for each class and overall upto 2 decimal points
Class 0 Accuracy:  93.37 %
Class 1 Accuracy:  67.49 %
Class 2 Accuracy:  93.51 %
Class 3 Accuracy:  87.82 %
Class 4 Accuracy:  90.84 %
Class 5 Accuracy:  79.93 %
Class 6 Accuracy:  89.04 %
Class 7 Accuracy:  86.19 %
Class 8 Accuracy:  88.71 %
Class 9 Accuracy:  82.46 %
Overall Accuracy:  85.73 %
```

Q2

- Taken 100 samples from each class forming an array of 784x1000 dimensions. Named as X and removed mean(X) from it making it centralized.

```
(60000, 784)
```

- Applying PCA now on my centeralized X, for that computing covariance $S = (XX^T)/999$ and then computing its eigen vectors and eigen values for that taking the help of NUMPY python library to compute respectively and stored in U in sorted order.

- Now reconstructing my X for that using $Y = U^T X$ to reconstruct Xre = UY. now determining the MSE between Xre and X comes out to be **MSE: 1.86e-26**
- For each p belongs [5, 10, 20] reconstructed the images of 5 set for every class.
- For each p computed the UpY and added the Mean back and plotted for each p belongs to [5, 10, 20]

For **P = 20** the 1 set of plot has been displayed in the left.

Couldn be able to done the last part of this question due to less time and some error in computation.