Project Report

In this part of the project, we are initially given 4 files namely training_data_0, training_data_1, testing_data_0, testing_data_1.mat files from MNIST dataset which represents the training and testing images for images 0 and 1. Each image is initially stored as 28 x 28 vector.

In the initial preprocessing of the files, I “vectorized” an image by concatenating its columns to form a 784-dimensional vector. For instance, the training_data_0 is now represented as 5923 x784 dimensional vector.

Task2: Calculating the PCA

To compute PCA we need to compute covariance matrix and use this covariance matrix to compute eigen values and eigen vectors. Covariance matrix is calculated using the numpy.cov. Eigen values and Eigen vectors are computed using numpy.linalg.eig function.

Eigen Vectors: 784 x 784 dimension

```
[[6.71636171e-04, 2.18256179e-03, 5.33374668e-04, ...,
     -3.80323705e-04, -4.39842096e-04, -8.56599803e-05],
    [ 5.47816983e-04, -1.50132206e-03, -3.88916647e-03, ...,
      6.83721472e-04,  1.08181840e-03, -6.36200779e-04],
    [ 6.55489835e-04,  9.36205584e-04, -4.37032530e-04, ...,
     -6.41272463e-04, -2.04554669e-03, -2.94319008e-04],
```

Eigen Values: array of size 784

```
Array ([9.94608219e+01, 3.98030418e+01, 2.62106449e+01, 2.33640930e+01,
    1.61768688e+01, 1.23765458e+01, 1.05256521e+01, 9.28483830e+00,.. ])
```

Cumulative variance:

```
[12.68632933, 17.76324793 ,21.10644243 ,24.08655633 ,26.14993245
  27.72857349 ,29.07113116 ,30.25542176 31.34857034, 32.28239869
….. ]
```

This shows that If we use the first feature, it will explain 12.68% of the data; if I use two features, we can capture 17.76 of the data.

After choosing the first and second principal component from eigen vectors, I computed the PCA transformation on the original dataset, getting the dot product of the original standardized dataset and the eigenvectors that I got from the eigen decomposition.

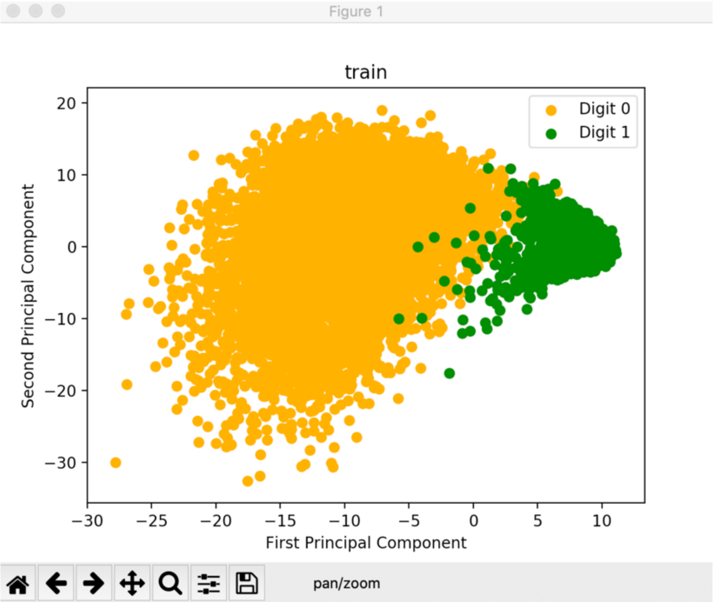
PCA for the training set:

```
          0                    1
 [-8.28750692 ,6.03794498]
```

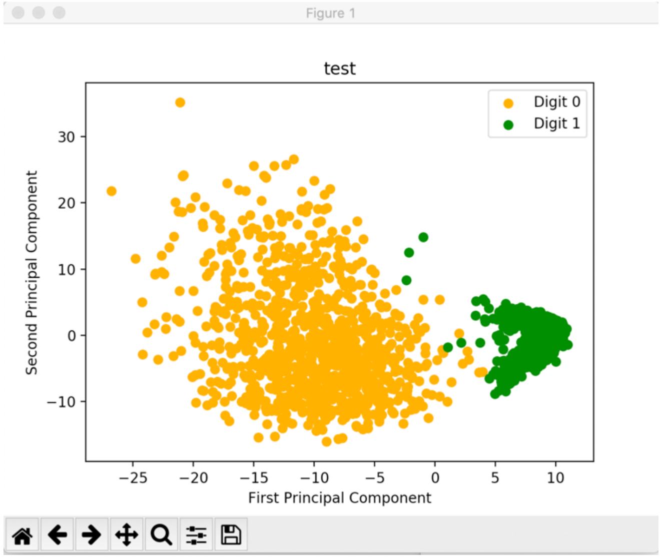
```
[-9.14412938 ,6.15518197]
[-3.58386457,11.40611319] ….
```

Task3: Dimension reduction using PCA

In this step, I plotted both the first and second principal components for digit 0 and digit 1. The following is the plot for training set:



The following is the plot for testing set



Task4: Density Estimation

Computing mean and variance for training data:

For Diigit0:
Mean: [[-9.92384569],
[0.8514571]
Covariance:
[[25.32460567, 15.90014464],
[15.90014464, 79.11311962]]

For Digit1:
Mean: [[8.71832365],
[-0.74802439]]
Covariance:
[[2.06676736, -0.02148539],
[-0.02148539,4.0841354]]

Task 5: Bayesian Decision Theory for optimal classification

Computing the the minimum error rate is calculated using Bayesian decision boundary rule

training data accuracy= 98.87879984208449 %
test data accuracy= 99.43262411347517 %