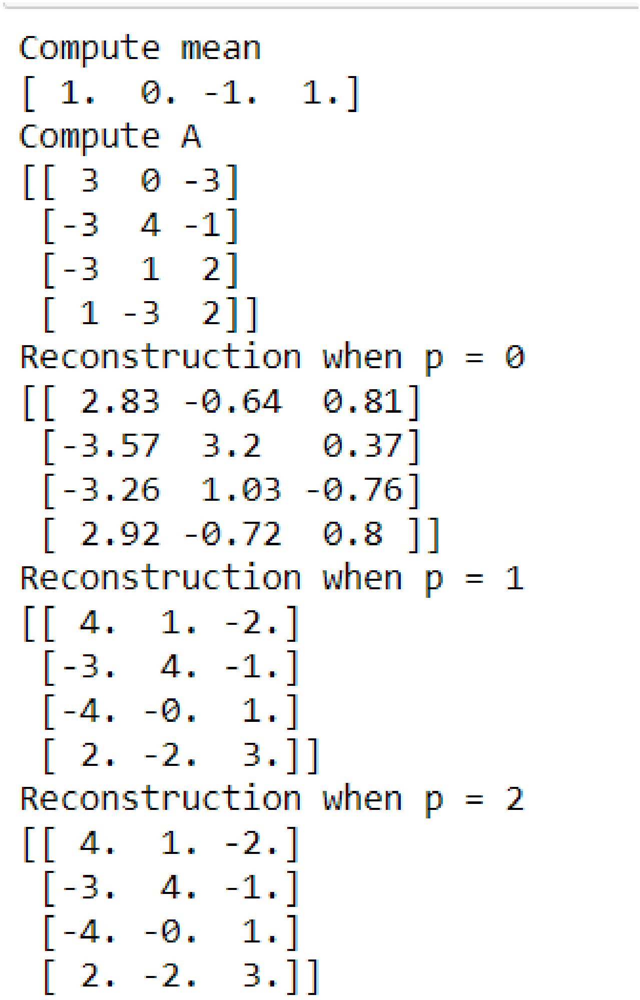
ECE586 MPX Principal Component Analysis (PCA)

Libo Zhang (NetID: lz200)

Exercise 1. (15 pts)

For each p = [0, 1, 2], print the resulting post-PCA data matrix with mean added back in and rounded to 2 decimal digits. The results are shown below.



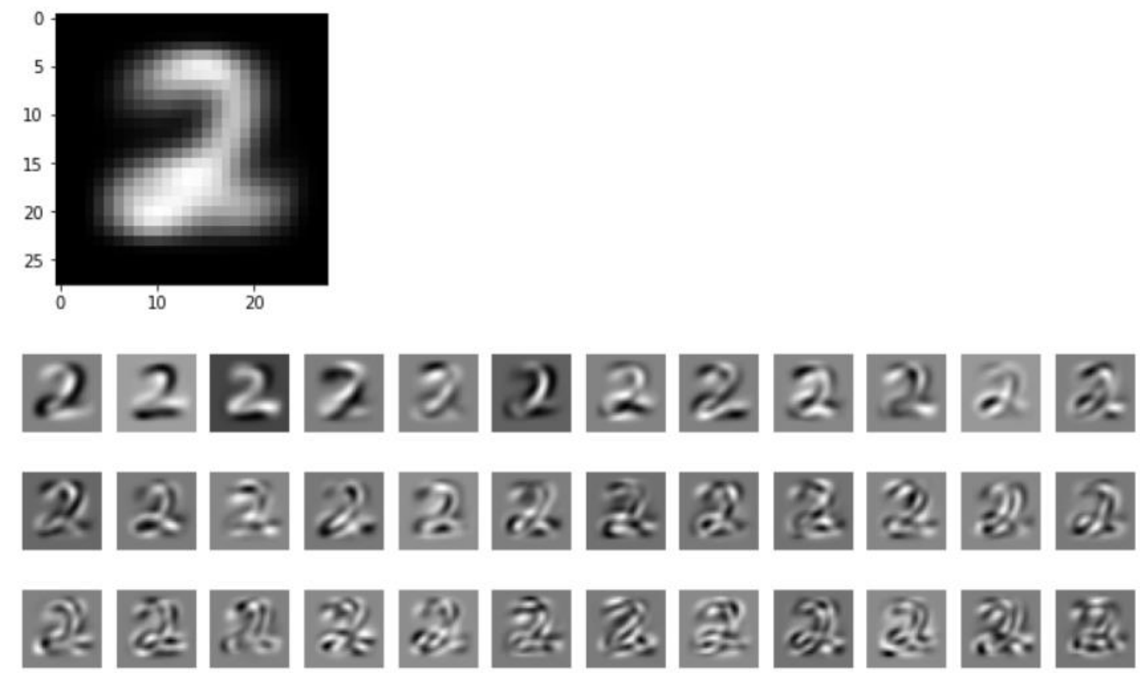
```
Compute mean
[ 1.  0. -1.  1.]
Compute A
[[ 3  0 -3]
 [-3  4 -1]
 [-3  1  2]
 [ 1 -3  2]]
Reconstruction when p = 0
[[ 2.83 -0.64  0.81]
 [-3.57  3.2   0.37]
 [-3.26  1.03 -0.76]
 [ 2.92 -0.72  0.8 ]]
Reconstruction when p = 1
[[ 4.  1. -2.]
 [-3.  4. -1.]
 [-4. -0.  1.]
 [ 2. -2.  3.]]
Reconstruction when p = 2
[[ 4.  1. -2.]
 [-3.  4. -1.]
 [-4. -0.  1.]
 [ 2. -2.  3.]]
```

Note: I feel a bit **confused** because in the **MPX handout Exercise 1**, we **are only** asked to report the post-PCA data matrix. In **MPX Python notebook template**, however, we are asked to report both the post-PCA data matrix and the error in the Frobenius norm. Since I have showed the above results to the professor during Tuesday's Office Hour and the professor confirmed the correctness of my results, I will follow the requirements in the pdf handout and choose not to report the error in the Frobenius norm. Thanks for your understanding.

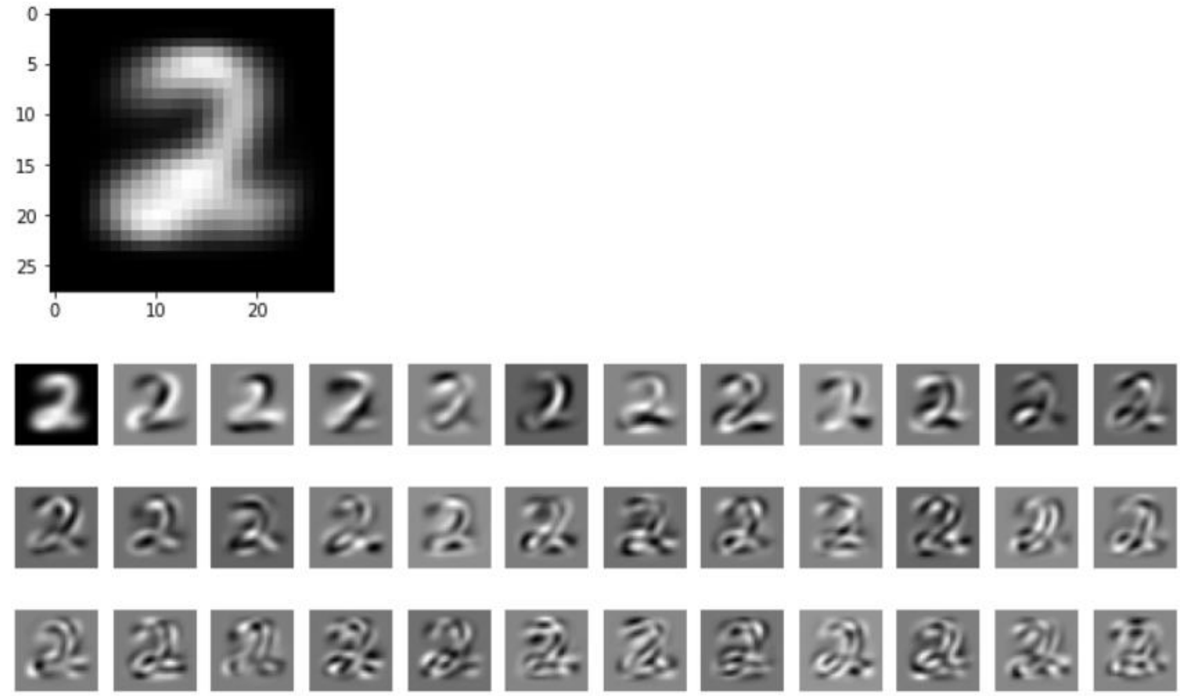
Please continue to the next page for Exercise 2.

Exercise 2. (25 pts)

Plot with mean removal. (Note: The biggest number “2” first showing up is the mean.)



Plot without mean removal. (Note: The biggest number “2” first showing up is the mean.)

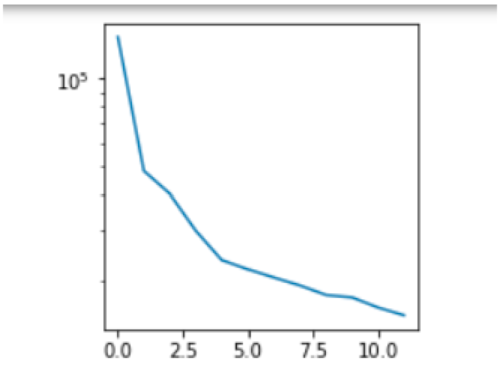


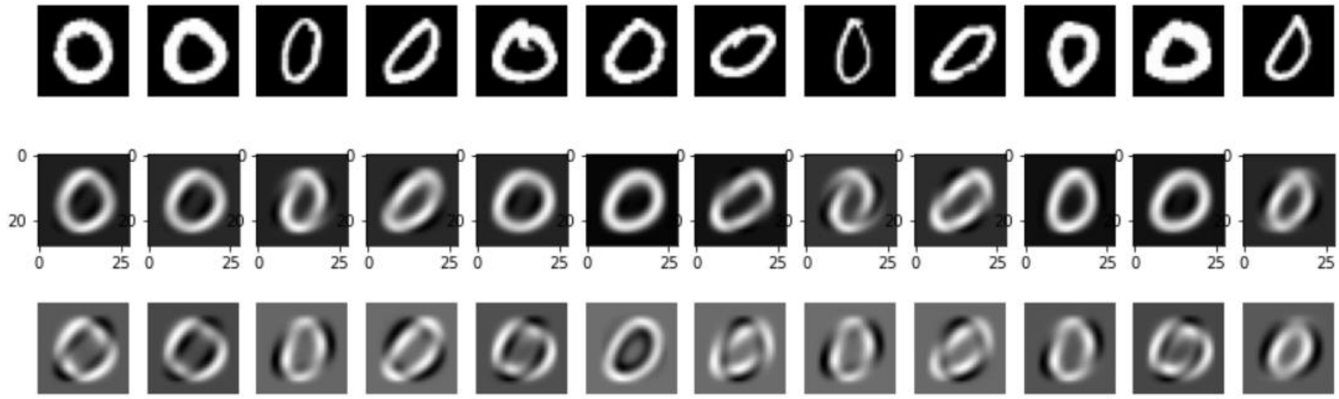
<u>What do you observe about these images? How many of these look a lot like digit 2?</u>

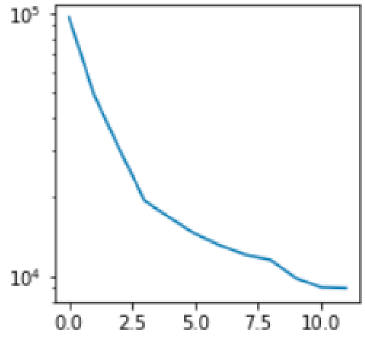
These greyscale images, compared with the original digit 2 images, seem to distort, or mingle the boundaries between the actual white color digit and the black color background, and have comprehensive grey color area as the background. As for the plots with mean removal, I think the first 24 images look a lot like digit 2. As for the plots without mean removal, I think the first image looks extremely like digit 2. However, all in all, only the first 18 images look a lot like digit 2 and the rest images are heavily distorted. It seems that with mean removal, although the first image may not look “extremely” like digit 2, the rest images are more “average” like digit 2 than images without mean removal.

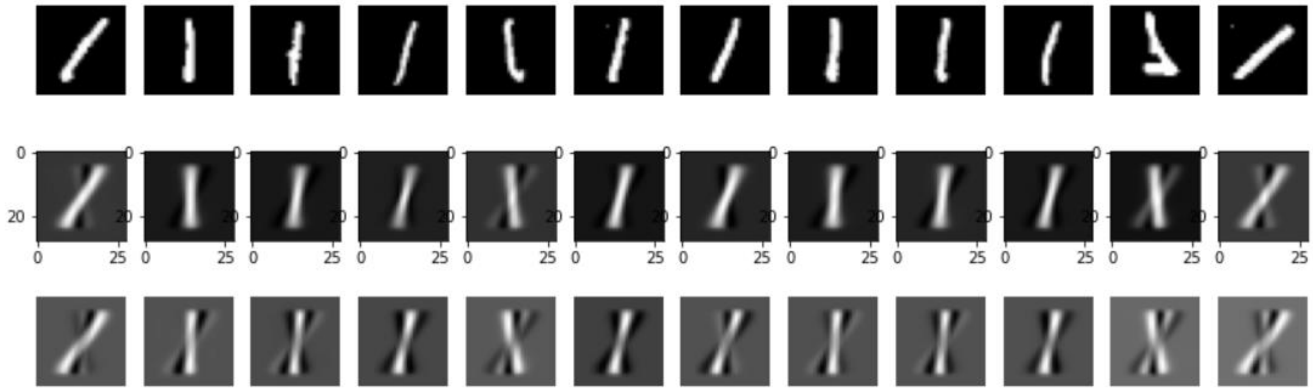
Exercise 3. (30 pts)

The results are shown below.

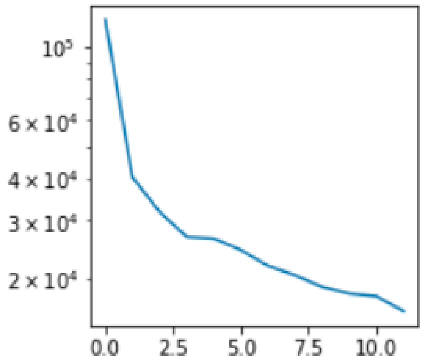


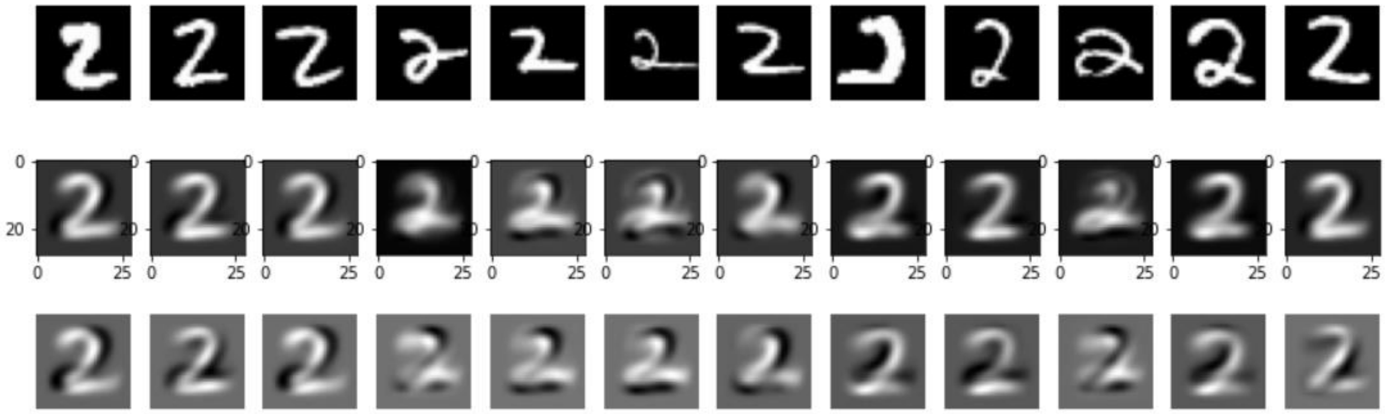


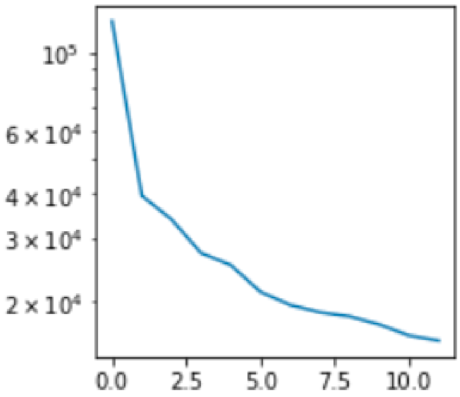


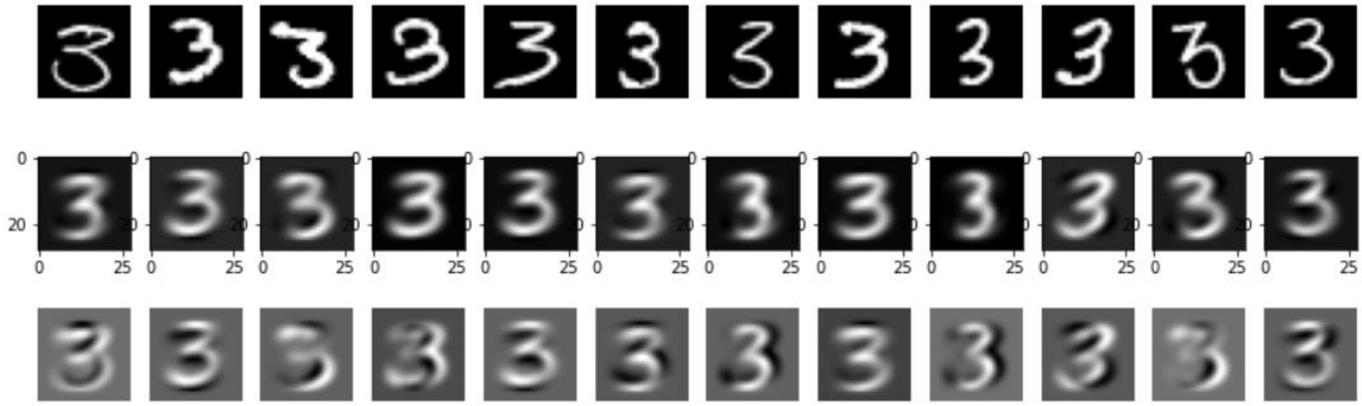


Please continue to the next page for more results.

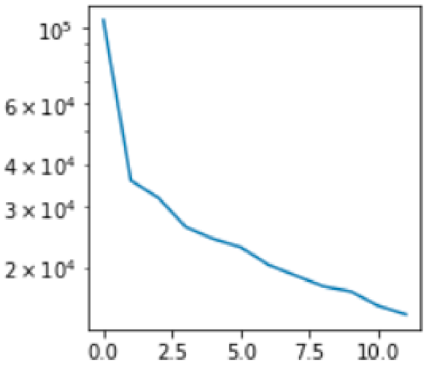


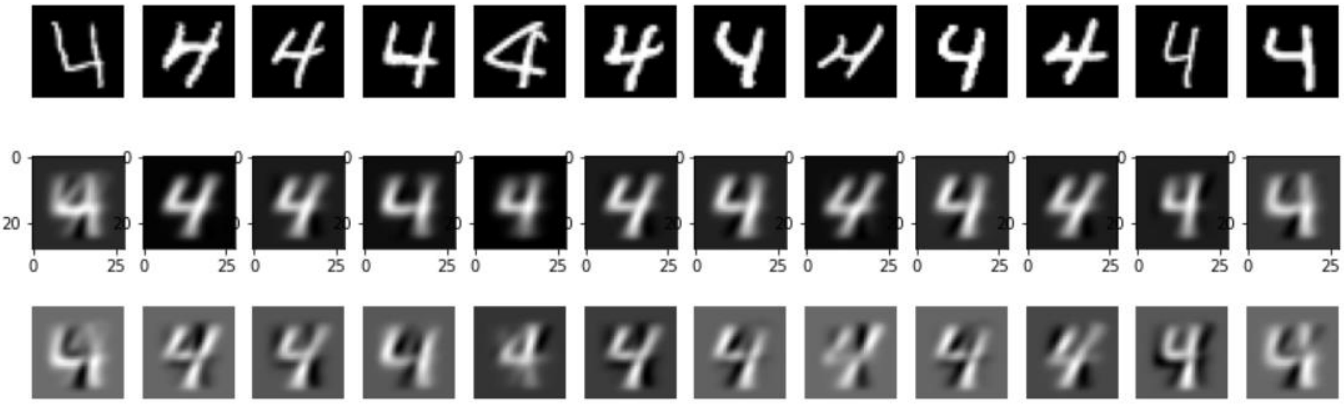


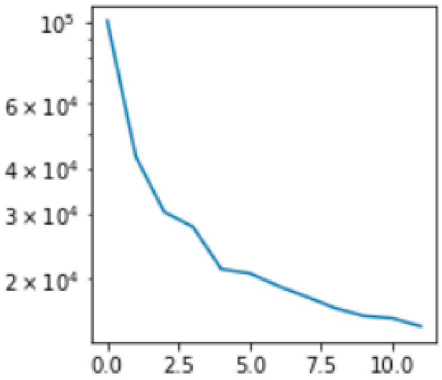


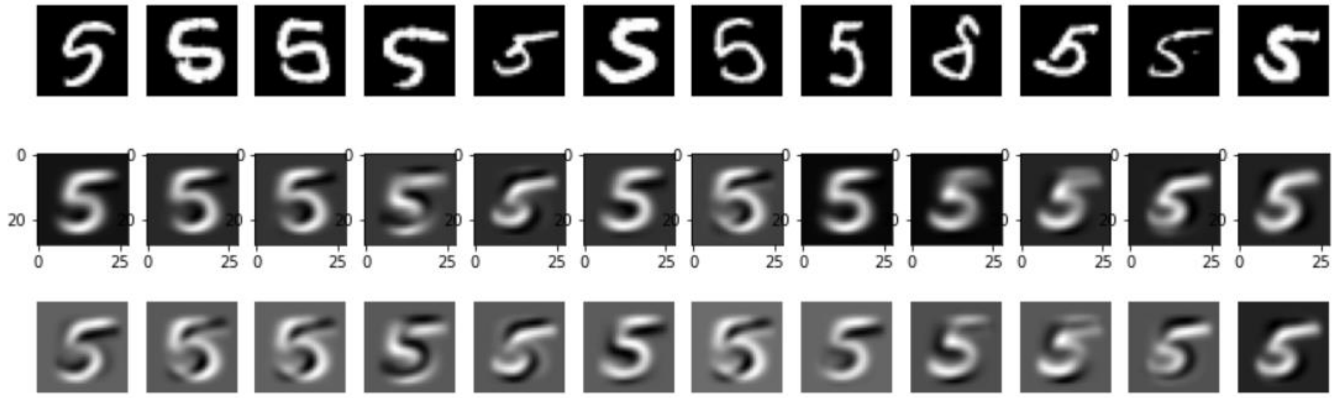


Please continue to the next page for more results.

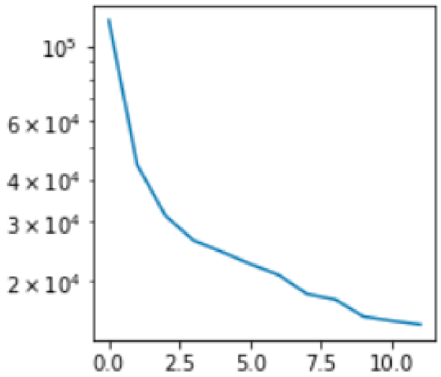


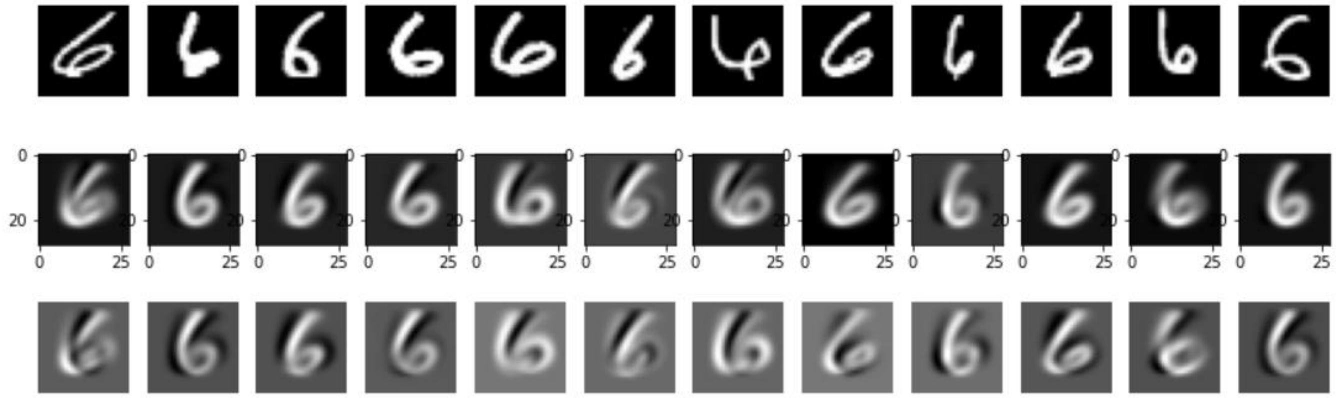


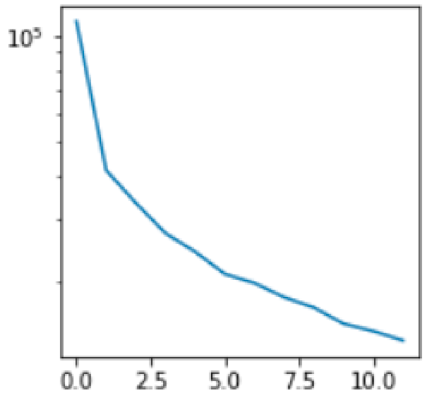


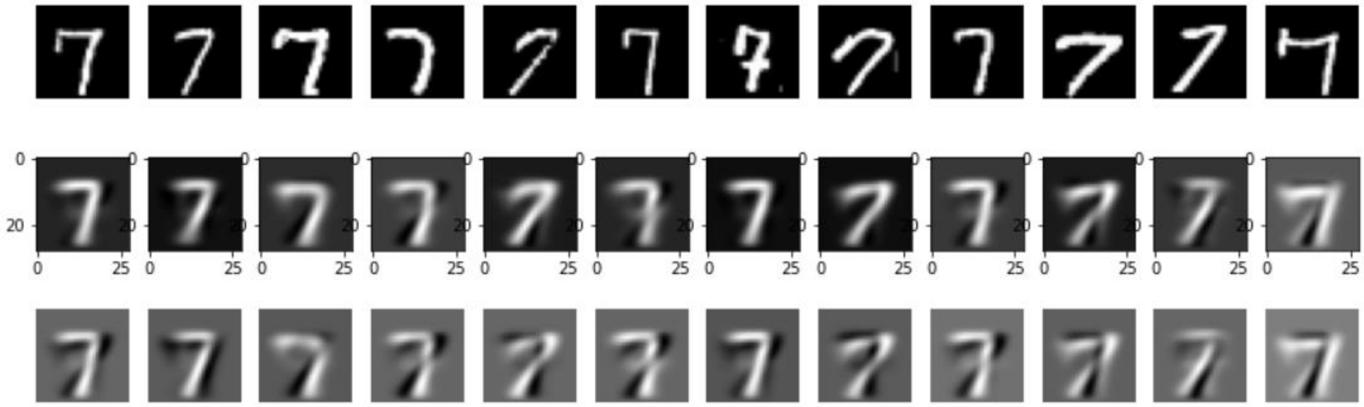


Please continue to the next page for more results.

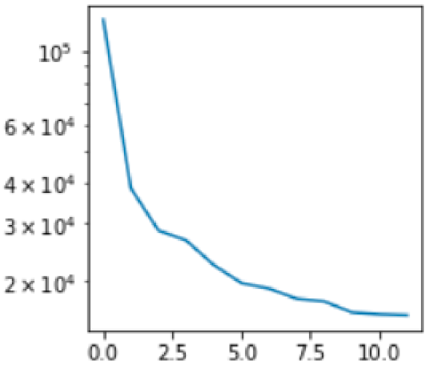


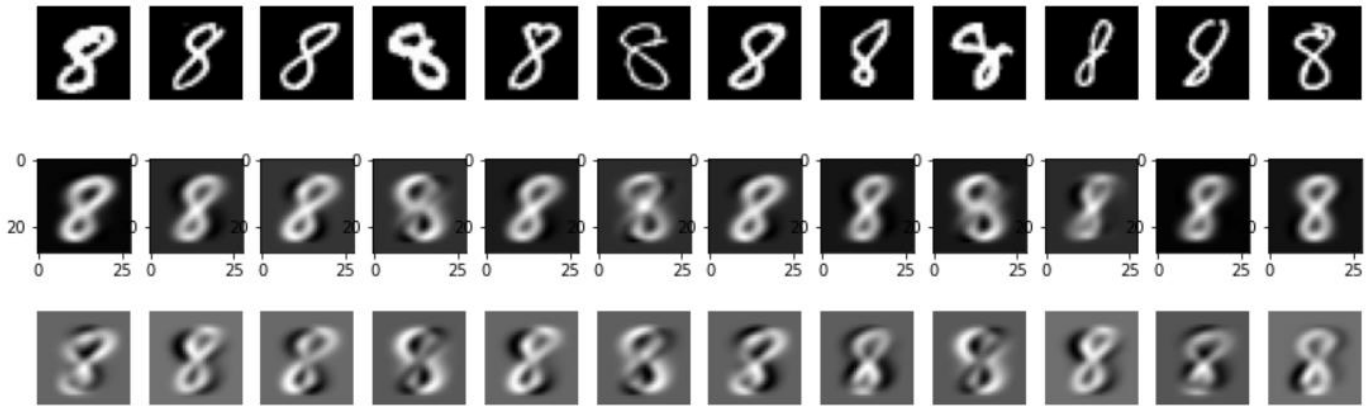


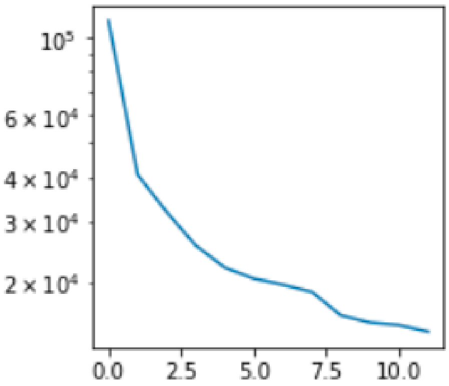


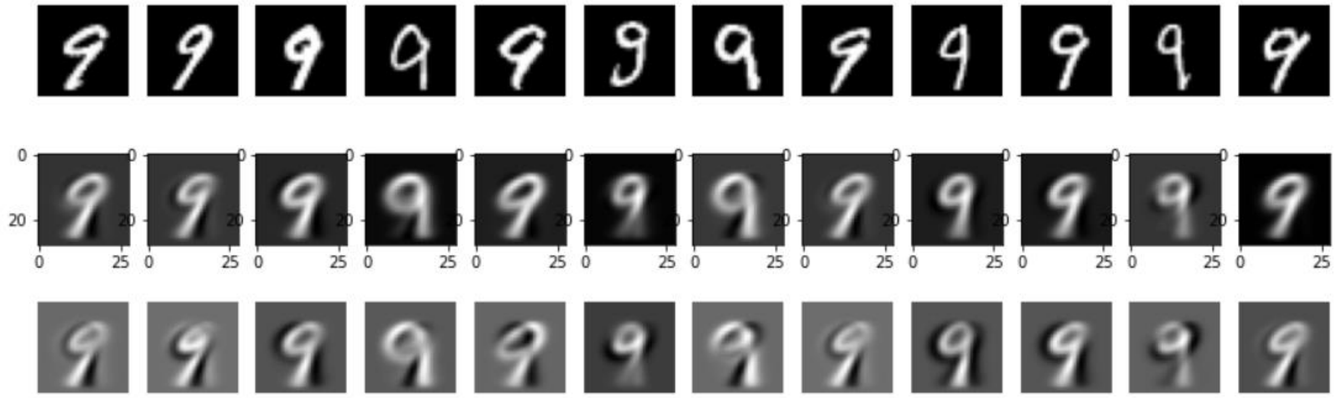


Please continue to the next page for more results.







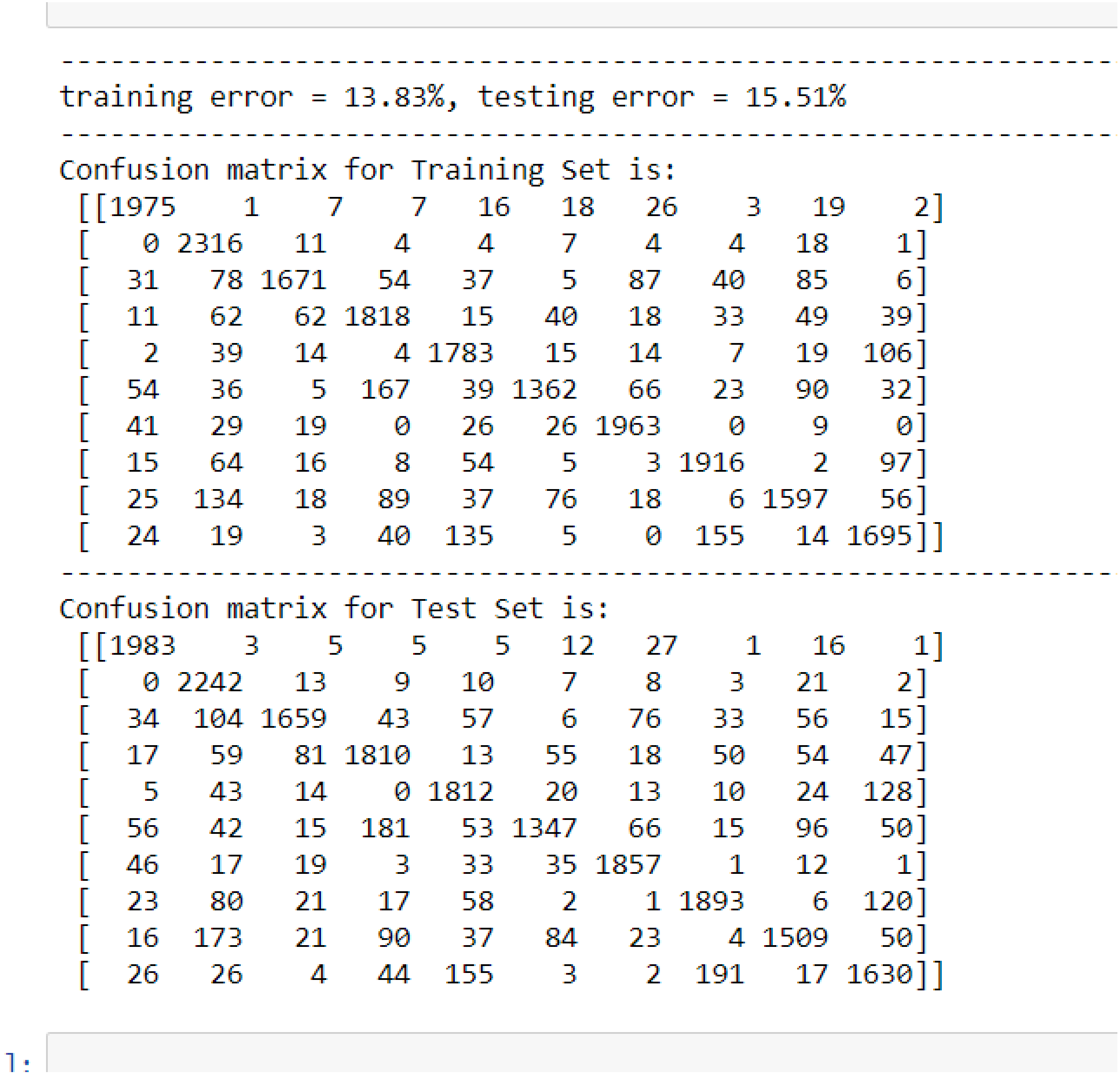


Please continue to the next page for Exercise 4.

Exercise 4. (30 pts)

First, I show the results after using PCA to reduce dimension from 28 * 28 = 784 dimensions to 500 dimensions.

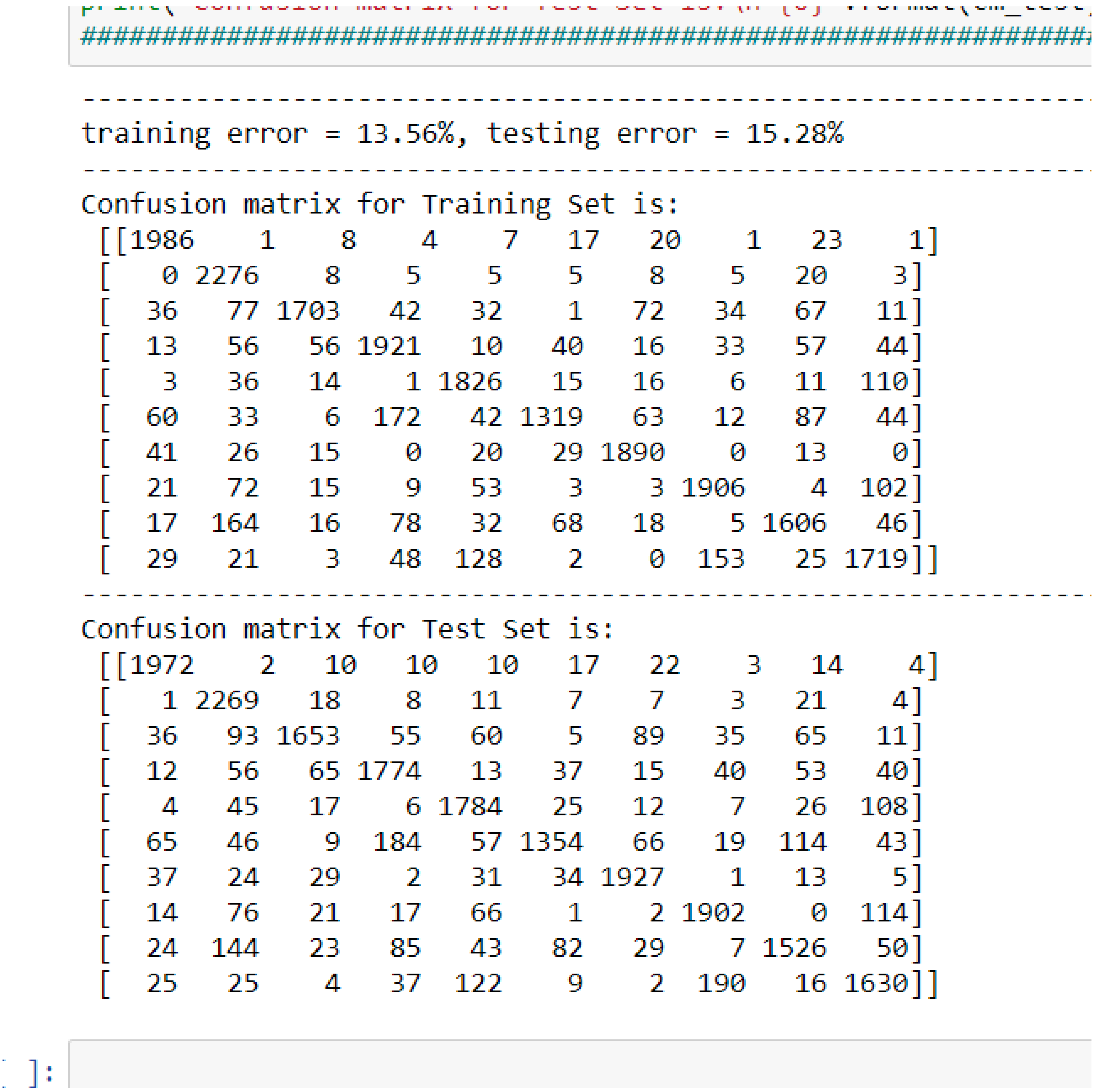
The results are shown below.



```
---------------------------------------------------------------
training error = 13.83%, testing error = 15.51%
---------------------------------------------------------------
Confusion matrix for Training Set is:
 [[1975    1    7    7   16   18   26    3   19    2]
 [   0 2316   11    4    4    7    4    4   18    1]
 [  31   78 1671   54   37    5   87   40   85    6]
 [  11   62   62 1818   15   40   18   33   49   39]
 [   2   39   14    4 1783   15   14    7   19  106]
 [  54   36    5  167   39 1362   66   23   90   32]
 [  41   29   19    0   26   26 1963    0    9    0]
 [  15   64   16    8   54    5    3 1916    2   97]
 [  25  134   18   89   37   76   18    6 1597   56]
 [  24   19    3   40  135    5    0  155   14 1695]]
---------------------------------------------------------------
Confusion matrix for Test Set is:
 [[1983    3    5    5    5   12   27    1   16    1]
 [   0 2242   13    9   10    7    8    3   21    2]
 [  34  104 1659   43   57    6   76   33   56   15]
 [  17   59   81 1810   13   55   18   50   54   47]
 [   5   43   14    0 1812   20   13   10   24  128]
 [  56   42   15  181   53 1347   66   15   96   50]
 [  46   17   19    3   33   35 1857    1   12    1]
 [  23   80   21   17   58    2    1 1893    6  120]
 [  16  173   21   90   37   84   23    4 1509   50]
 [  26   26    4   44  155    3    2  191   17 1630]]
```

Please continue to the next page for Exercise 4.

Second, I show the original results acquired from the Least Squares project. The original results are shown below.



```
##########################################################################

----------------------------------------------------------------------
training error = 13.56%, testing error = 15.28%
----------------------------------------------------------------------
Confusion matrix for Training Set is:
 [[1986    1    8    4    7   17   20    1   23    1]
 [   0 2276    8    5    5    5    8    5   20    3]
 [  36   77 1703   42   32    1   72   34   67   11]
 [  13   56   56 1921   10   40   16   33   57   44]
 [   3   36   14    1 1826   15   16    6   11  110]
 [  60   33    6  172   42 1319   63   12   87   44]
 [  41   26   15    0   20   29 1890    0   13    0]
 [  21   72   15    9   53    3    3 1906    4  102]
 [  17  164   16   78   32   68   18    5 1606   46]
 [  29   21    3   48  128    2    0  153   25 1719]]
----------------------------------------------------------------------
Confusion matrix for Test Set is:
 [[1972    2   10   10   10   17   22    3   14    4]
 [   1 2269   18    8   11    7    7    3   21    4]
 [  36   93 1653   55   60    5   89   35   65   11]
 [  12   56   65 1774   13   37   15   40   53   40]
 [   4   45   17    6 1784   25   12    7   26  108]
 [  65   46    9  184   57 1354   66   19  114   43]
 [  37   24   29    2   31   34 1927    1   13    5]
 [  14   76   21   17   66    1    2 1902    0  114]
 [  24  144   23   85   43   82   29    7 1526   50]
 [  25   25    4   37  122    9    2  190   16 1630]]

[ ]:
```

<u>How does the error rate compare to the results from the Least Squares project?</u>

From the above two figures, we know the PCA results have a training error of 13.83%, which is a little bit higher but very similar to the Least Squares project training error 13.56%. Similarly, the PCA results have a testing error of 15.51%, which is only a little bit higher but very similar to the Least Squares project testing error 15.28%.

To conclude, after reducing dimensions from 784 to 500 with PCA, we can still achieve very good or very competitive training and testing accuracies compared with the original Least Squares results.