SML ASSIGNMENT 3 REPORT

Ques1.

A. Implemented PCA and LDA from scratch.

B. The classifier used in Gaussian Naive Bayes. Accuracy on test set is 66.8% on the face dataset.

```
clf= GaussianNB()
clf= clf.fit(X_train, Y_train)
pred= clf.predict(X_test)
print ("The accuracy on test set as it is:- " + str((pred== Y_test).sum()/len(pred)))
```

```
The accuracy on test set as it is:- 0.6682464454976303
```

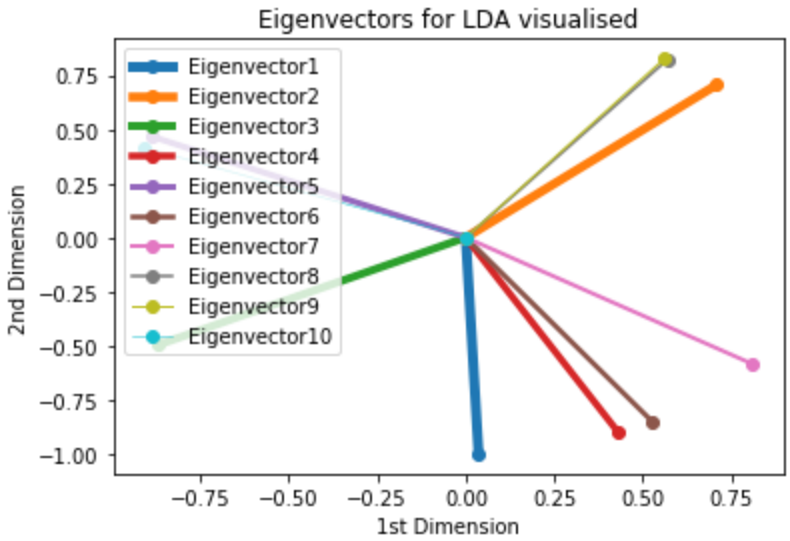
The same classifier when used on CIFAR 10 dataset gives 29.76% accuracy only.

```
clf= GaussianNB()
clf= clf.fit(X_train, Y_train)
pred= clf.predict(X_test)
print ("The accuracy on test set as it is:- " + str((pred== Y_test).sum()/len(pred)))
```

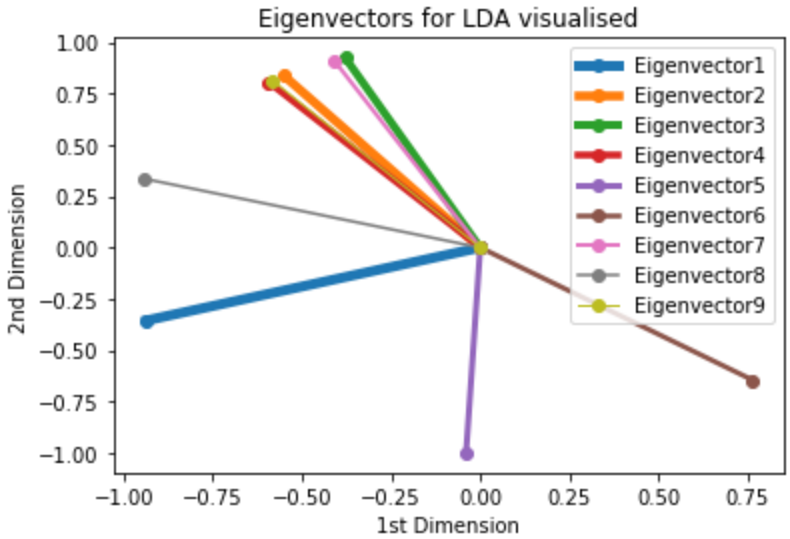
```
The accuracy on test set as it is:- 0.2976
```

C. The projections directions for LDA were plotted as follows:-

1. Face dataset



2. Cifar-10 dataset



D. Project the data onto the projection matrix obtained from lda function giving projected data.

1) Face dataset

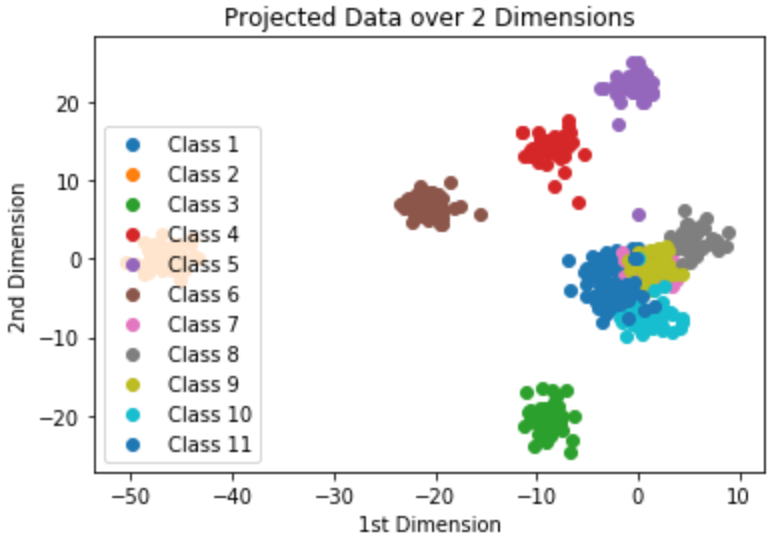
```
X_trainlda= []
X_testlda= []
for i in range(len(X_train)):
    X_trainlda.append(np.dot(X_train[i], Wlda))
for i in range(len(X_test)):
    X_testlda.append(np.dot(X_test[i], Wlda))

print(len(X_trainlda), len(X_trainlda[0]))
gnb= GaussianNB()
Y_pred2= gnb.fit(X_trainlda, Y_train).predict(X_testlda)
#print (Y_pred2)
```

```
493 10
```

```
print ((Y_pred2== Y_test).sum()/len(Y_test))
```

```
0.8909952606635071
```



2) CIFAR 10 dataset

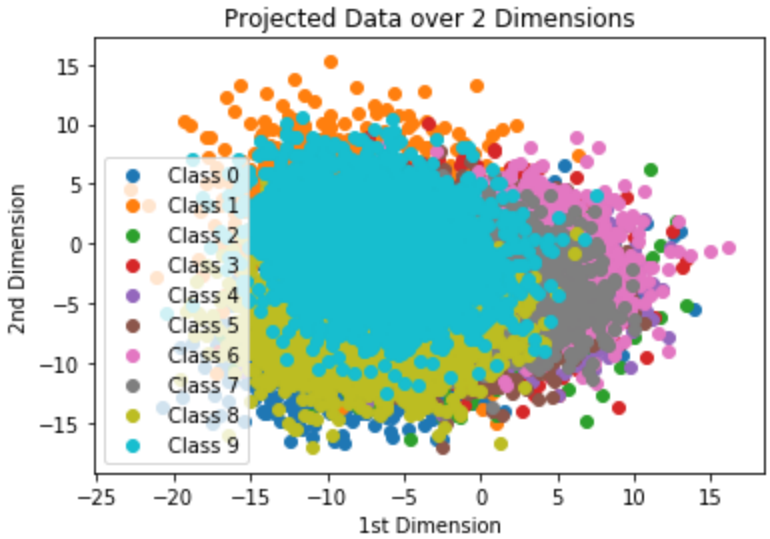
```
X_trainlda= []
X_testlda= []
for i in range(len(X_train)):
    X_trainlda.append(np.dot(X_train[i], Wlda))
for i in range(len(X_test)):
    X_testlda.append(np.dot(X_test[i], Wlda))

print(len(X_trainlda), len(X_trainlda[0]))
gnb= GaussianNB()
Y_pred2= gnb.fit(X_trainlda, Y_train).predict(X_testlda)
#print (Y_pred2)
```

```
50000 9
```

```
print ((Y_pred2== Y_test).sum()/len(Y_test))
```

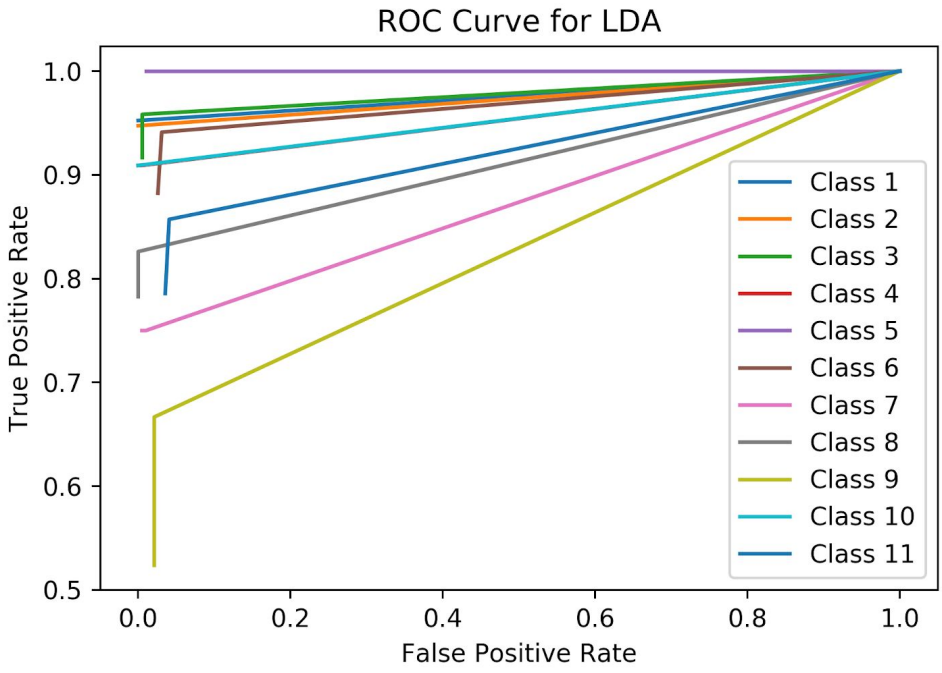
```
0.3659
```

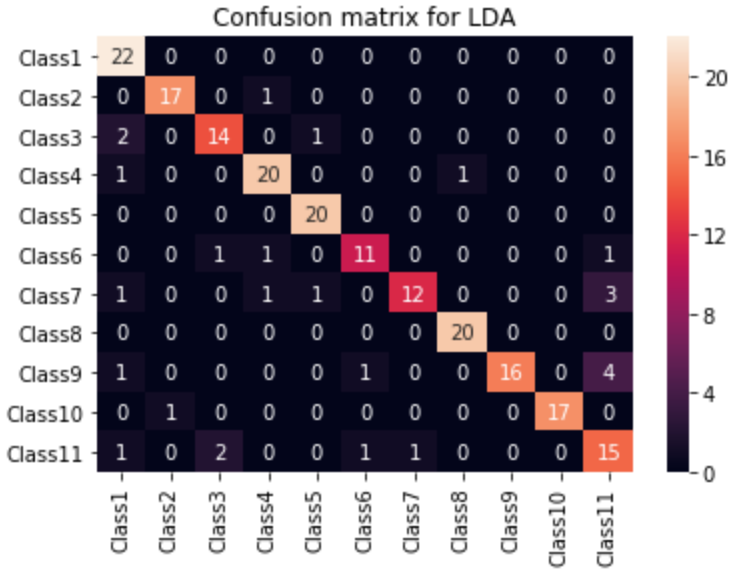


E.

1) Face dataset

The mean accuracy of LDA on the validation set during 5-fold cross=validation is 98.98% and the standard deviation is 0.6%.
The best model during C.V. gives an accuracy of 88.6% on the test set when projected data is reduced to 10 dimensions.

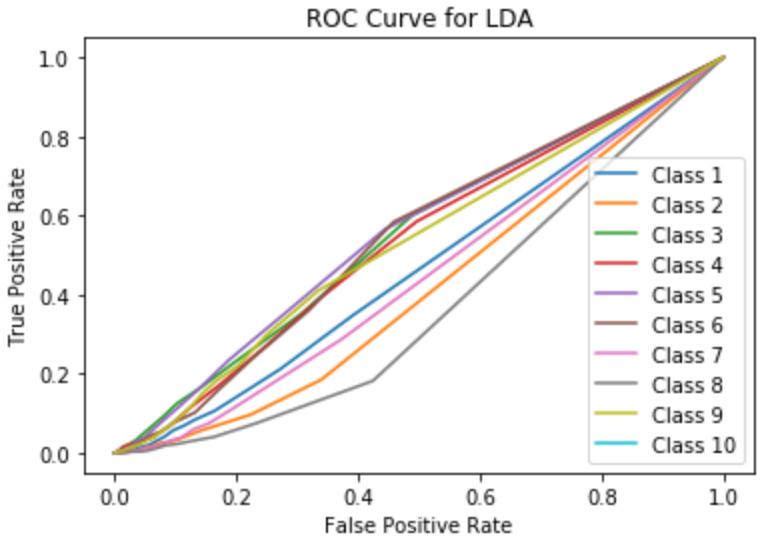


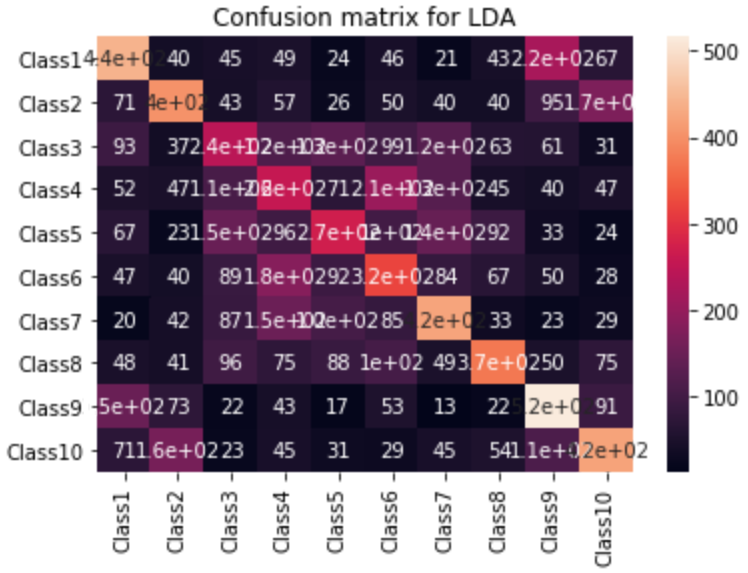


2) CIFAR 10 dataset

The mean accuracy of LDA on the validation set during 5-fold cross=validation is 50.93% and the standard deviation is 0.45%.

The best model during C.V. gives an accuracy of 36.76% on the test set when projected data is reduced to 9 dimensions.

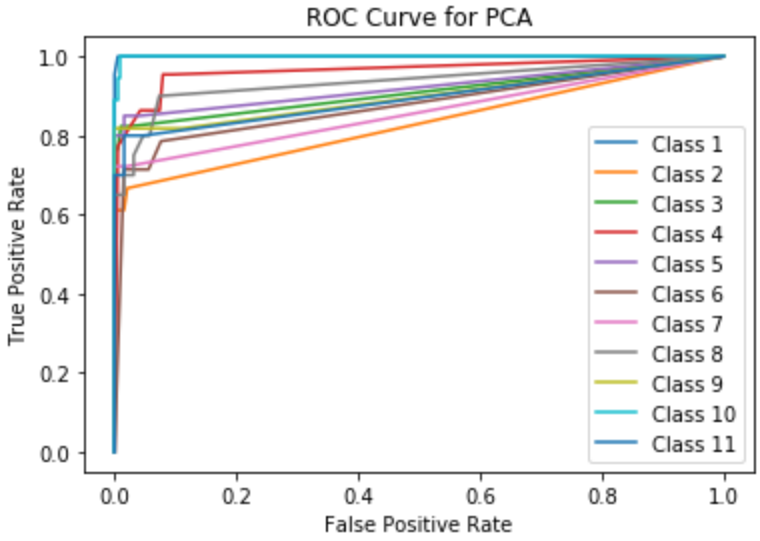


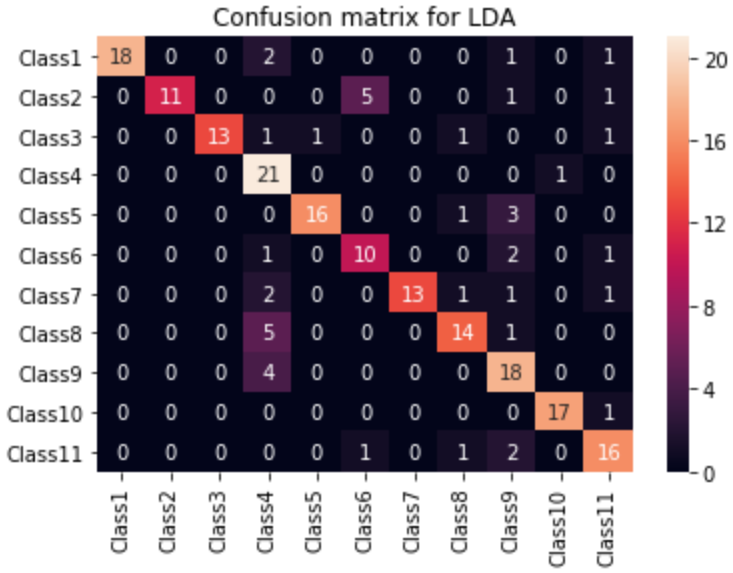


F.

1) Face dataset

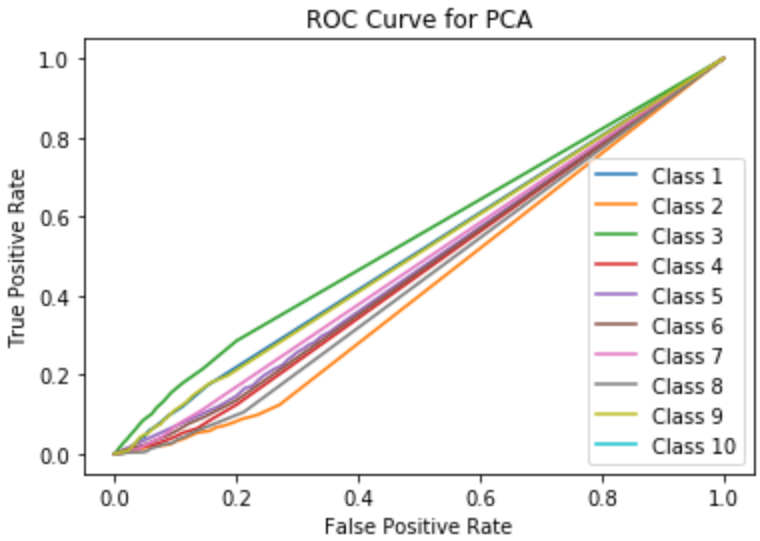
The accuracy on the test set for the best model during C.V. in the case of PCA with 0.95 eigenenergy is 81.99%. The mean accuracy on the validation set across 5-folds is 82.3% and the standard deviation is 2.7%.





2) CIFAR 10 dataset

The accuracy on the test set for the best model during C.V. in the case of PCA with 0.95 eigenenergy is 31.31%. The mean accuracy on the validation set across 5-folds is 30.996% and the standard deviation is 0.41%.



The accuracy for PCA with 0.95 eigenenergy is much lower in case of the Cifar-10 dataset compared to the face dataset. **This is since the Cifar-10 dataset is extremely varied.** In general, even human annotators don't get more than 95% accuracy on the dataset. The classes are although only 10, but **have large variations among classes.** This makes it difficult to reduce the dimensionality of the data by projecting along the maximum variance.

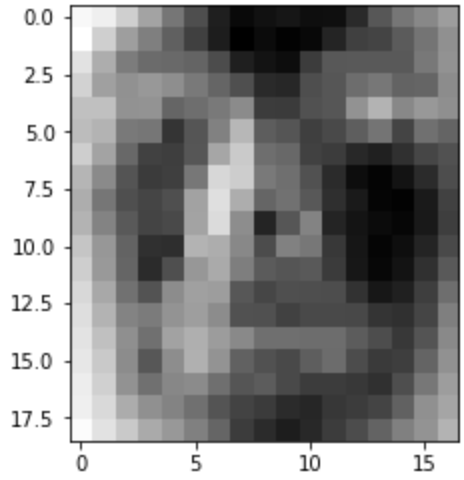
G.

It is clearly observed that accuracy in case of PCA is generally lower than LDA. This is since PCA is a dimensionality reduction technique that projects data onto the lower dimensions along the maximum variance. On the other hand, LDA is a dimensionality reduction as well as a clustering technique that increases the separability of the data thus giving better accuracies while classification in general.

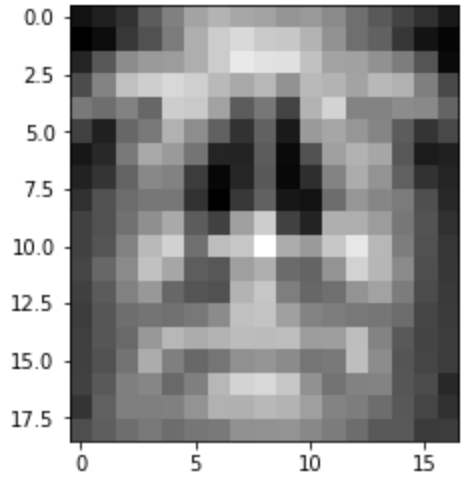
h.

Face dataset

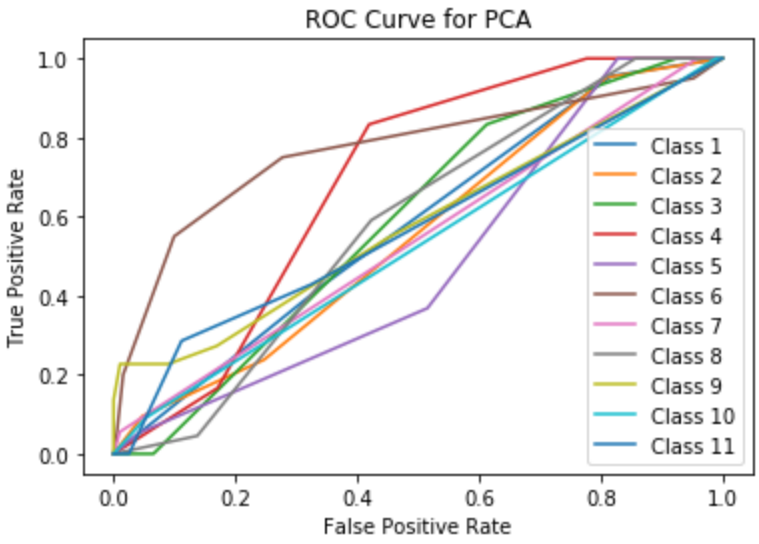
Eigenface 1:-

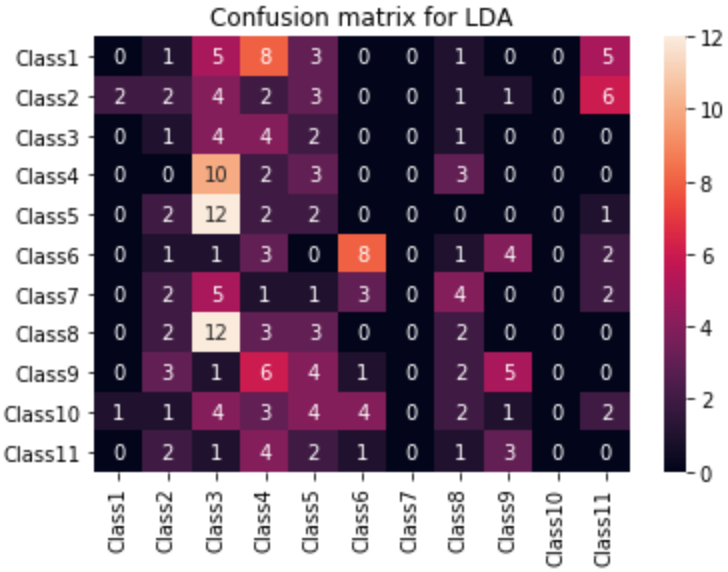


Eigenface 2:-



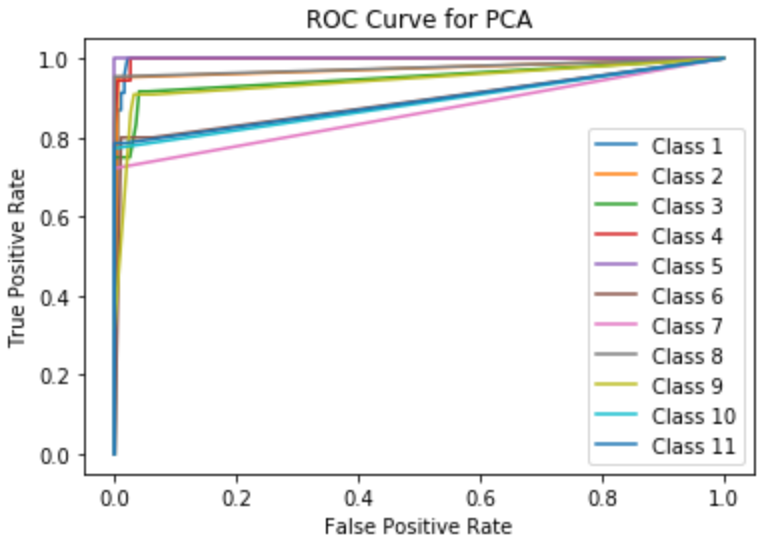
2) For 70% eigenenergy(top 2 eigenvectors): 10.4% accuracy

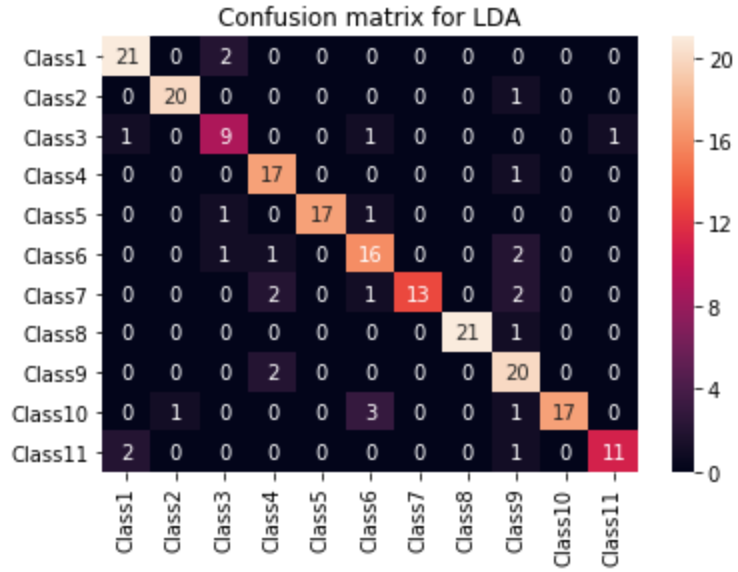




For 90% eigenenergy(top 12 eigenvectors): 72.5% accuracy

For 99% eigenenergy(top 92 eigenvectors): 86.255% accuracy

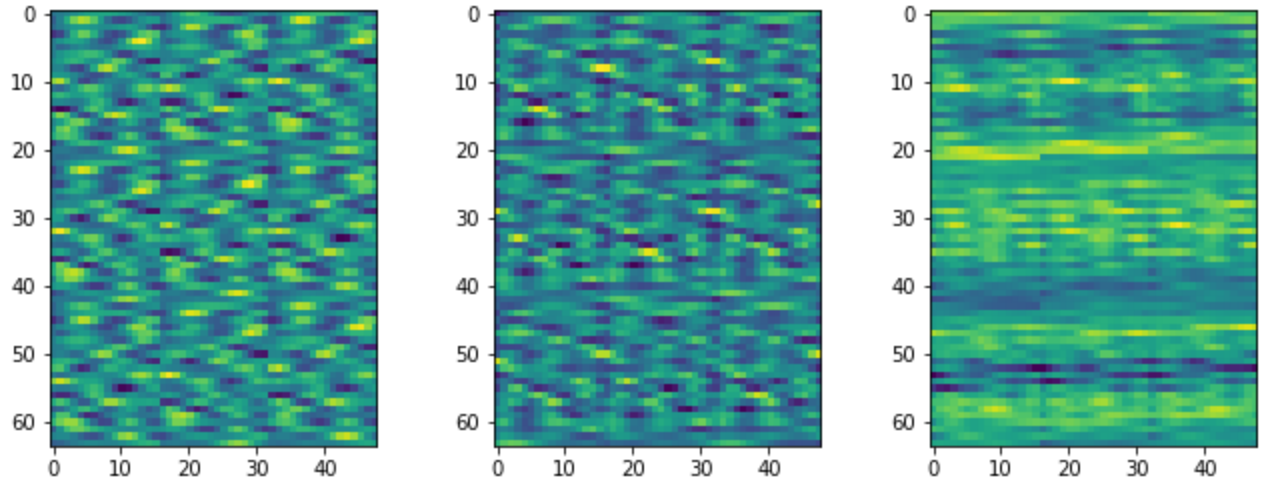




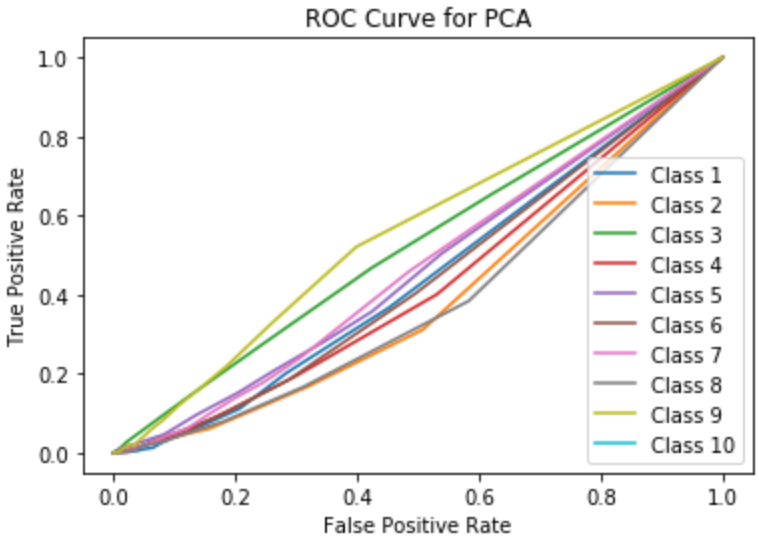
As it is clearly evident, there is a drastic rise in accuracy while just going from 70% to 90% eigenenergy and as the eigenenergy keeps on increasing, this rate of change in the accuracy slowly saturates.

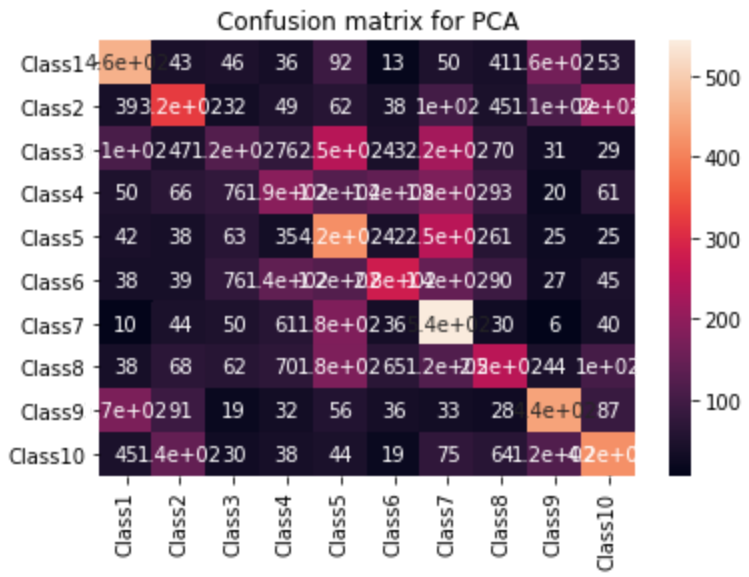
Cifar-10 dataset

1) Eigenfaces:-



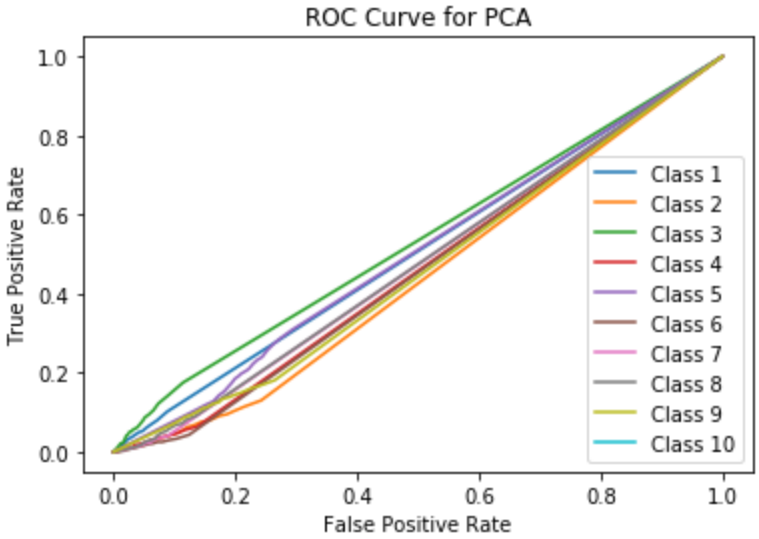
2) For 70% eigenenergy(top 15 eigenvectors): 34.55% accuracy

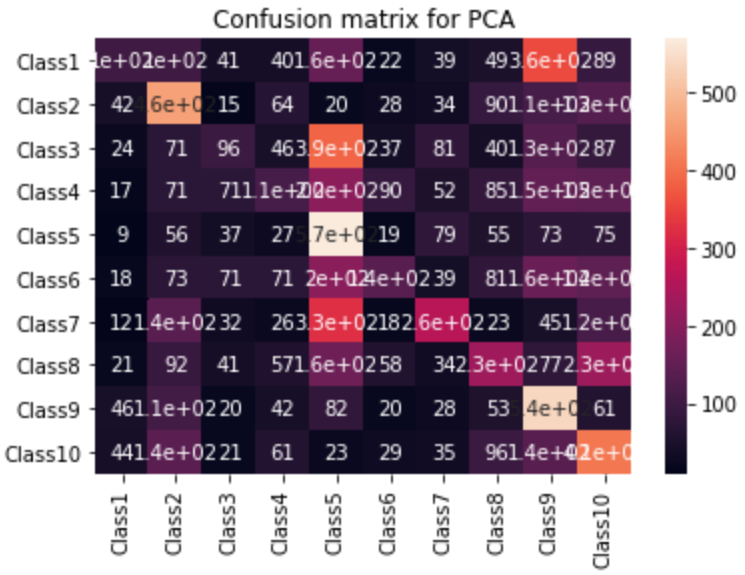




For 90% eigenenergy(top 99 eigenvectors): 33.91% accuracy

For 99% eigenenergy(top 658 eigenvectors): 29.27% accuracy





Unlike expected, the accuracy on PCA projected Cifar-10 data decreases with increase in the eigenenergy could be simply because the the **# of dimensions is increasing rapidly keeping the # of samples as constant** making it difficult for the classifier to classify the samples correctly. This doesn't happen for the Face dataset since it doesn't have this many features.

i.

Face dataset

1. LDA on PCA projected data: 92.4% accuracy.
2. PCA on LDA projected data: 89.2% accuracy.

Cifar-10 dataset

1. LDA on PCA projected data: 40.46% accuracy
2. PCA on LDA projected data: 36.59% accuracy

This is because **PCA projected data is already along maximum variance in some lower dimensional space and further applying LDA to the data helps in better clustering of the classes leading to better classification** accuracies obviously. On the other hand, applying PCA on LDA projected data doesn't really make sense since the LDA projected data is already clustered as per the classes and further projecting it to lower dimensions might lose the clusters.

Ques2.

1. A function boosting was implemented for the same that takes the argument of N, where N is the # of weak-learners learned for the task. A few notes:-
   a. Since most of the weak-learners have error > 50%, alphas are taken to be of absolute value.
   b. Decision is made by picking out the class for which probability is max after the predictions by all weak-learners.

The accuracy after Boosting the weak-learners(10 trees) on the training set comes out to be 46.72% and that on the test set comes out to be 46.4%. Respective error rates are (1-accuracy).

```
accuracy= getAcc(Y_test, probstest)
accuracy2= getAcc(Y_train, probstrain)
print ("The accuracy on the train set for Boosting is: " + str(accuracy2))
print ("The accuracy on the test set for Boosting is: " + str(accuracy))
```

```
[ 0  1  2  6  7  8  9 10 11 12 13 14 15 17 18 20 21 22 23 24 25]
[ 0  1  2  6  7  8  9 10 11 12 13 14 15 17 18 20 21 22 23 24 25]
The accuracy on the train set for Boosting is: 0.4672142857142857
The accuracy on the test set for Boosting is: 0.464
```

3. With 10 learners, the mean accuracy on the validation set is 44.99% and the standard deviation is 1.89%. The best model during C.V. gives an accuracy of 46.66% on the test set.

4. The accuracy on the test set with Bagging using 10 weak learners comes out to be 27.51%.
During Cross-Validation the mean accuracy on the validation set 25.06% and the standard deviation as 3.5%. The best model during C.V. gives accuracy of 29.68% on the test set.

5.

Accuracy without any sort of normalization is 25.55%

1) Min-Max normalization: 25.55% accuracy
2) Z-score normalization: 25.55% accuracy
3) Tanh normalization: 25.55% accuracy

The reason for the above observations could be attributed to the fact that all the above functions are monotonic. They only change the values of the probabilities by

squashing them to (0-1) range but the probability that was maximum earlier would still be the maximum thus causing no change in the accuracies whatsoever.