# CS 7616
# HW 1

# Student: Anisha Gartia
# GTID: 903136557


## For Wine Dataset

Descriptiption of method used:

Here we used SVD, eigen values, covariance, and Baye's theorem to implement PCA, LDA and Naïve Baye's theorems.

PCA is used for dimensionality reduction. It can also be used as pre processing as we used in MNIST dataset, as MNIST test matrix is a singular matrix.

LDA is a linear analysis method that separates the probability distribution functions of different classes by making them more closely related within each class, and father apart from each other.

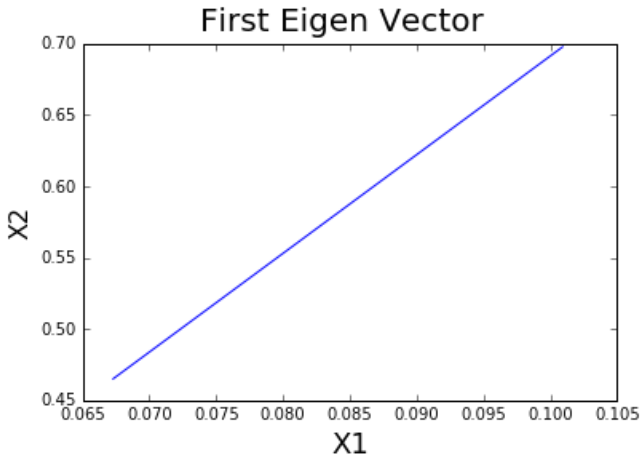
Here we have used only Binary classes. For multiple classes, one can be considered as class1, and the others, class2. Then, within the other's class, we can have another binary classification. Thus finally we can classify between multiple classes. For LDA, we will need to use a higher dimension eigen vectors of we want to classify for more than 2 classes.

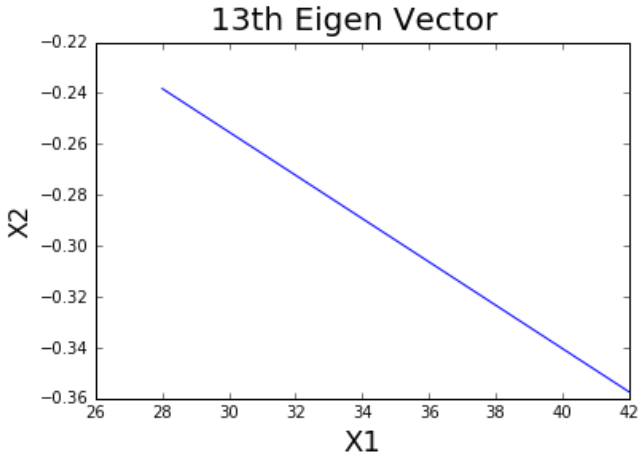
In MNIST dataset, we saw a stark difference in the classification accuracy of 0,1 and 3,5. This is so as 0 and 1 are geometrically quite different, where as 3 and 5 are more closer to look at. Thus their features(pixel wise) would be much closer to each other.

## class 1 and 2, train set = five randomly chosen examples of each class

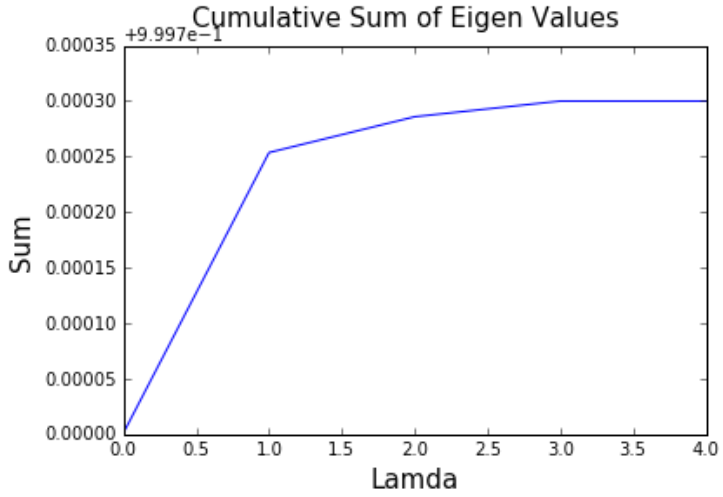
## PCA

- display the most important eigenvector and the 20th eigenvector

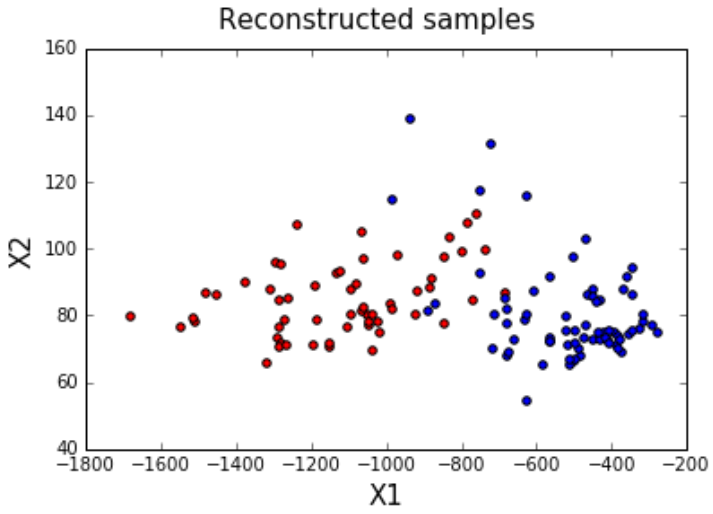



- plot the sorted cumulative sum of eigenvalues



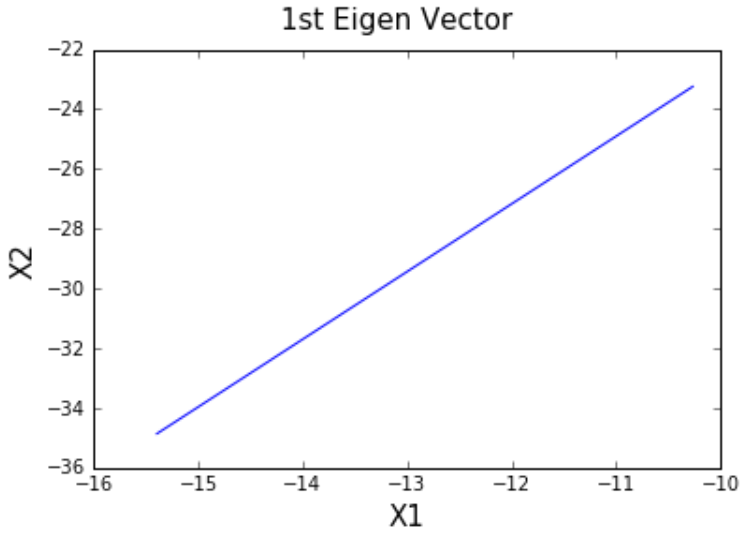
- reconstruct a test example and show the reconstruction error



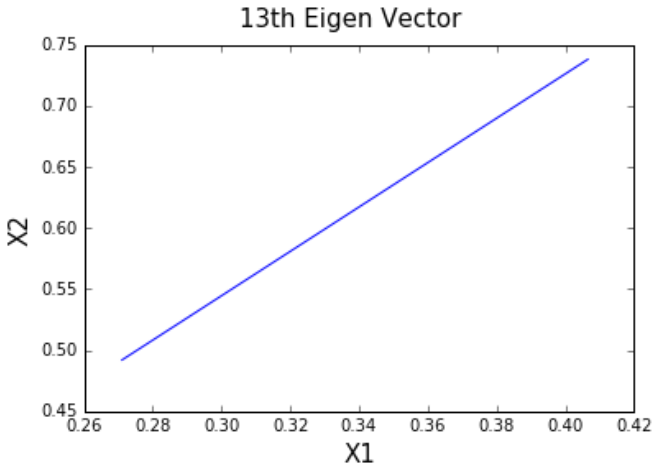

Reconstruction Error: `4.63651669365e-05`

## LDA

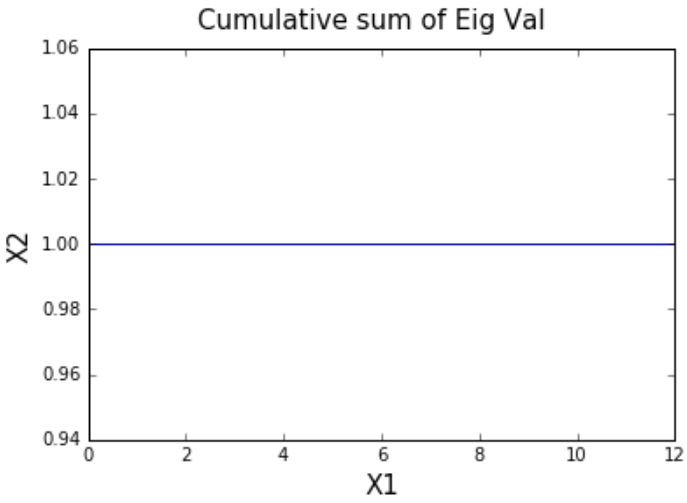
- display the most important eigenve



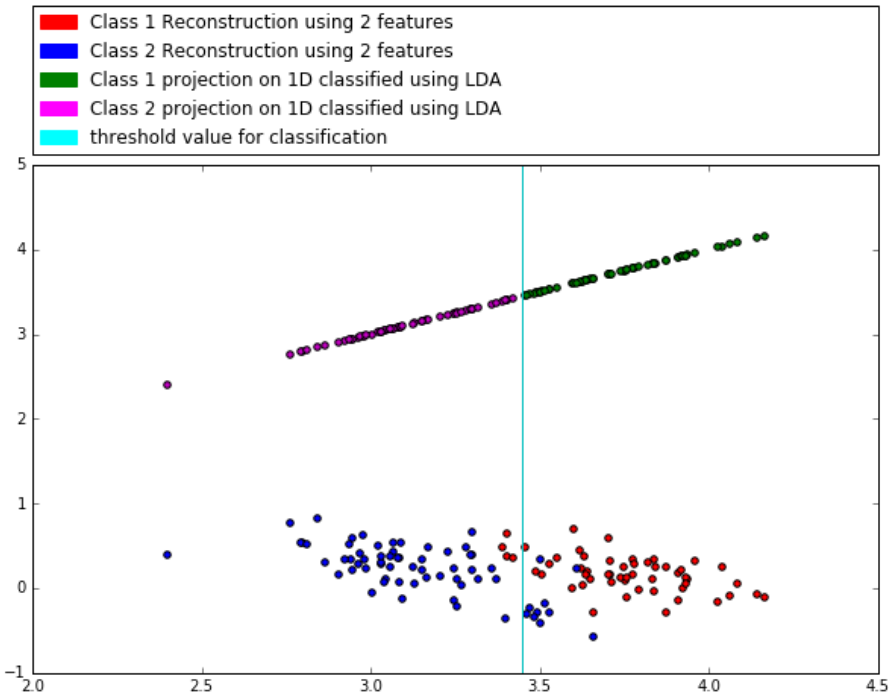

- ctor and the 20th eigenvector




- plot the sorted cumulative sum of eigenvalues



- reconstruct a test example and show the reconstruction error




- A confusion matrix on the results of your algorithm on the test data.

```
Accuracy of LDA is 88.3333333333
```

- An accuracy table that has all of your classifcation results

```
confusion matrix is
[[50  4]
[10 56]]
```

### Naïve Baye's

- A confusion matrix on the results of your algorithm on the test data.

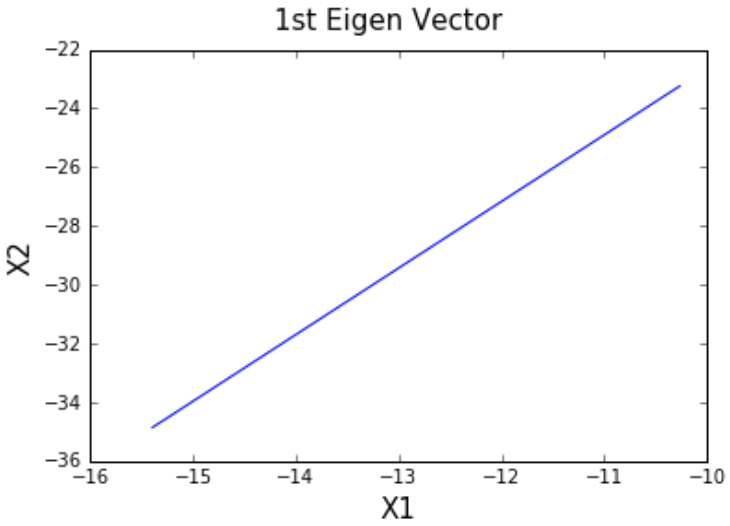
```
Accuracy percentage is  87.5
Confusion Matrix is
[[54  0]
[15 51]]
```

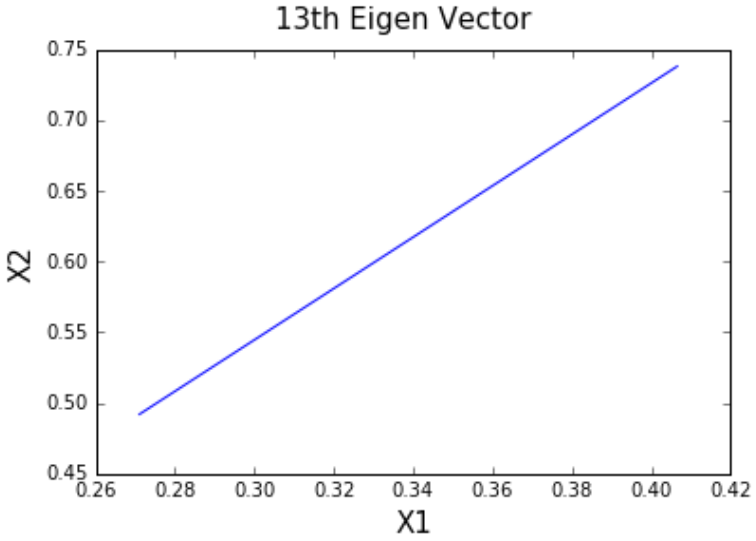
- An accuracy table that has all of your classi_cation results

## class 1 and 2, train set = 50 randomly chosen examples of each class

### PCA

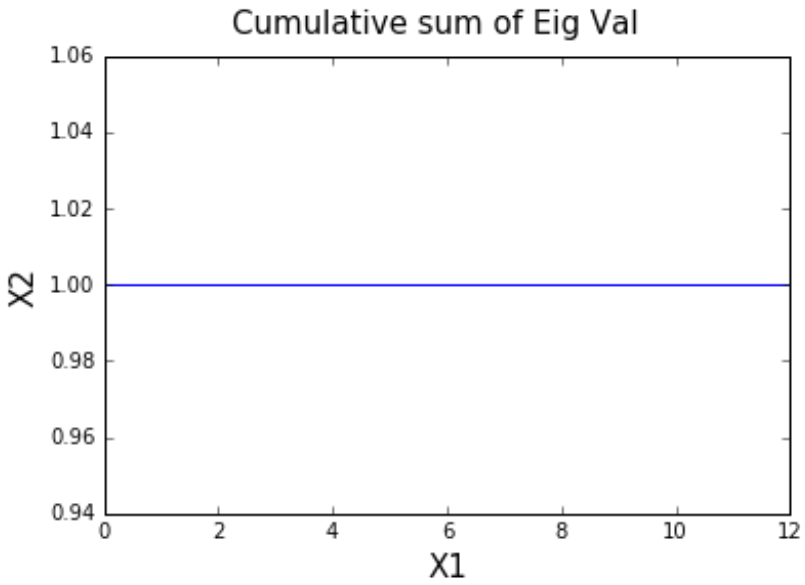
- display the most important eigenvector and the 20th eigenvector


1st Eigen Vector
X2
X1
−22
−24
−26
−28
−30
−32
−34
−36
−16
−15
−14
−13
−12
−11
−10


13th Eigen Vector
X2
X1
0.75
0.70
0.65
0.60
0.55
0.50
0.45
0.26
0.28
0.30
0.32
0.34
0.36
0.38
0.40
0.42

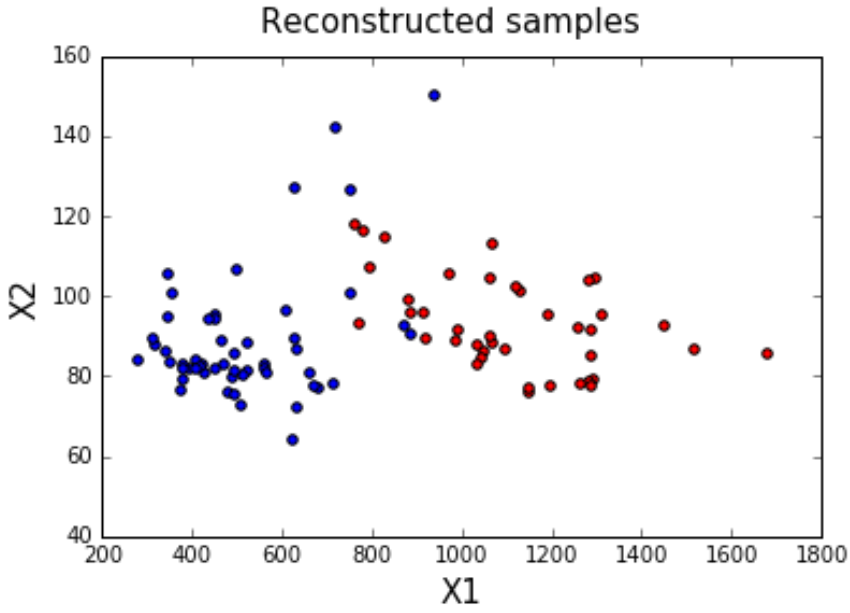
- 
- 

- plot the sorted cumulative sum of eigenvalues



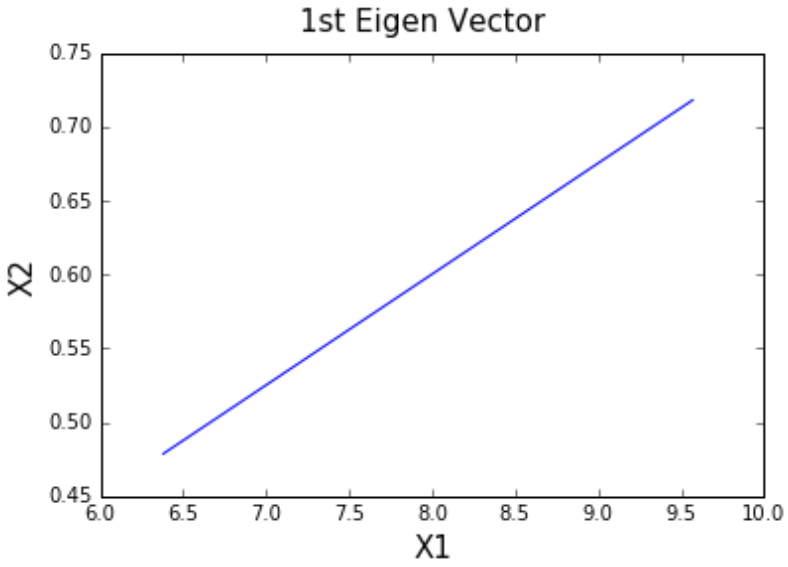

-
  - 
  - reconstruct a test example and show the reconstruction error

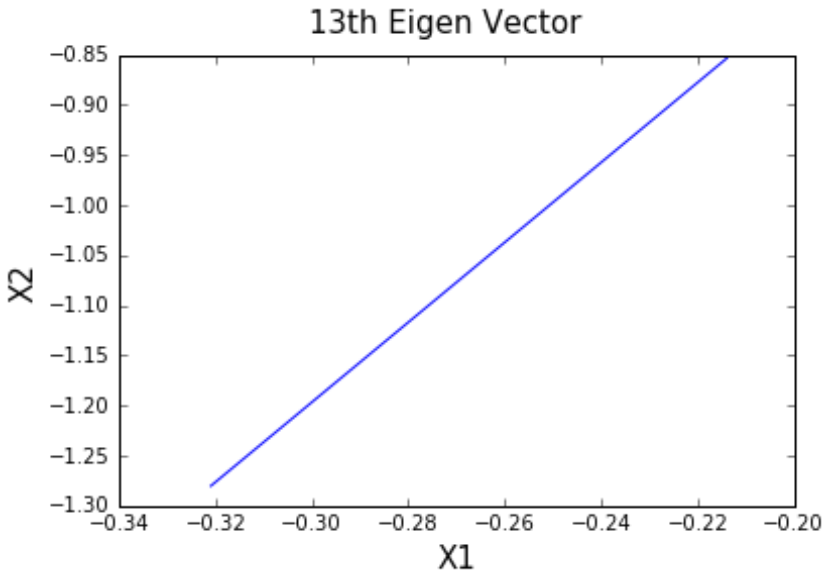
Reconstruction Error is `0.000183439848386`



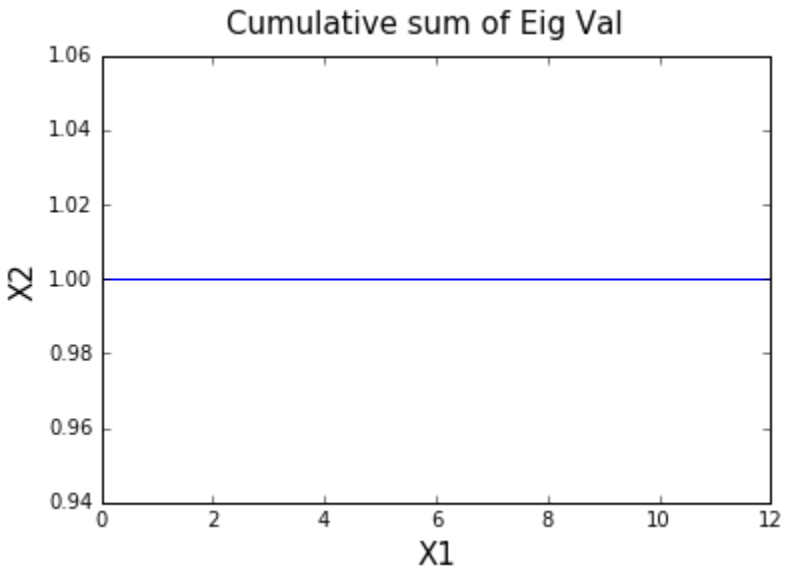
## LDA

- display the most important eigenvector and the 20th eigenvector

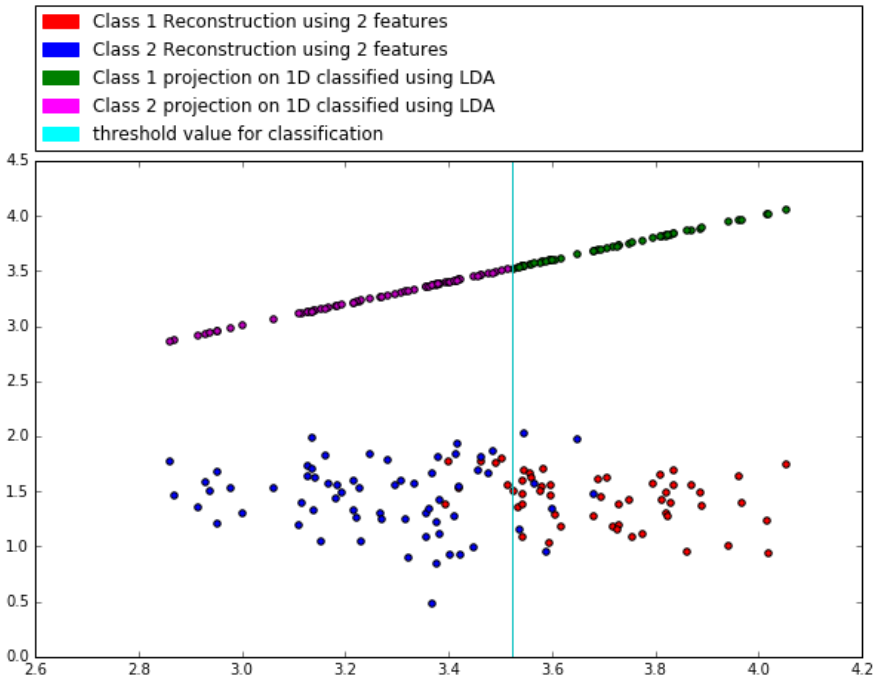


- plot the sorted cumulative sum of eigenvalues




- reconstruct a test example and show the reconstruction error




- A confusion matrix on the results of your algorithm on the test data.

```
conusion matrix is
[[47  7]
[ 7 59]]
```

- An accuracy table that has all of your classifcation results

```
Accuracy of LDA is
88.3333333333
```

## Naïve Baye's

- A confusion matrix on the results of your algorithm on the test data.

```
Confusion Matrix is
[[ 9  0]
[ 1 20]]
```

- An accuracy table that has all of your classi_cation results

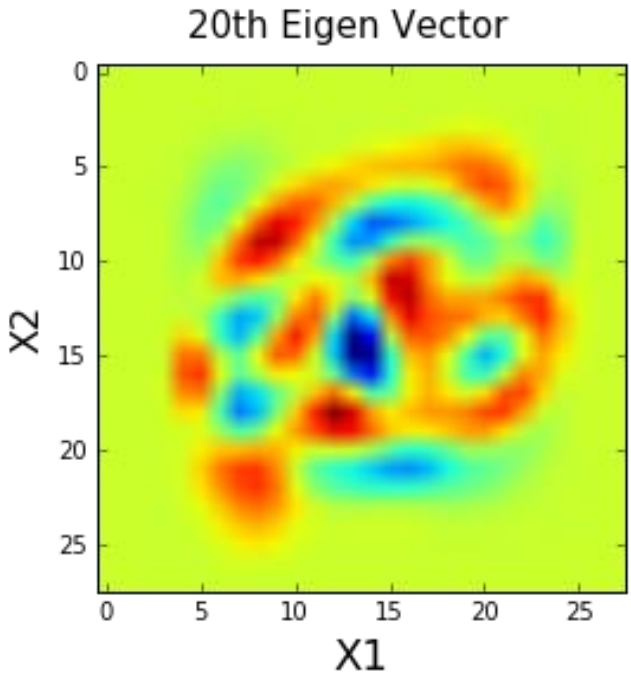
```
Accuracy percentage is  96.6666666667
```

## For MNIST Dataset

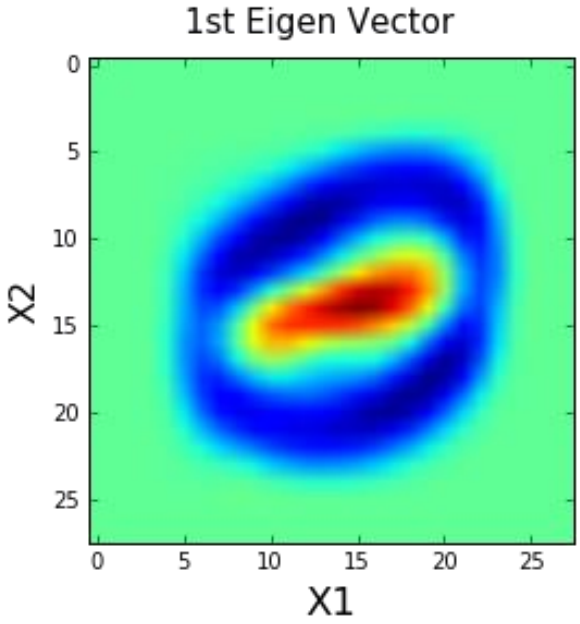
### Train and test on just the numbers of 0 and 1.

#### PCA

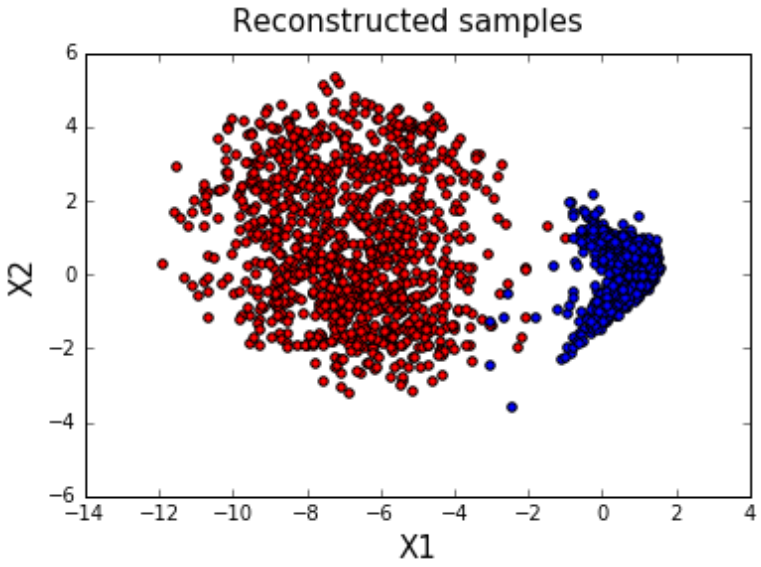
- display the most important eigenvector and the 20th eigenvector. Display as

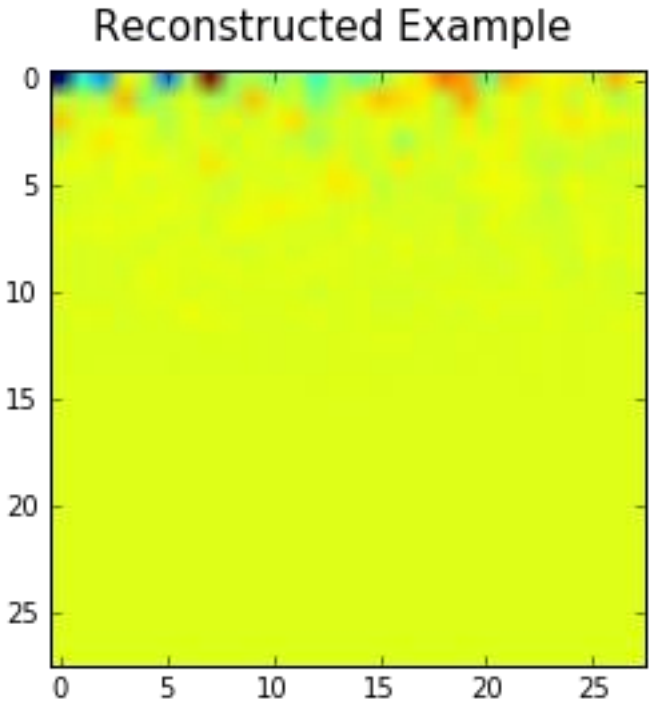



image

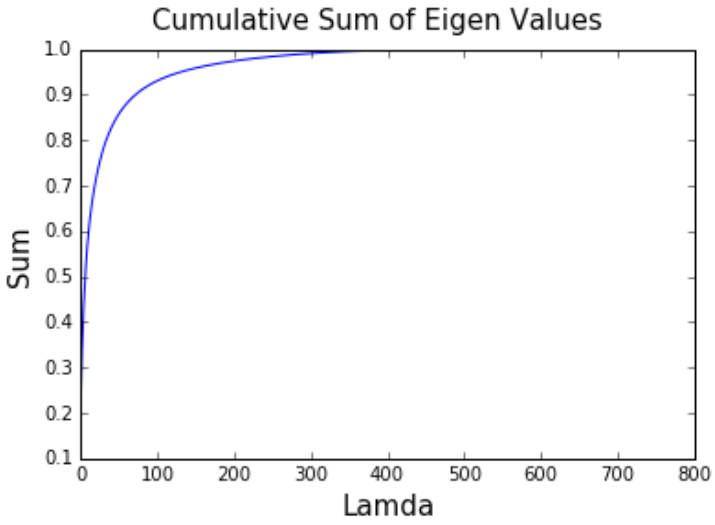


- reconstruct a test example and show the reconstruction error



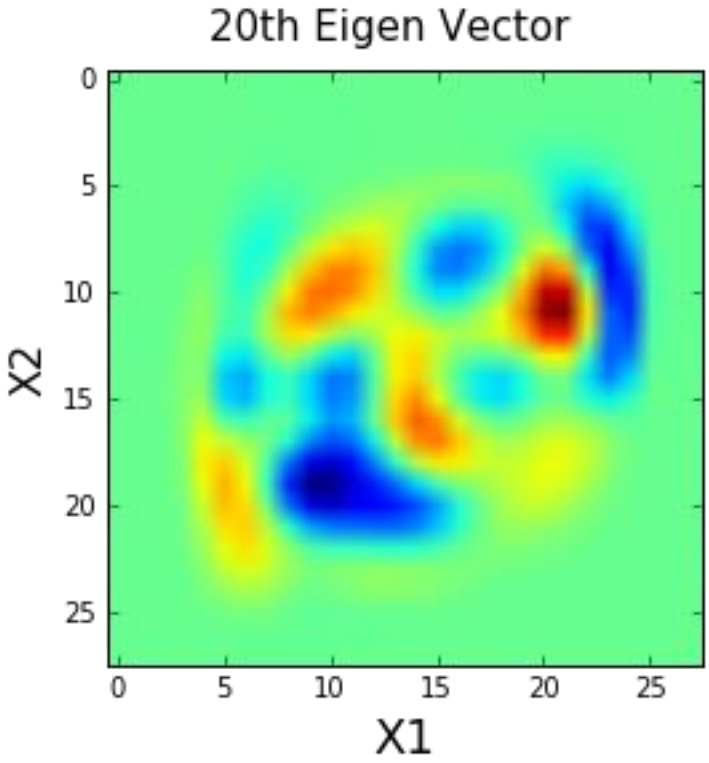



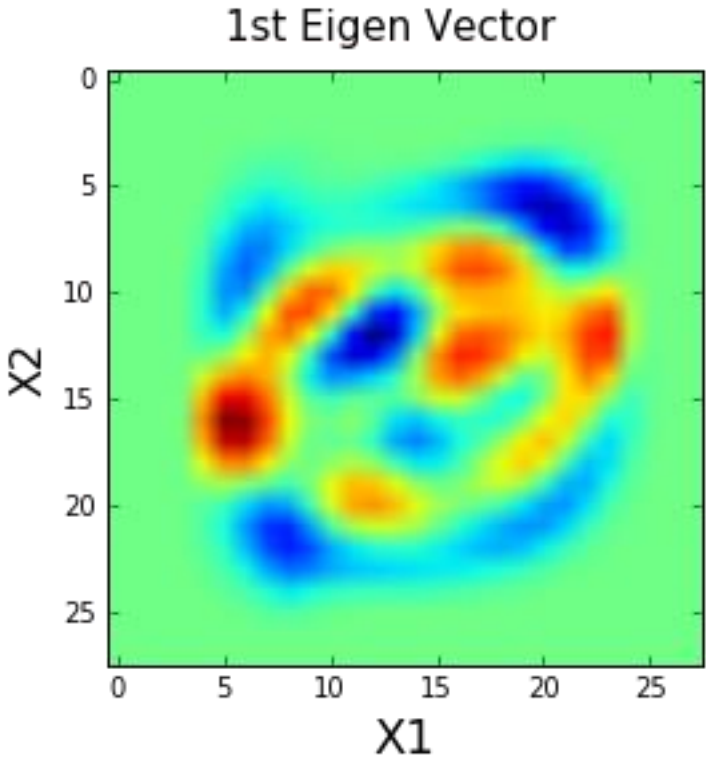
- plot the sorted cumulative sum of eigenvalues



## LDA

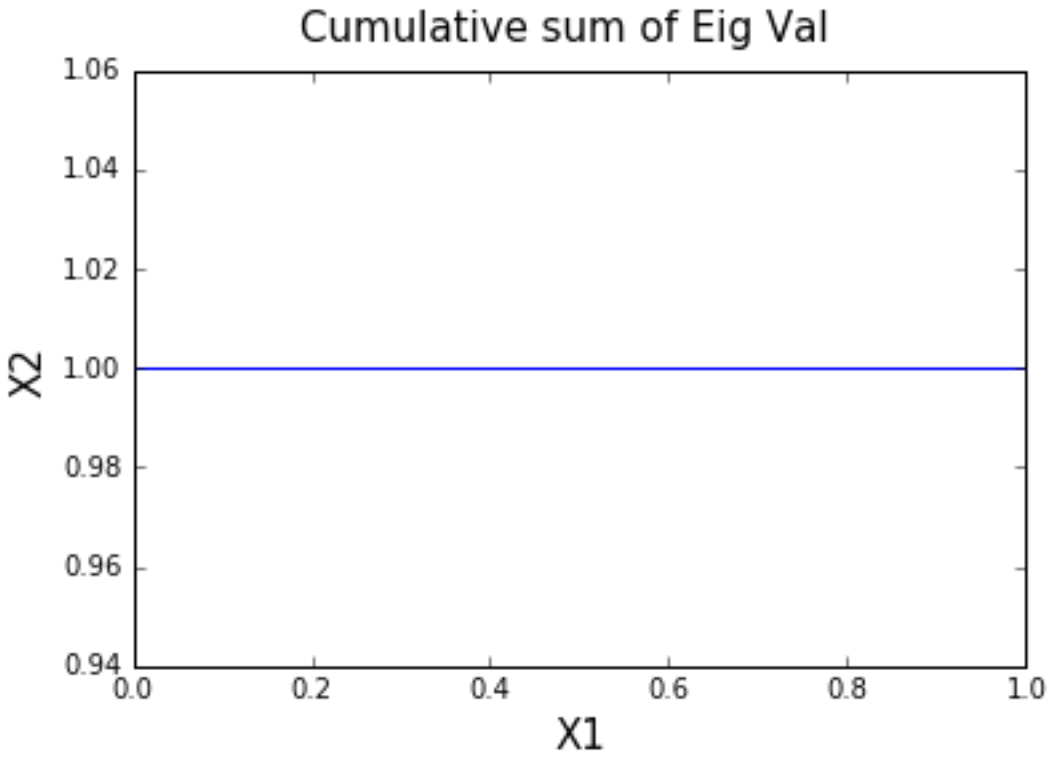
- display the most important eigenvector and the 20th eigenvector. Display as



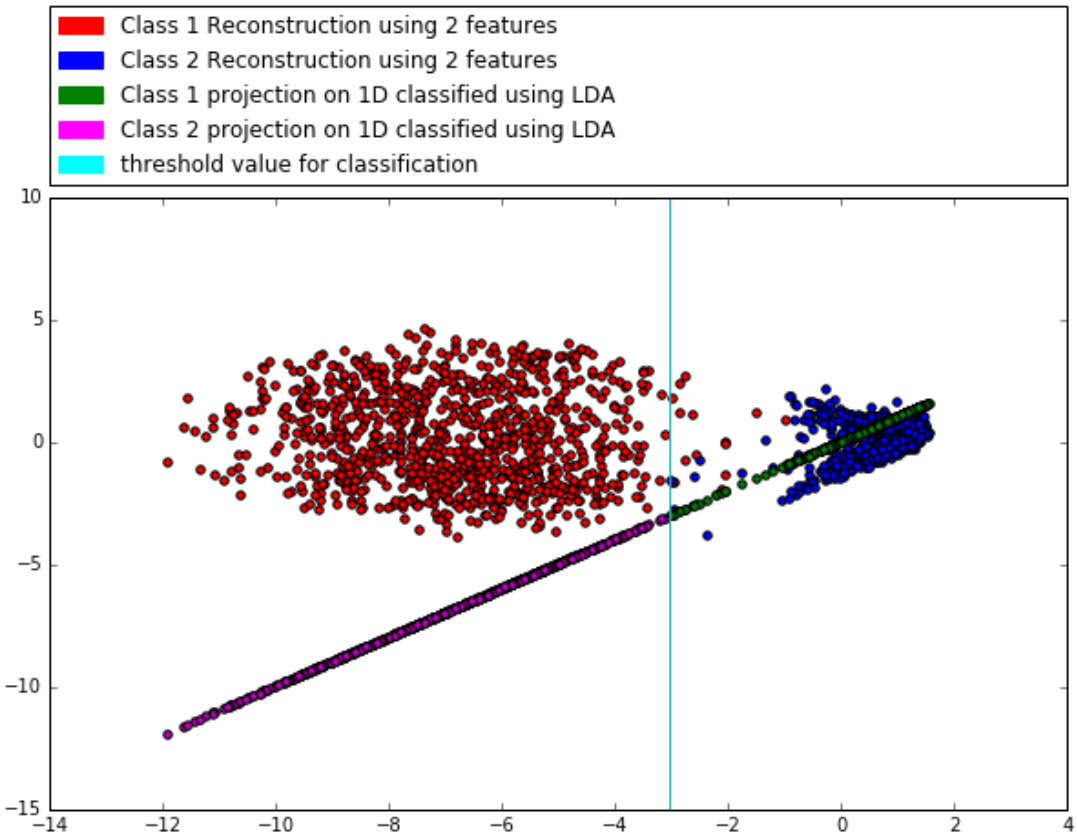



- display the most important eigenvector and the 20th eigenvector. Display as image

- plot the sorted cumulative sum of eigenvalues




- reconstruct a test example and show the reconstruction error




- A confusion matrix on the results of your algorithm on the test data.

```
conusion matrix is
   [[ 964   16]
   [   0 1135]]
```

- An accuracy table that has all of your classi_cation results

```
Accuracy of LDA is  99.243498818
```

## Naïve Baye's

- A confusion matrix on the results of your algorithm on the test data.

```
Confusion Matrix is
    [[ 980    0]
    [  77 1058]]
```

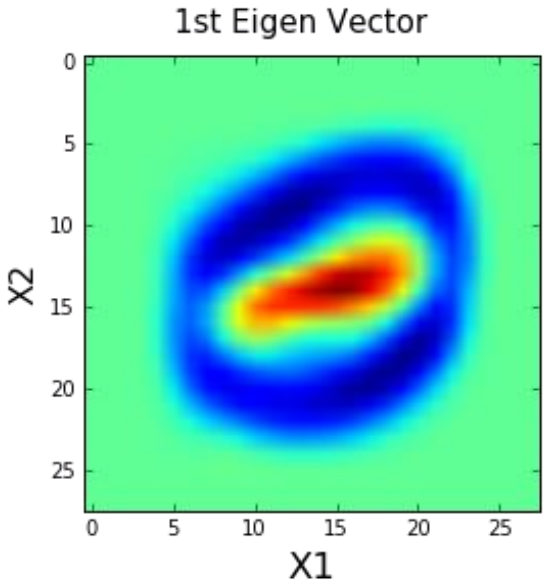
- An accuracy table that has all of your classi_cation results

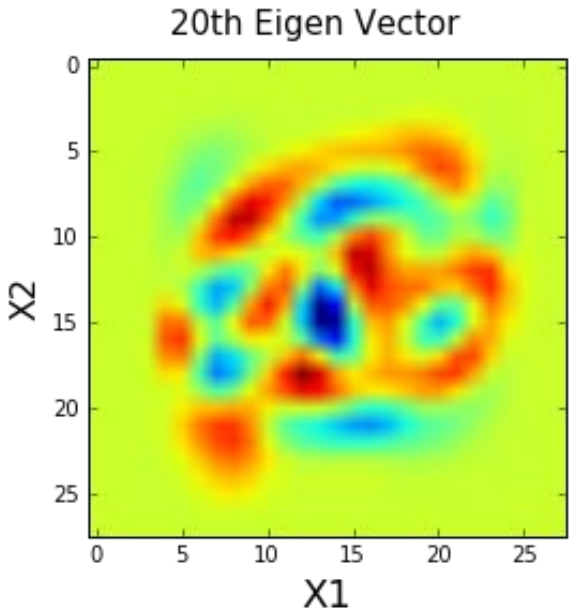
```
Accuracy percentage is  96.3593380615
```

## Train and test on just the numbers of 3 and 5.

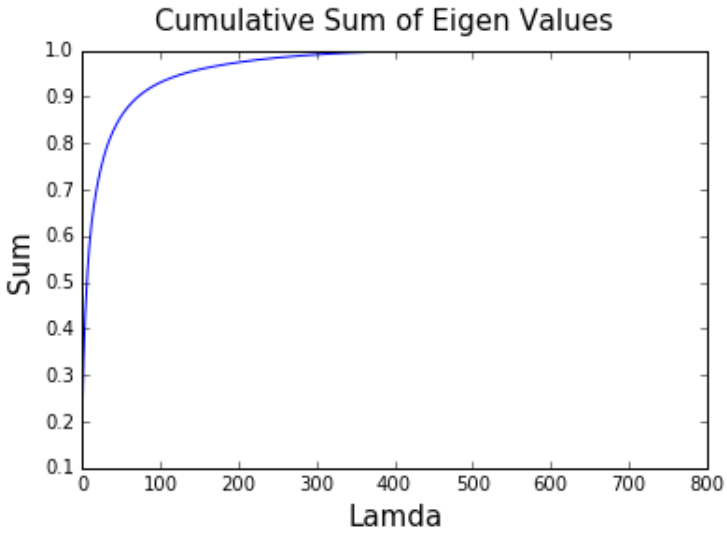
### PCA

- display the most important eigenvector and the 20th eigenvector. show these as an image



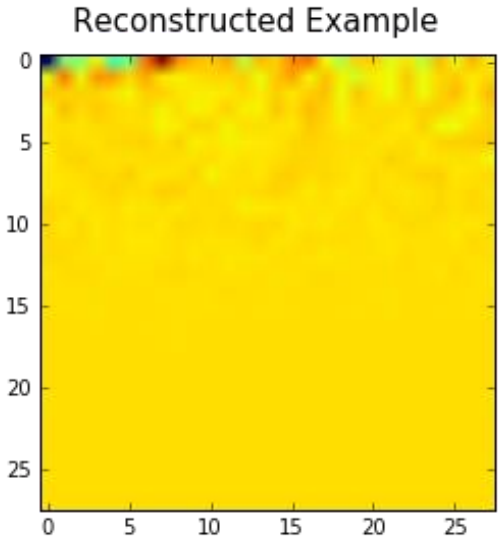



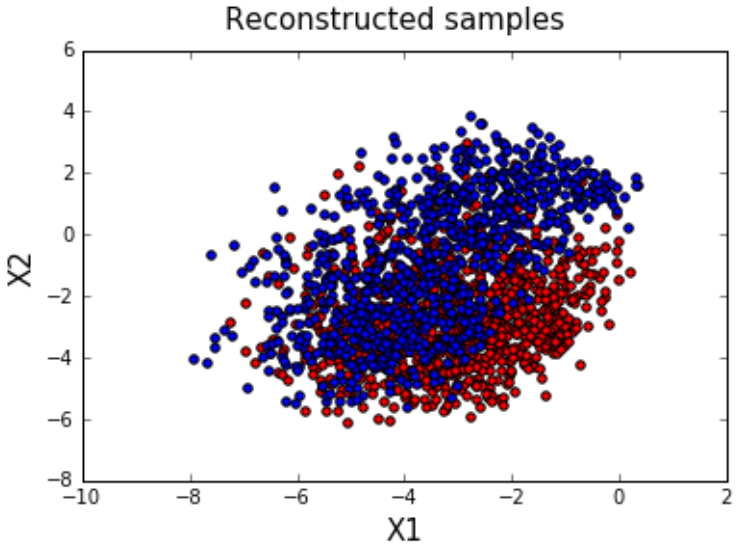

- plot the sorted cumulative sum of eigenvalues




- reconstruct a test example and show the reconstruction error

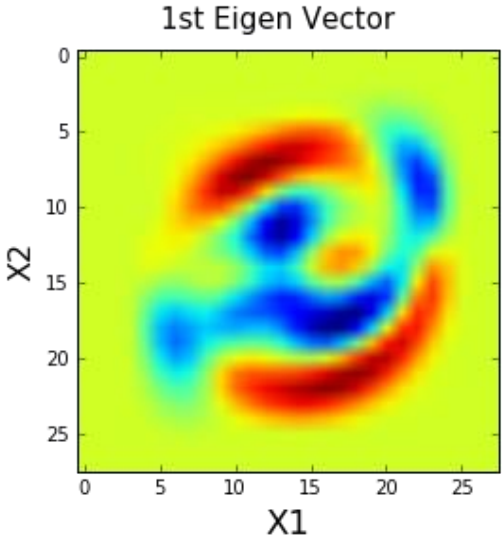
```
0.730744247346
```

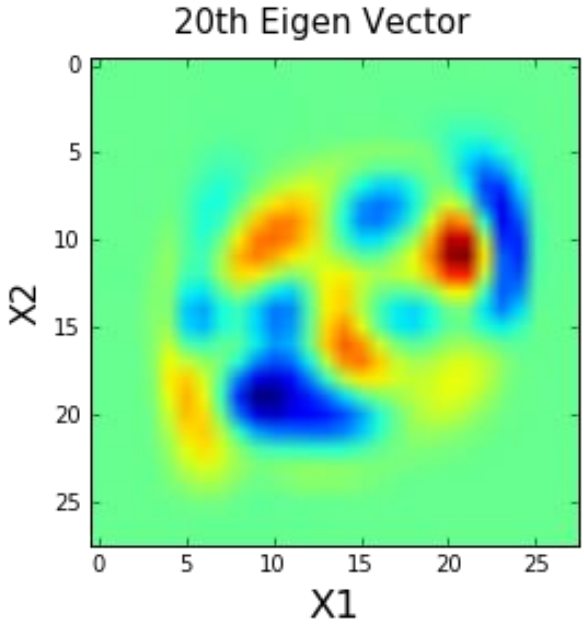



Reconstructed samples
X2
X1
6
4
2
0
−2
−4
−6
−8
−10
−8
−6
−4
−2
0
2

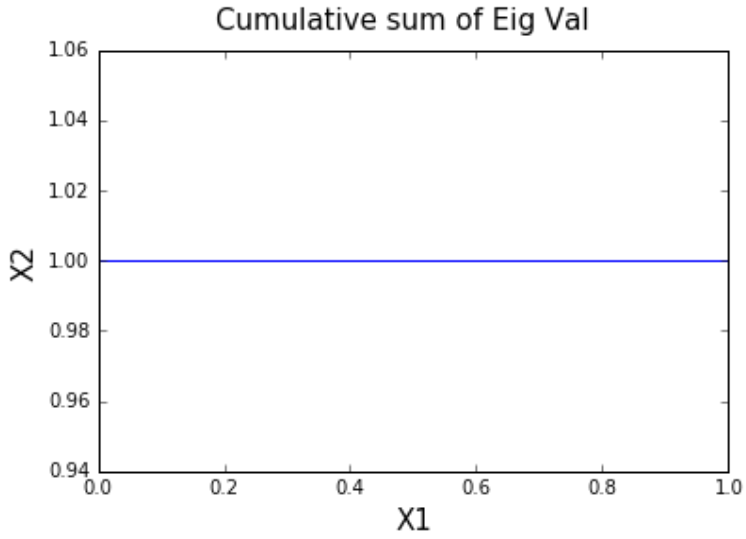
## LDA

- display the most important eigenvector and the 20th eigenvector show these as an image

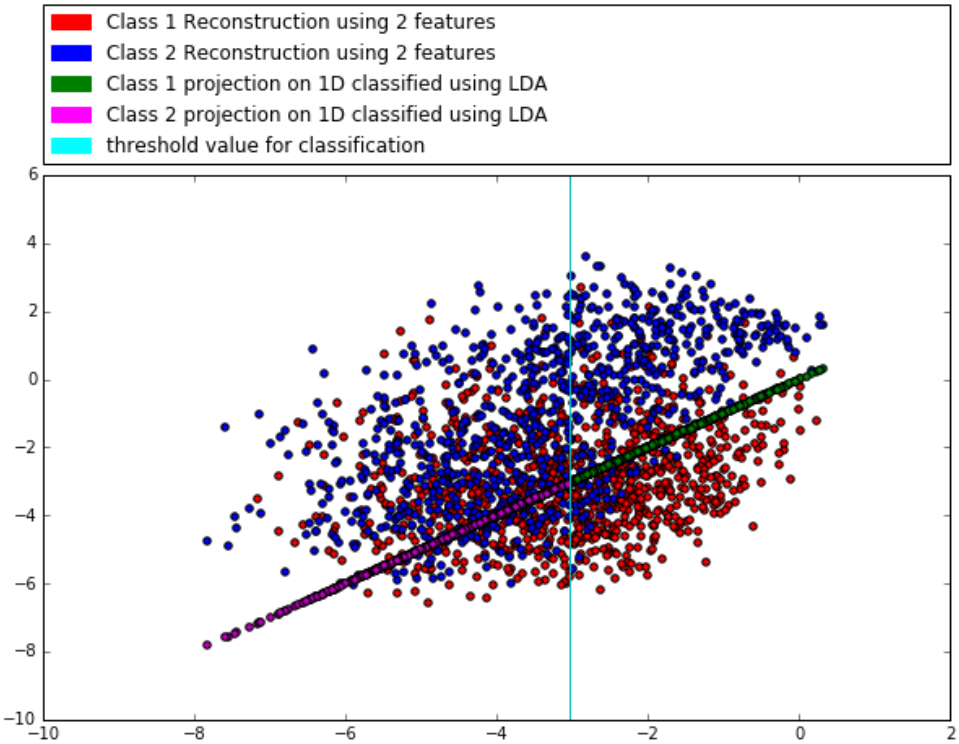


- plot the sorted cumulative sum of eigenvalues




- reconstruct a test example and show the reconstruction error




- A confusion matrix on the results of your algorithm on the test data.

```
conusion matrix is
[[488 522]
[531 361]]
```

- An accuracy table that has all of your classi_cation results

```
Accuracy of LDA is  44.6372239748
```

## Naïve Baye's

- A confusion matrix on the results of your algorithm on the test data.

```
Confusion Matrix is
    [[1006    4]
    [ 390  502]]
```

- An accuracy table that has all of your classi_cation results

```
Accuracy percentage is  79.2849631966
```