Wine Dataset Report

Nivetha Thiruverahan, Ayush Jasuja
A53205226 , A53103338
{nthiruve, ayjasuja}@eng.ucsd.edu

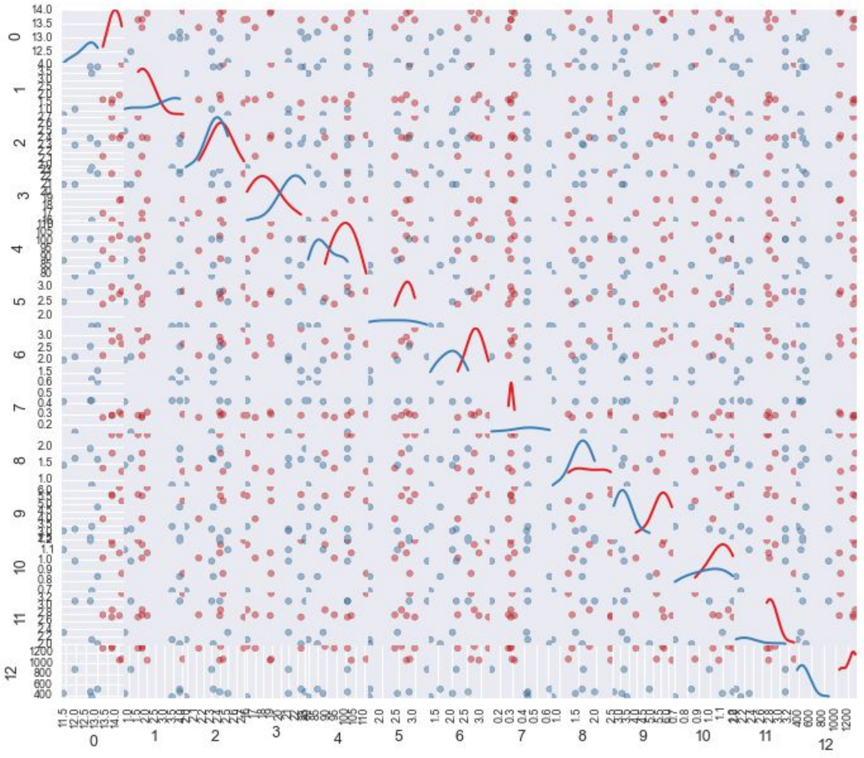
Method

We apply naive bayes to do binary classification of data. We also do data compression using PCA and LDA. Further, we apply LDA and naive bayes classification on compressed data set. In general applying PCA and then doing classification seems to give better accuracy.

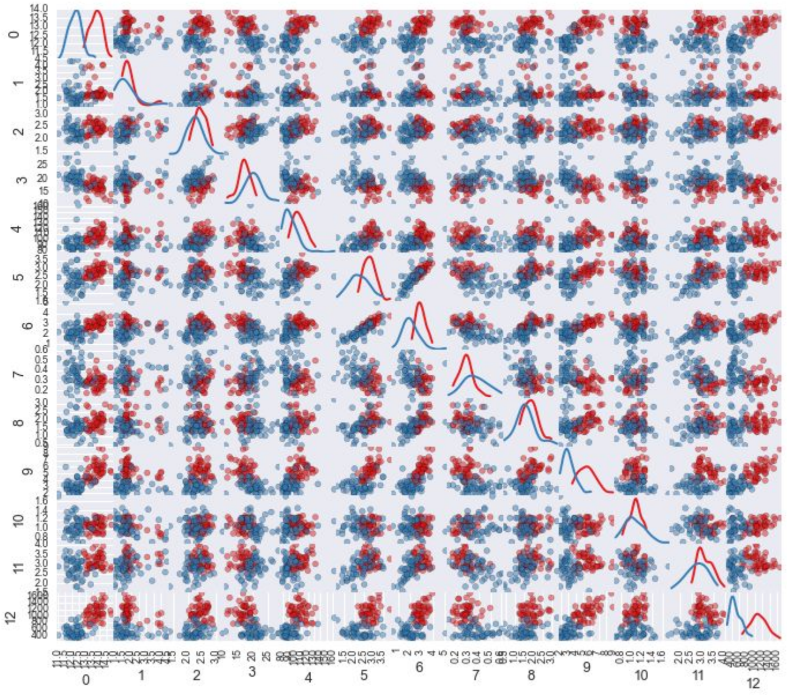
EigenVectors

We tried applying PCA and LDA compression on the dataset with varying number of sample. Below are the scatter plots of the samples used to extract eigenvectors. In all of them it is pretty evident that using just the first eigenvector can give good classification accuracy.

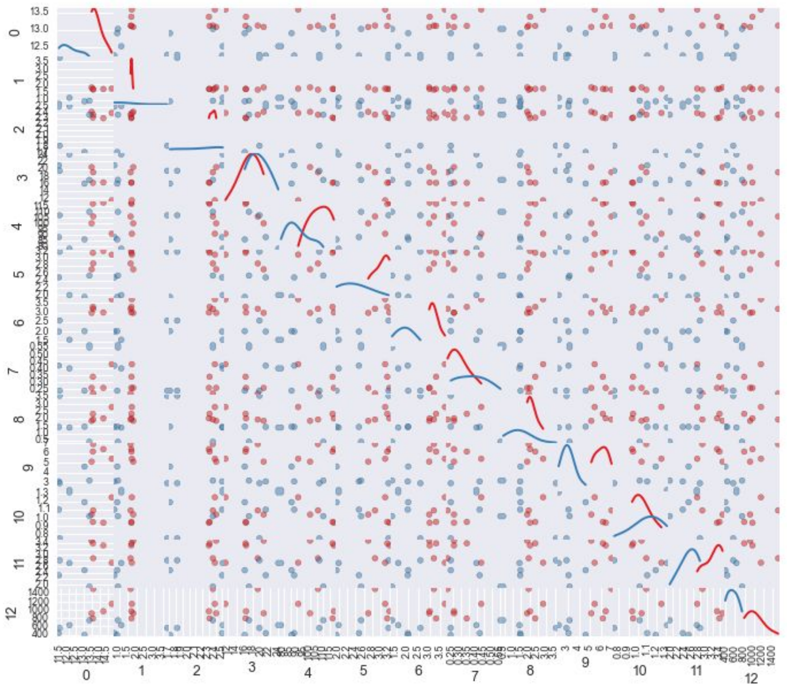
1. PCA with 5 training Samples per class



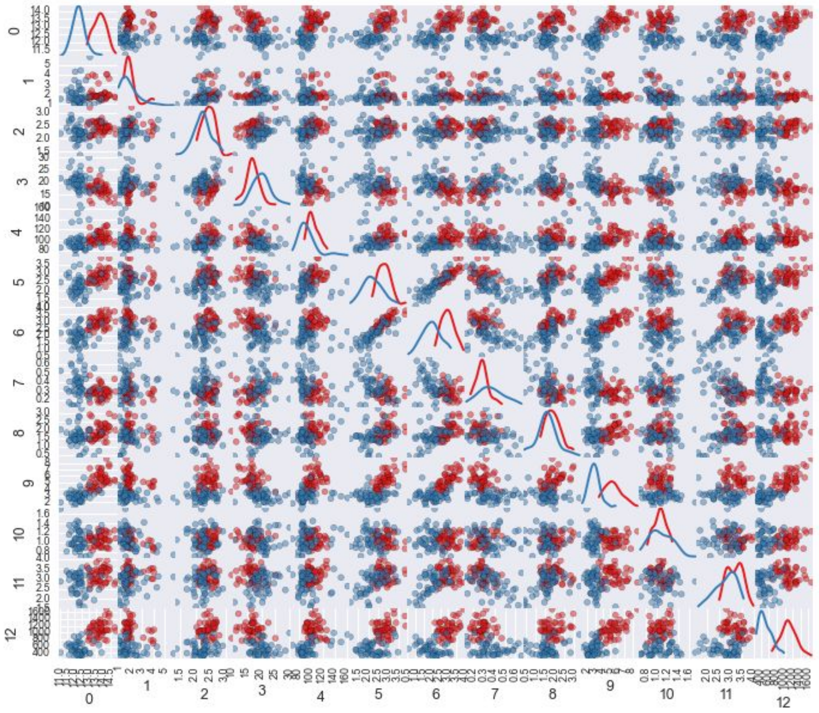
2. PCA with 50 training Samples per class



3. LDA with 5 training samples per class



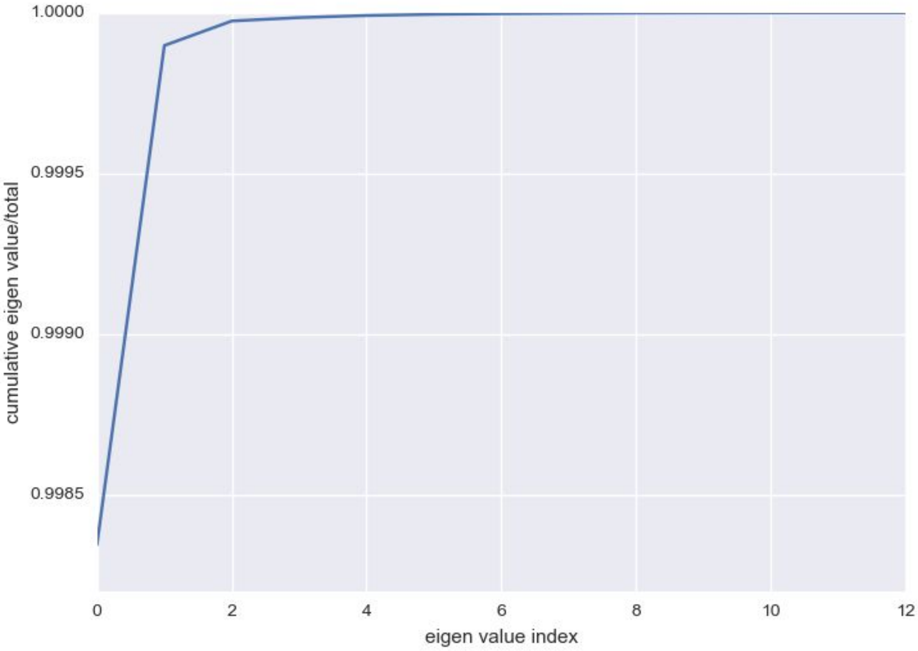
4. LDA with 50 training samples per class



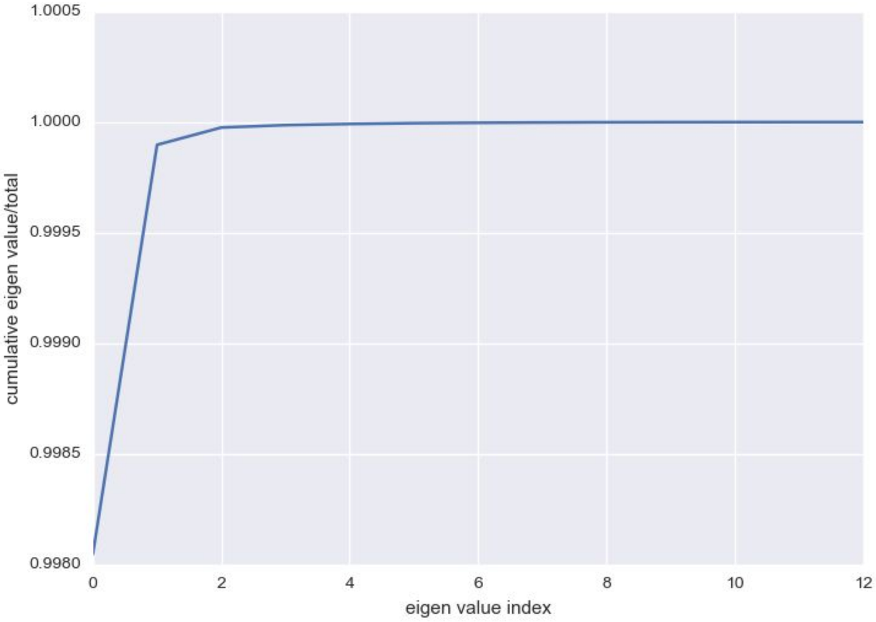
Cumulative EigenValues

Below are the plots of the cumulative eigenvalues corresponding to the eigenvectors of the above approaches to extract eigenvalues. In all of them, first eigenvalue takes more than 70 percent of the sum of eigenvalues. This combined with the observation of eigenvectors (in the above section) provides another reason to just the first eigenvector for classification.

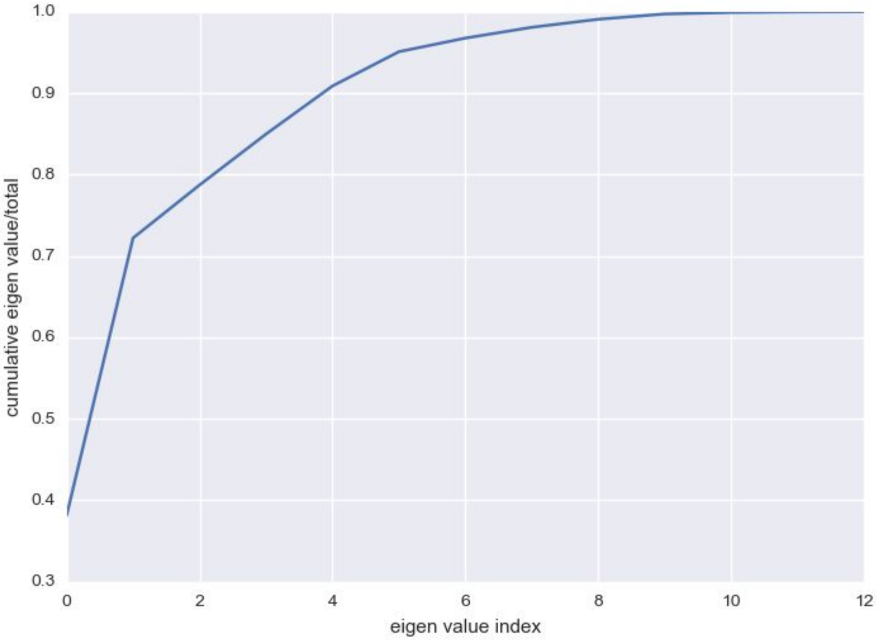
1. PCA with 5 training Samples per class



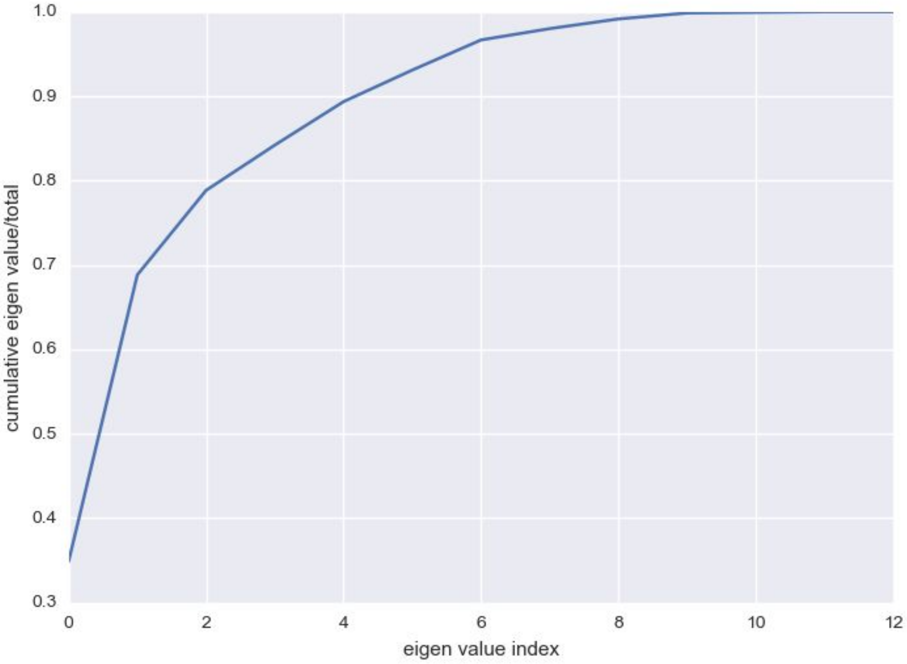
2. PCA with 50 training Samples per class



3. LDA with 5 training Samples per class



4. LDA with 50 training Samples per class



Reconstruction

1. PCA with 5 training Samples per class
   Original = [13.73, 1.5, 2.7, 22.5, 101.0, 3.0, 3.25, 0.29, 2.38, 5.7, 1.19, 2.71, 1285.0]

# Dimensions	Compressed	Reconstructed	Reconstruction Error
13	[[5.07336446e+02 9.84600938e-01 8.48419905e+00 -2.62661032e-02 7.92327090e-01 -1.05306909e+00 -2.56977523e-01 -2.28652766e-01 -3.18712846e-01 4.79955968e-01 1.39193783e-03 1.89757572e-02 -1.64217963e-01]]	[[1.37300000e+01 1.50000000e+00 2.70000000e+00 2.25000000e+01 1.01000000e+02 3.00000000e+00 3.25000000e+00 2.90000000e-01 2.38000000e+00 5.70000000e+00 1.19000000e+00 2.71000000e+00 1.28500000e+03]]	[[2.88714463e-27]]
1	[[493.42096668]]	[[1.36694825e+01 1.82565762e+00 2.34036142e+00 1.69031543e+01 1.09742491e+02 2.81670822e+00 2.84817657e+00 2.25631283e-01 1.94943991e+00 5.74149607e+00 1.02382416e+00 2.96243624e+00 1.28488781e+03]]	[[108.48510938]]

2. PCA with 50 training Samples per class
   Original = [12.08, 2.08, 1.7, 17.5, 97.0, 2.23, 2.17, 0.26, 1.4, 3.3, 1.27, 2.96, 710.0]

# Dimensions	Compressed	Reconstructed	Reconstruction Error
13	[[-1.21752200e+02 -1.26962497e+00 -1.55794468e+00 -9.66022207e-01 1.38401522e-01 -7.50598944e-02 -4.41307884e-01 -3.17022644e-01 1.68722661e-01 -4.25679769e-01 -4.21777530e-01 1.20394112e-01 6.45501292e-03]]	[[1.20800000e+01 2.08000000e+00 1.70000000e+00 1.75000000e+01 9.70000000e+01 2.23000000e+00 2.17000000e+00 2.60000000e-01 1.40000000e+00 3.30000000e+00 1.27000000e+00 2.96000000e+00 7.10000000e+02]]	[[1.57755233e-27]]
1	[[-121.75219996]]	[[1.28050402e+01 1.98284681e+00 2.30026667e+00 1.91641910e+01 9.82054319e+01 2.42401245e+00 2.34438104e+00 3.34748764e-01 1.69860696e+00 3.90300452e+00 1.05313282e+00 2.90901766e+00 7.09980043e+02]]	[[5.69448638]]

3. LDA with 5 training Samples per class

Original = [1.48300000e+01 1.64000000e+00 2.17000000e+00 1.40000000e+01 9.70000000e+01 2.8 2.98000000e+00 2.90000000e-01 1.98000000e+00 5.20000000e+00 1.08000000e+00 2.85 1.04500000e+03]

# Dimensions	Compressed	Reconstructed	Reconstruction Error
13	[[3.64052417e+00 -3.94844896e+00 2.53124420e+00 -1.04574239e+00 -1.40954062e+00 -6.06310693e+00 1.40059685e+00 2.21097275e+02 3.96151598e-01 3.91947787e-02 4.22863909e-01 -1.48091248e-01 3.17922192e-01]]	[[1.48300000e+01 1.64000000e+00 2.17000000e+00 1.40000000e+01 9.70000000e+01 2.80000000e+00 2.98000000e+00 2.90000000e-01 1.98000000e+00 5.20000000e+00 1.08000000e+00 2.85000000e+00 1.04500000e+03]]	[[5.98324866e-26]]
1	[[3.64052417]]	[[1.27719340e+01 1.65200092e+00 2.37835673e+00 1.95297988e+01 1.01279810e+02 1.35859910e+00 3.10927269e+00 7.17615900e-01 2.00088352e+00 4.12657639e+00 3.06603415e+00 2.30701688e+00 8.23894997e+02]]	[[48948.2658948]]

4. LDA with 50 training Samples per class

Original = [[1.24700000e+01 1.52000000e+00 2.20000000e+00 1.90000000e+01 1.62000000e+02 2.50000000e+00 2.27000000e+00 3.20000000e-01 3.28000000e+00 2.60000000e+00 1.16000000e+00 2.63000000e+00 9.37000000e+02]]

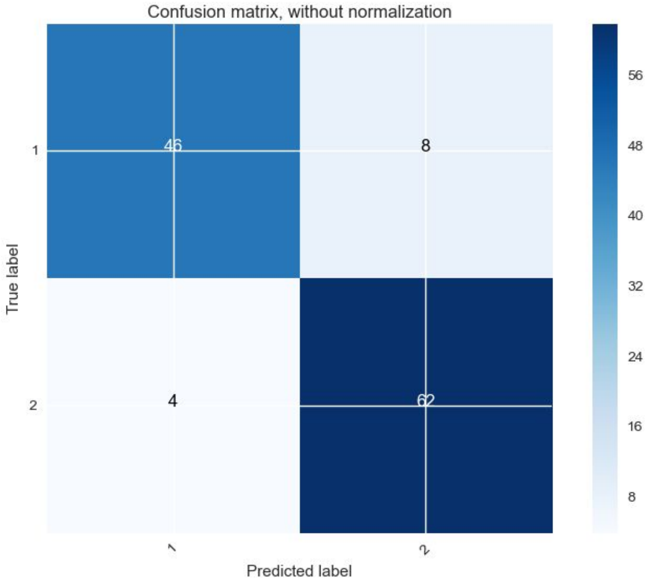
# Dimensions	Compressed	Reconstructed	Reconstruction Error
13	[[-4.11602620e+01 4.16284478e+01 -7.14896244e+00 3.02287733e+00 -1.15889206e+02 -7.98928467e+00 6.47805146e+00 1.54324843e+01 4.70524739e-02 5.60871509e+00 1.61611415e+00 9.78810964e-01 -4.75799308e-01]]	[[1.24700000e+01 1.52000000e+00 2.20000000e+00 1.90000000e+01 1.62000000e+02 2.50000000e+00 2.27000000e+00 3.20000000e-01 3.28000000e+00 2.60000000e+00 1.16000000e+00 2.63000000e+00 9.37000000e+02]]	[[1.51241183e-26]]
1	[[-41.16026202]]	[[16.67883171 2.08646962 3.41656068 11.32302287 127.55011493 2.6208368 3.48354453 14.11890244 -0.99749304 8.60087734 -14.23204239 23.28848132 821.12267345]]	[[15602.69283546]]

Reconstruction error decreases as we increase the number of retained components in the transformed data for both PCA and LDA compression. But in general, PCA compression if better for reconstructing data (especially if the number of components in transformed data is less) because the reconstruction is very less in PCA as compared to LDA.

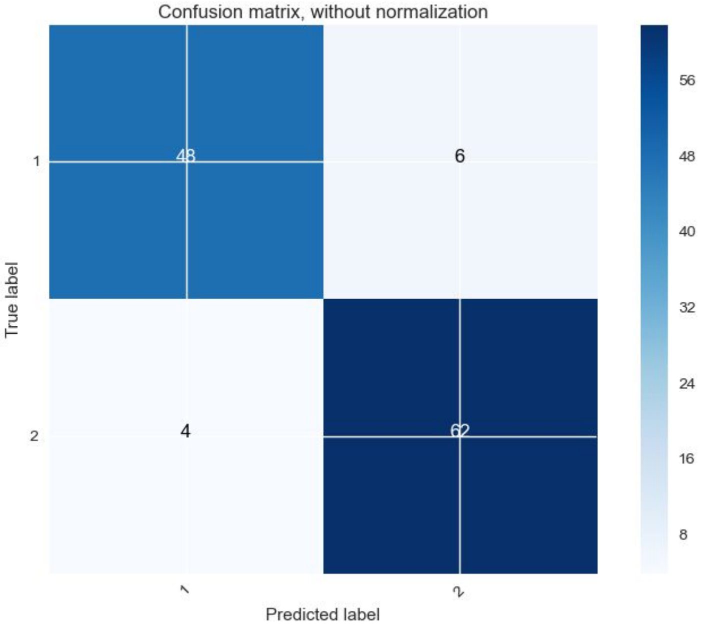
Confusion Matrix

Below are the confusion matrices for various cases (NB, LDA with NB/LDA classification, PCA with NB/LDA classification) for two classes (1 and 2)

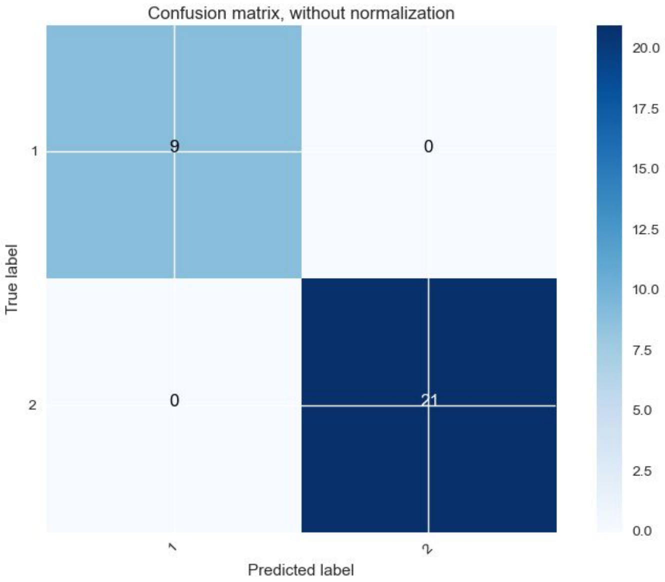
1. PCA compression and LDA classification with 5 training Samples per class



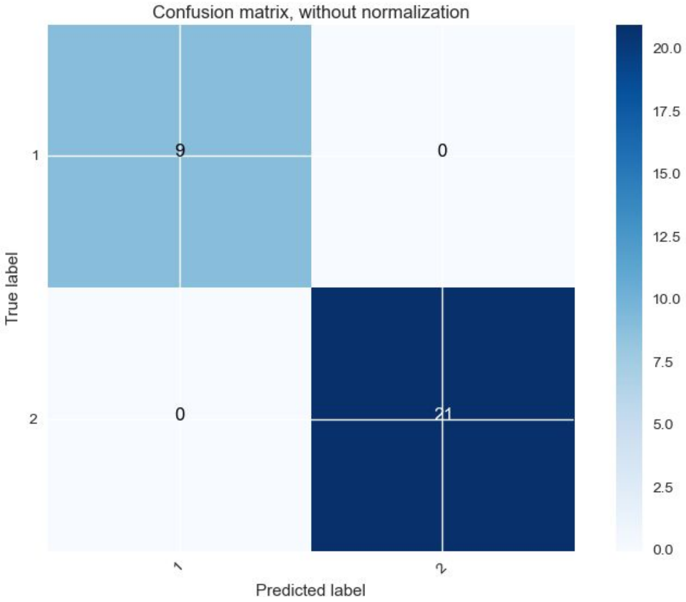
2. PCA compression and NB classification with 5 training Samples per class



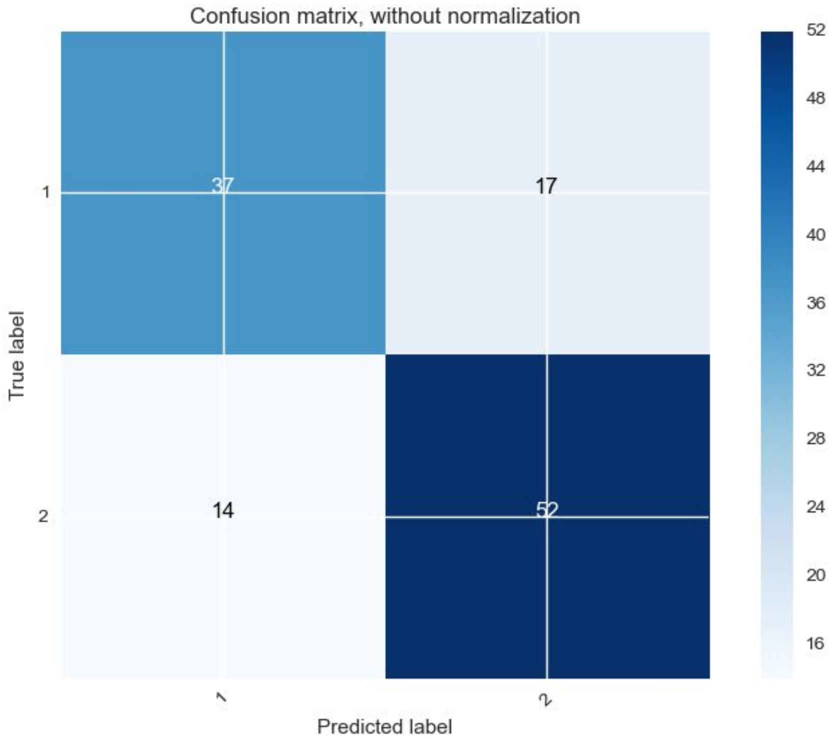
3. PCA compression and LDA classification with 50 training Samples per class



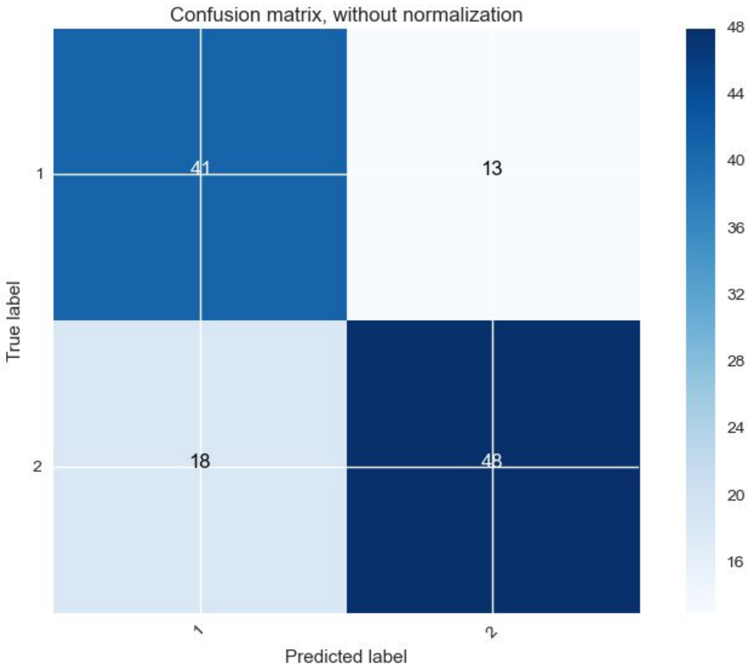
4. PCA compression and NB classification with 50 training Samples per class



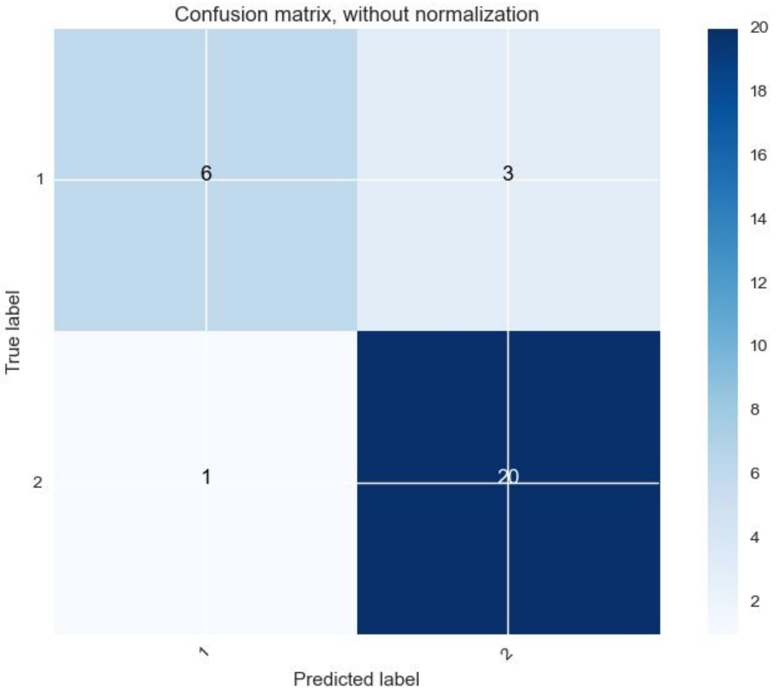
5. LDA compression and LDA classification with 5 training Samples per class



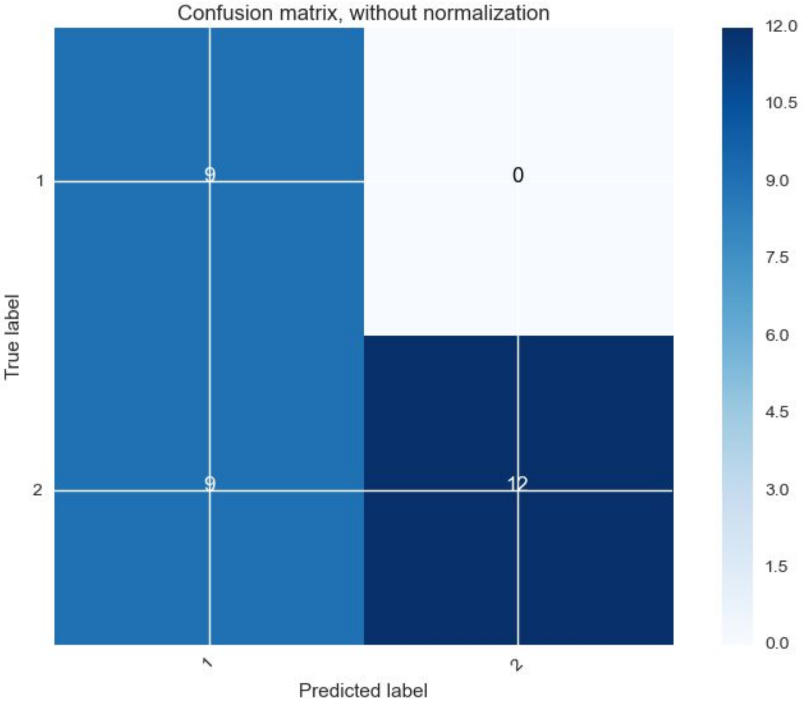
6. LDA compression and NB classification with 5 training Samples per class



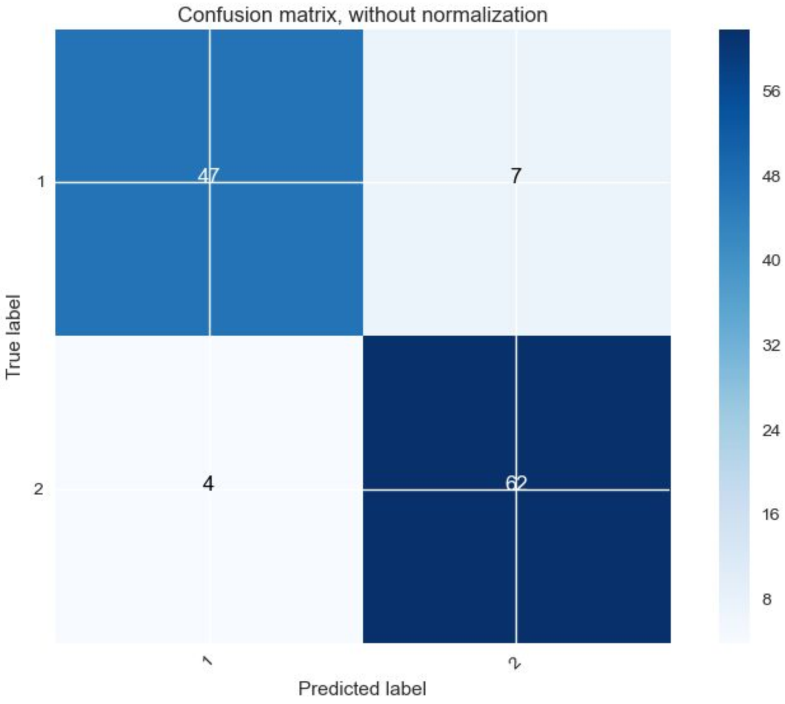
7. LDA compression and LDA classification with 50 training Samples per class



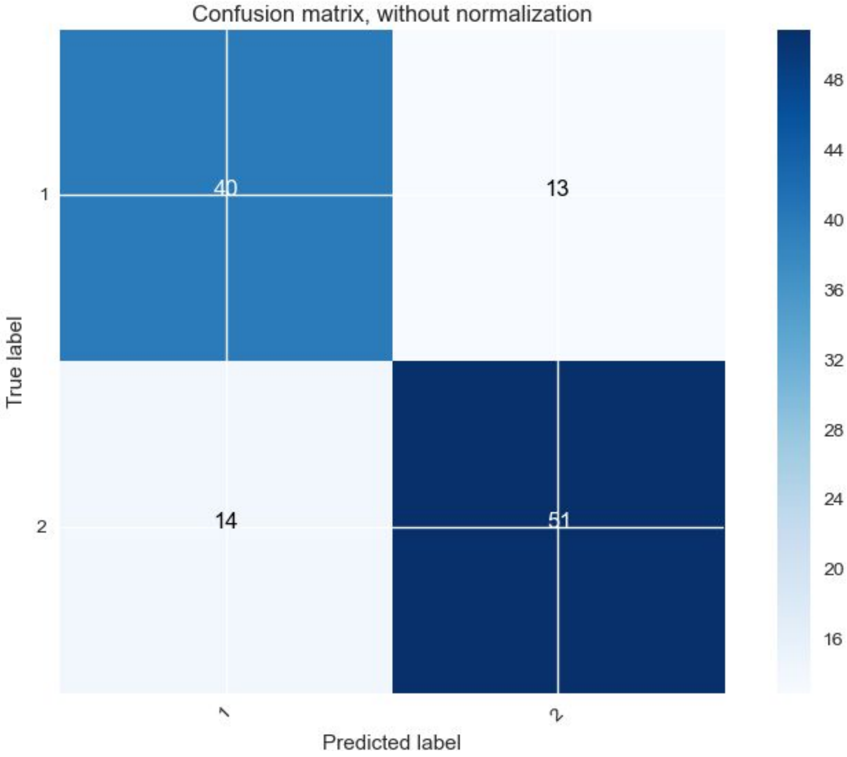
8. LDA compression and NB classification with 50 training Samples per class



9. NB Classifier on raw data



10. 10-fold cross validation



Accuracy Table

# Training Samples/Class	Method (Compressor + Classifier)	Accuracy %
5	NB on Raw Data	90.83
5	PCA with LDA	90
5	LDA with LDA	74.16
5	PCA with NB	91.6
5	LDA with NB	74.16
50	PCA with LDA	100
50	LDA with LDA	86
50	PCA with NB	100
50	LDA with NB	70
6	LDA with LDA 10-Fold Cross Validation	77.11

Results

Applying naive bayes with just 5 samples per class to train parameters gives good accuracy of slightly over ninety percent. Further applying PCA compression improves the classification accuracy both when doing LDA and NB classification on compressed data. But applying LDA compression over data and then doing classification deteriorates the accuracy. Doing a 10-fold classification of data using LDA compression and classification gives an accuracy of about 77 percent. As with other sections, PCA seems to be better for a dataset with small number of features like the wine dataset which has just 13 features per data point.

Discussion

One of the reasons PCA seems to outperform LDA for the wine dataset is because the dataset is very small and the number of samples we take are also less. This is something which has been shown experimentally by Martinez et al. for the face recognition dataset[1].

Another reason which possibly explains why LDA under performed for this dataset if because the variance of the two classes is not equal (which is an underlying assumption for LDA). Since, PCA is agnostic of variance for individual classes and it tries to find maximum variance components in complete dataset, it outperforms LDA.

[1] http://www2.ece.ohio-state.edu/~aleix/pami01.pdf

Multi-class Classification

In case of multiple class we can take a one vs all approach and the one which gives the best probability could be chosen. We can also consider applying Multi-class LDA which in most of the practical scenarios performs better than 2-class LDA (or one-vs-all scenario).