Fall 2016
CSCE 666 Pattern Analysis
Homework #2

Due date: 10/10/2016

In recognition of the Texas A&M University policies of academic integrity, I certify that I have neither given nor received dishonest aid in this homework assignment.

Name: Caio Duarte Diniz Monteiro Signature:

Problem 1 (25%)

For the part 1 of this problem, we were asked to compute the density estimation of a dataset using a Gaussian kernel density estimate. We should then plot the results for three different bandwidths (one considered small, one considered to be optimal, and one large) comparing each result with the normalized histogram of the data. Figure 1 presents the three graphs. Using a small bandwidth, the estimate has higher peaks but it is very jagged, indicating that more samples are needed in order to use such a short bandwidth. With the best bandwidth, the density estimate is much smoother but still able to capture the overall shape of the histogram with a good level of details. When using a large bandwidth, we lose most of the distribution shape, resulting in a very smooth curve with little resemblance to the actual distribution.

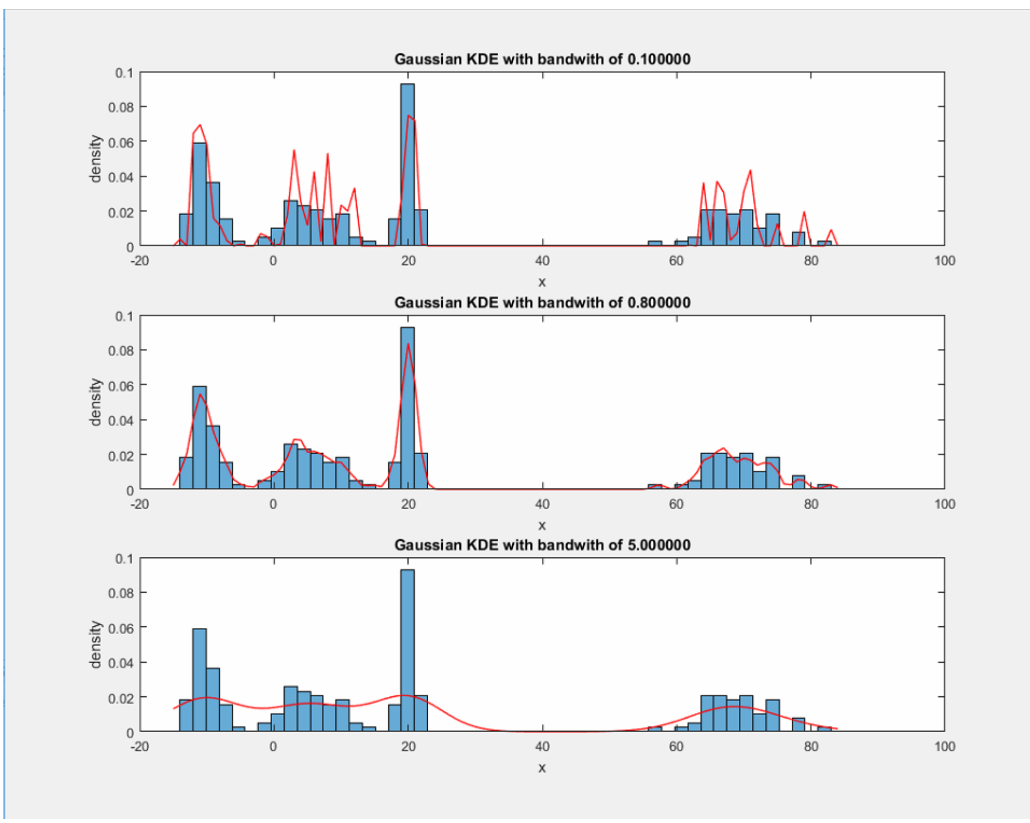

Figure 1. Resulting KDEs for bandwidths of 0.1 (top), 0.8 (middle), 5.0(bottom). The 0.8 bandwidth is able to properly capture the dataset density while it is much smoother than the curve with a bandwidth of 0.1.

Part 2 of the problem, requested us to compute the average log-likelihood for different bandwidths using the leave-one-out method. Bandwidths were chosen to be log-spaced between $h0/100$ and $100h0$, where $h0$ is the plug-in bandwidth estimate. Figure 2 presents the results for the multiple bandwidths tested, and the comparison between the best bandwidth (the one with the highest log-likelihood) and the plug-in estimate. As one can observe, the best bandwidth yields to a very similar result to the best one obtained in part 1. The plug-in estimate value is higher and thus, is not able to capture the details of the distribution.

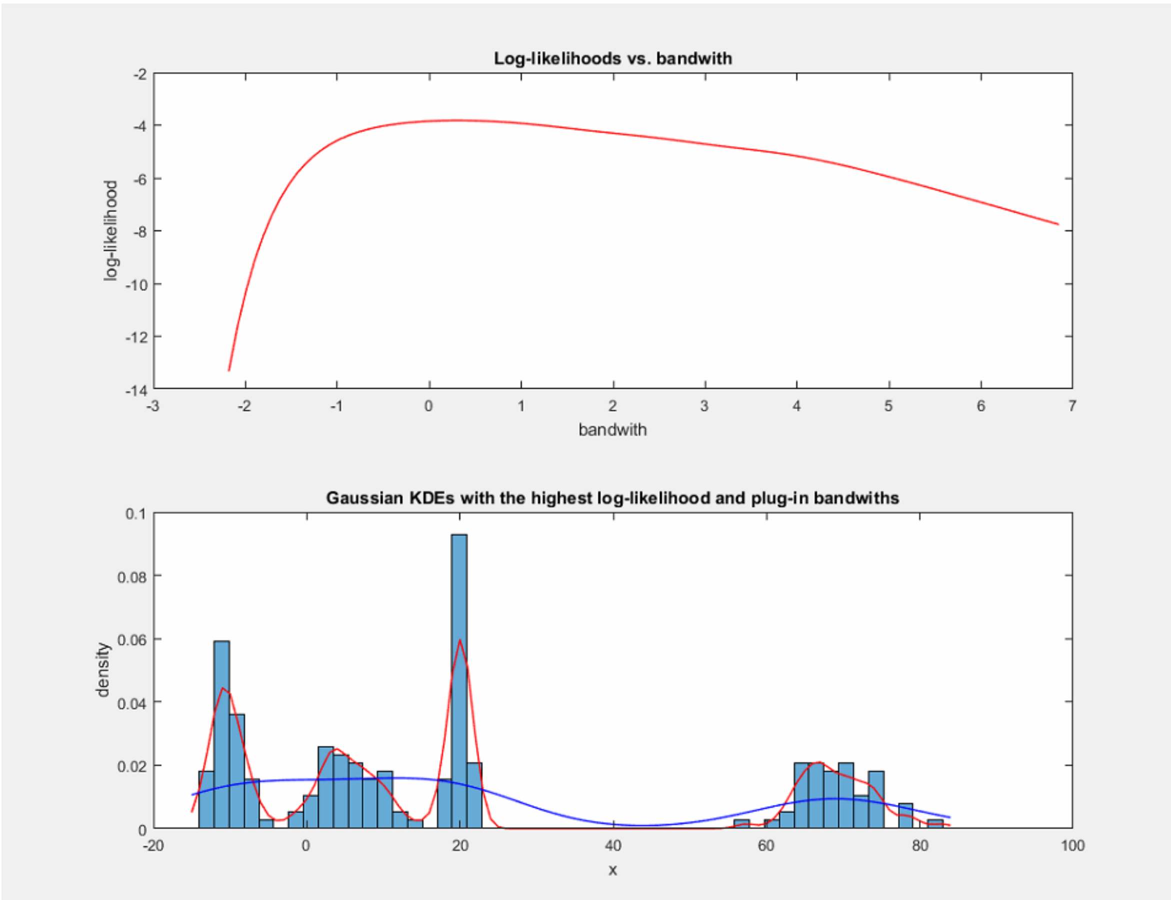

Figure 2. Top plot presents the log-likelihood vs. bandwidth curve, where the bandwidth axis is log-scaled. Bottom plot compares the results of the best bandwidth (red curve), with the plug-in estimate (blue curve).

Problem 2 (10%)

On problem 2, we should compute the principal components of a dataset and analyze which structures are observable on the new lower dimensional space. On the first experiment, a scatter plot of the first three dimensions of the original feature space is plotted. Figure 3 presents the result, as observable no distinct pattern can be found.

We then compute the principal components of the data and analyze how many components are required to represent the majority of the variance of the dataset. Figure 4 has two plots, the first one contains the eigenvalue of each component, and the second one the normalized sum of the first N components, both plots are log scaled on the *x* axis.

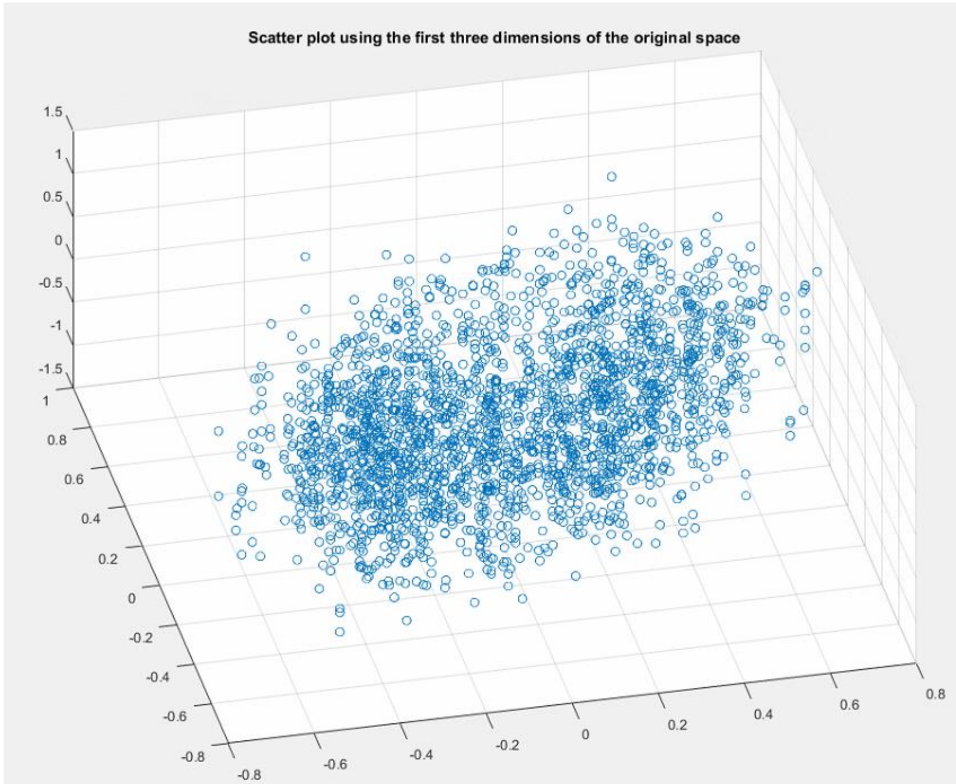

Figure 3. Scatter plot of the dataset considering the first three dimensions of the original feature space.

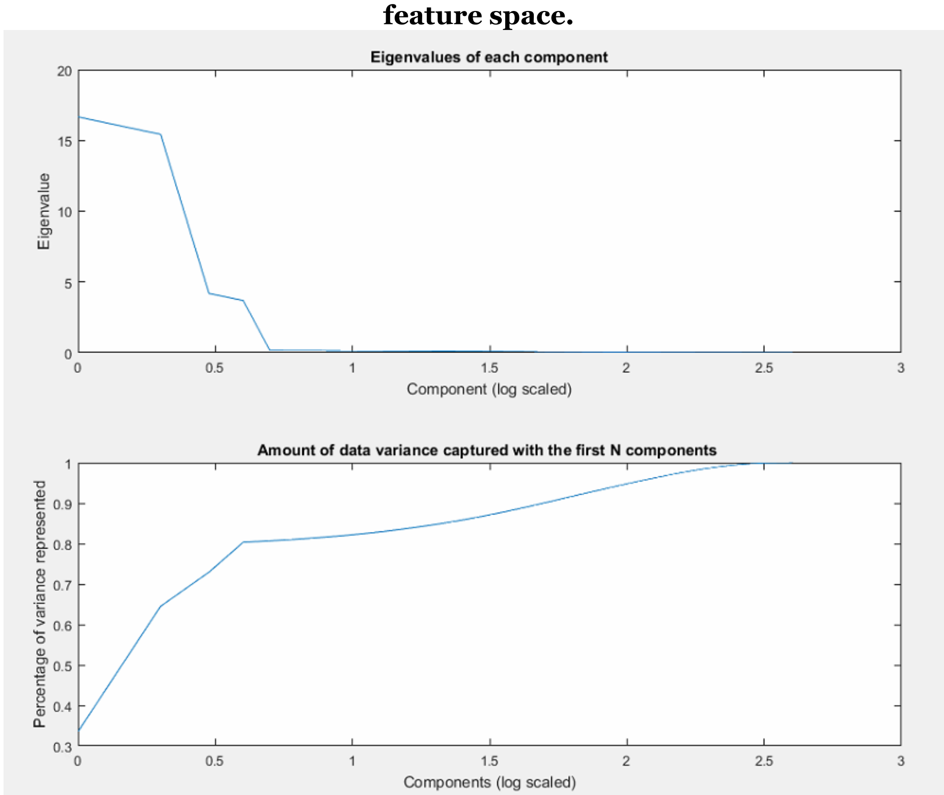

Figure 4. Top plot represents the eigenvalue of each component. Bottom plot represents how much data variance is captured using the first X components. Both plots are with the *x* axis log-scaled.

As shown in the second plot of Figure 4, the first 4 components of PCA are responsible for capturing around 80% of the dataset variance. Considering this info, we plotted 3-D scatters of different combinations of the first 4 components. Figure 5 presents the distinct structures found when using the first three components (the penguin on the top plot), and the second, third, and fourth components (the alien on the bottom plot).

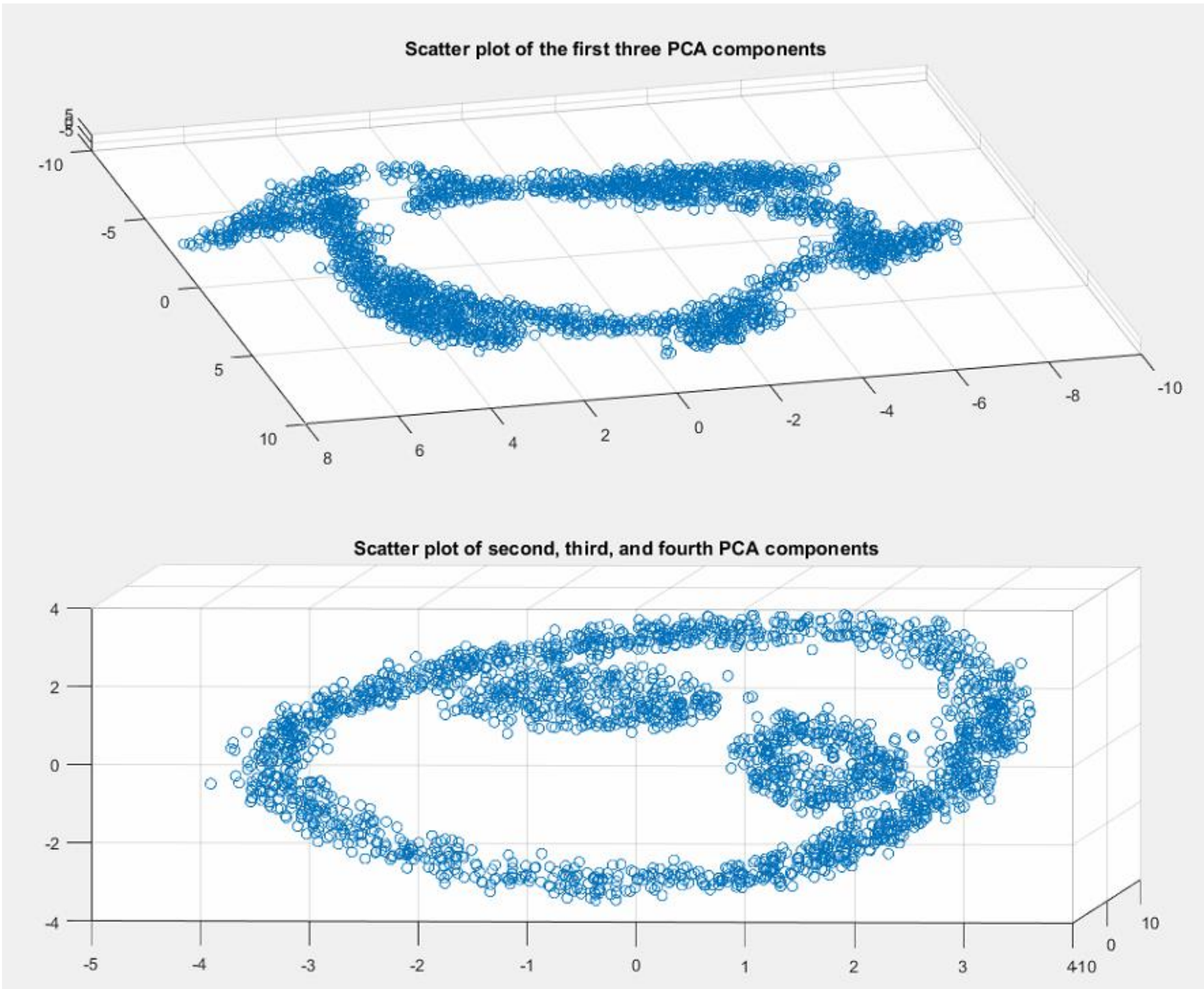

Figure 5. 3-D scatters using the first three components of PCA (top), and components 2, 3, and 4 (bottom). Distinct structures can be found.

Problem 3 (10%)

Problem 3 asked us to generate a three class dataset, where the feature space was made of 3 regular dimensions plus 48 dimensions of noise. Noise was generated based on the same distribution for all classes. After generating the dataset, we had to compute the PCA and LDA of it and compare their scatters. Since we have just three classes, LDA is able to generate at most 2 dimensions, so to keep the comparisons fair, we also used two dimensions of the PCA. Figure 6 presents the results. Because of the variance on the noisy channels, PCA is not able to properly separate the classes. Whereas LDA, that uses the mean to increase the projection class classification, achieves a much better separation.

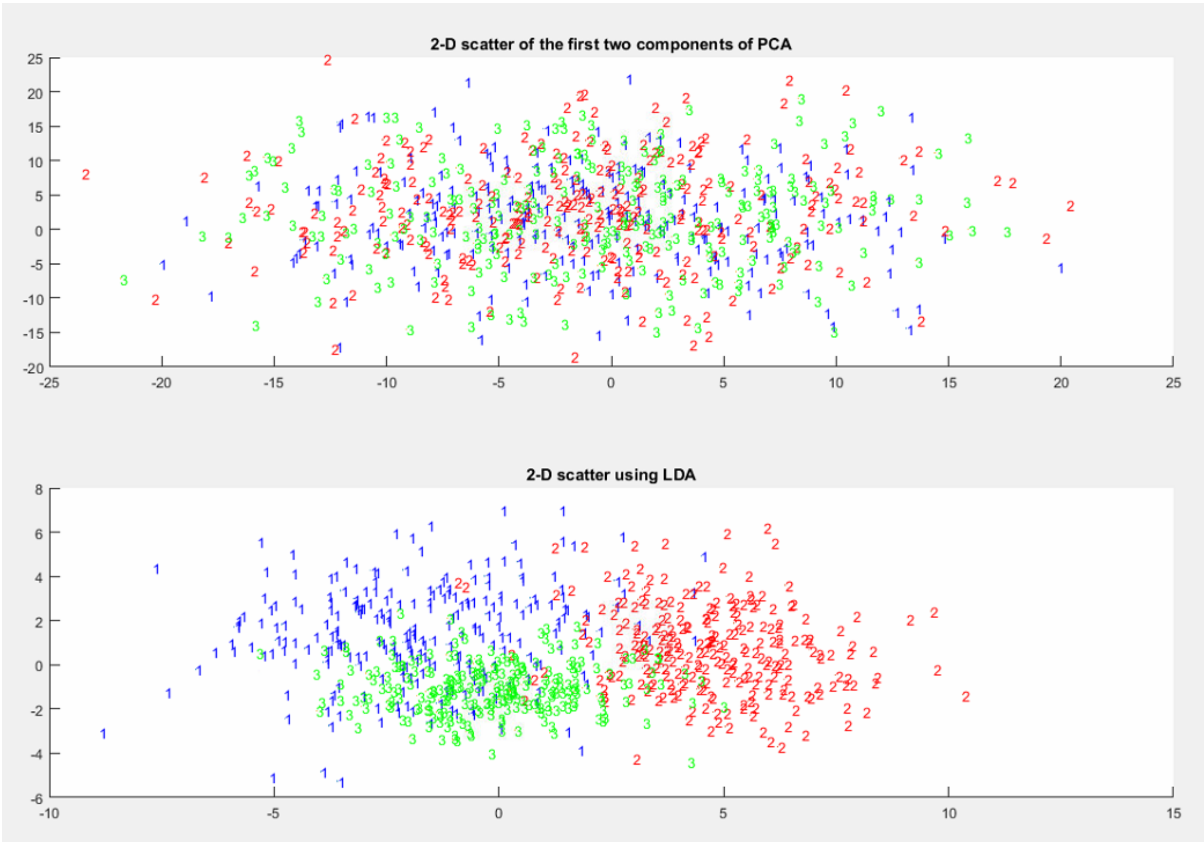

Figure 6. Comparison of 2-D PCA scatter plot (top) with LDA one. Class separation is much better using LDA.

Problem 4 (10%)

On Problem 4, we had to generate a dataset with the same characteristics of the one in Problem 3. Then we had to implement a Quadratic classifier and evaluate it on the generated dataset. To evaluate the classifier, we computed the average accuracy of 30 iterations. Experiments were made using the original feature space, a 2-D PCA space, and a 2-D LDA space. Since we knew from the last problem that a 2-D PCA did not perform very well in distinguishing classes, we plotted a 3-D PCA to check if the extra dimension would improve the class separations. Figure 7 has the 3-D scatters for the training and test sets in one of the iterations. Increasing it to three dimensions did not provide a much better separation, thus we continued to use the 2-D space for the classification task. Figure 8 shows the scatter of a training and test set in one of the iterations, as expected the results are very similar to the one in the previous problem. Table 1, has the accuracy averages for the 30 iterations on each method, as expected, LDA achieves higher accuracy than PCA, being just 8% below than the results in the complete feature space.

Table 1. Average accuracies for each one of the three feature spaces tested.

	Accuracy
Original feature space	84.9%
2-D PCA	35%
2-D LDA	77%

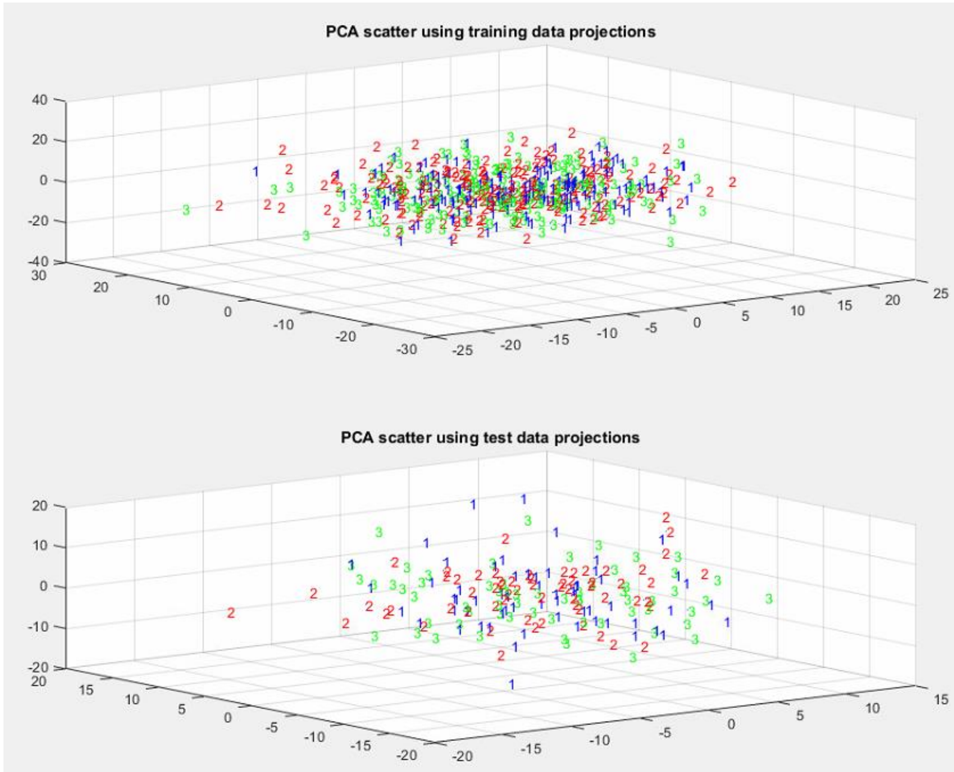

Figure 7. 3-D scatters of the training (top) and test (bottom) sets using the first three PCAs.

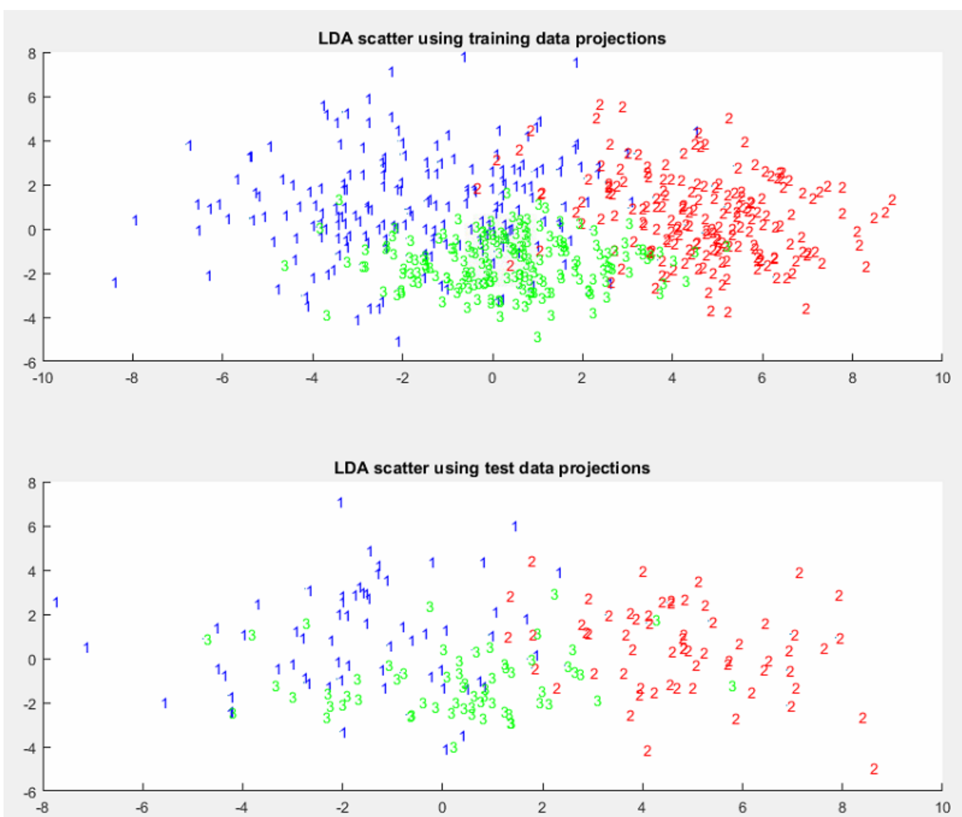

Figure 8. 2-D scatters of the training (top) and test (bottom) sets using LDA.

Problem 5 (10%)

Problem 5 asks us to perform the same experiments of Problem 4 replacing the Quadratic classifier with a KNN classifier. Figure 9 presents the results for the many values of K experimented. Accuracy remained almost the same as the one with Quadratic classifier when using LDA or PCA, but when considering the original feature space, the KNN performs much worse than the Quadratic version. This can be explained by the amount of noise that is going to interfere on the distances computations used in KNN. While increasing the value of K slightly improves the classifier performance for the original feature space and LDA up to a point, it starts to stabilize around k=100 and then decrease when K approximates the number of elements in the set. Results for the PCA suffer a little decrease when K goes up to 20, and then remains almost the same regardless of K, because it reaches chance probability.

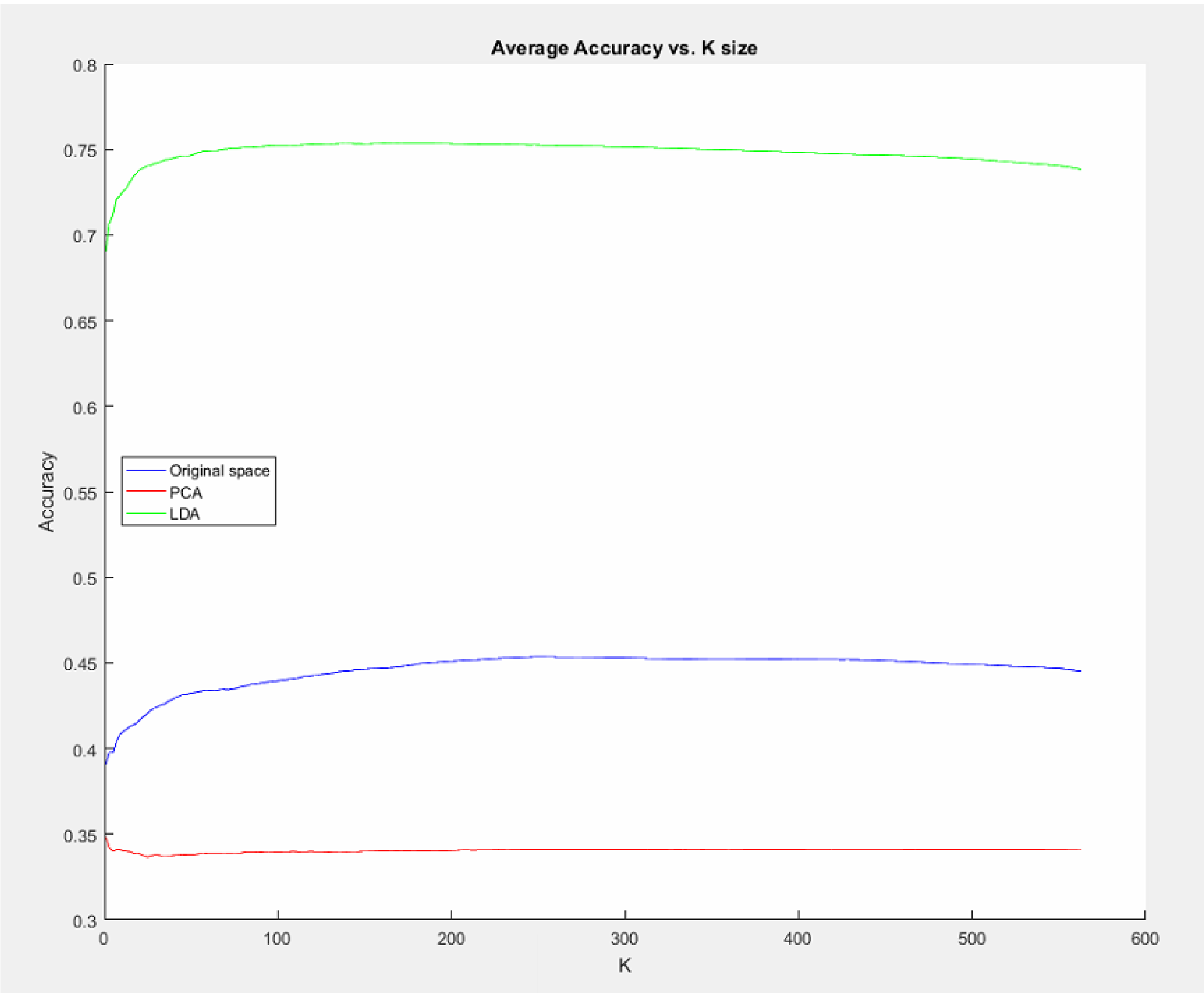

Figure 9. Average accuracies for 30 iterations on each feature space using a KNN classifier, with K evaluated from 1 to 563 (training data size minus 1) in increments of two.

Problem 6 (35%)

On Problem 6 we have to solve a classification problem that resembles the situation we would find in real world, where we are going to train and evaluate our classifiers on a particular training set, and then the classifier would be tested on 'real world' against data that was never seen before.

To decide on the approach that works best with this data, it was used the three-way-split validation. The whole training set was separated into training and validation sets, like we had on problems 4 and 5, and the test set was just used to evaluate the predicted performance of our classifier.

As a first step, it was computed the PCA and LDA of the entire training set, and evaluated how many dimensions were necessary to properly represent the data in each approach. Figure 10 presents the results when using PCA and Figure 11 when using LDA. As observable, on the PCA scatter plot, it performs very poorly in separating the classes, and to represent around 80% of the dataset variance we need the first 34 dimensions (out of 100 on the original space). In the other hand, LDA is able to achieve a similar level of representativeness with just 1 dimension.

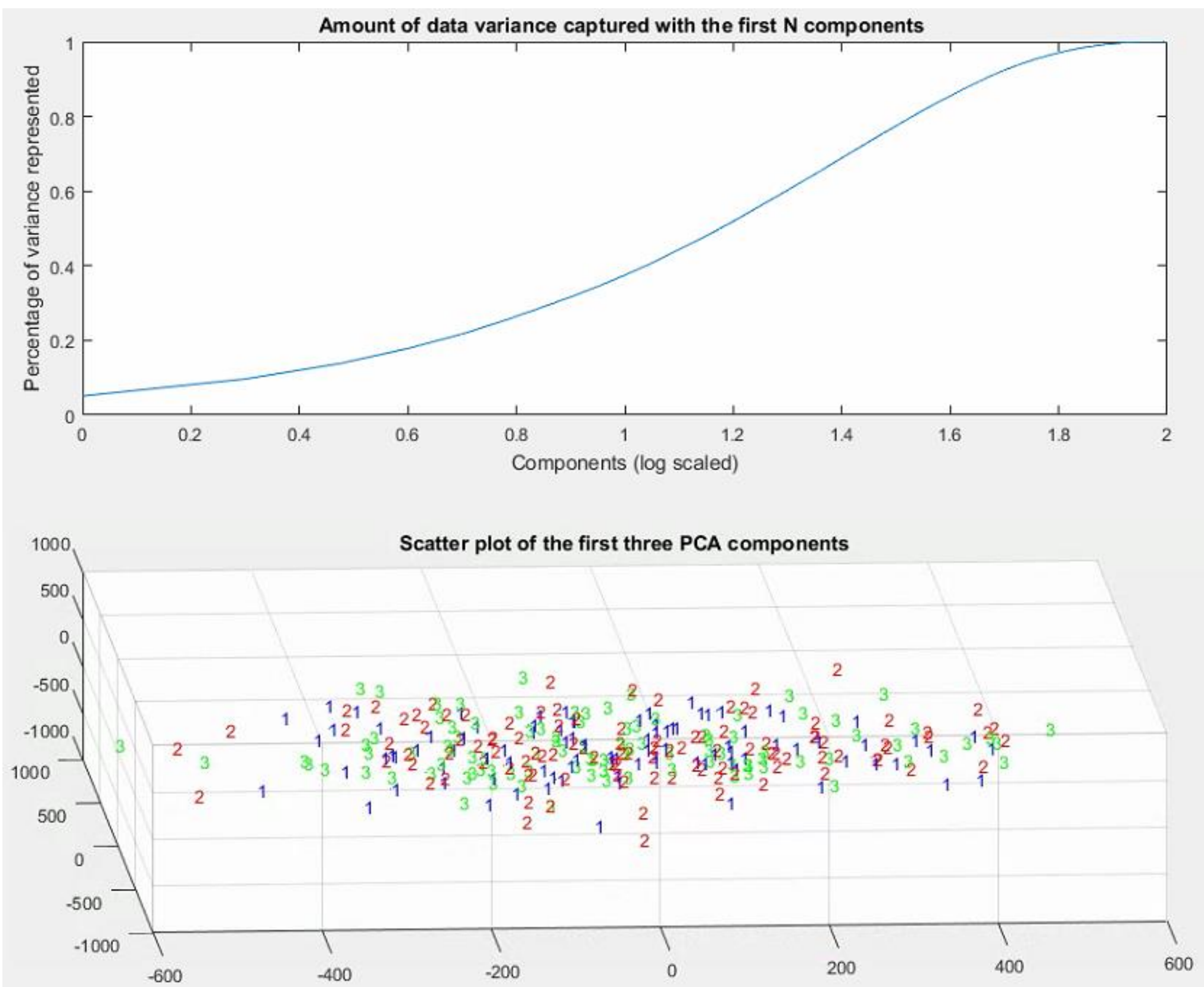

Figure 10. Amount of data variance captured on the first X PCA components (top), and the scatter plot using the first three dimensions of it.

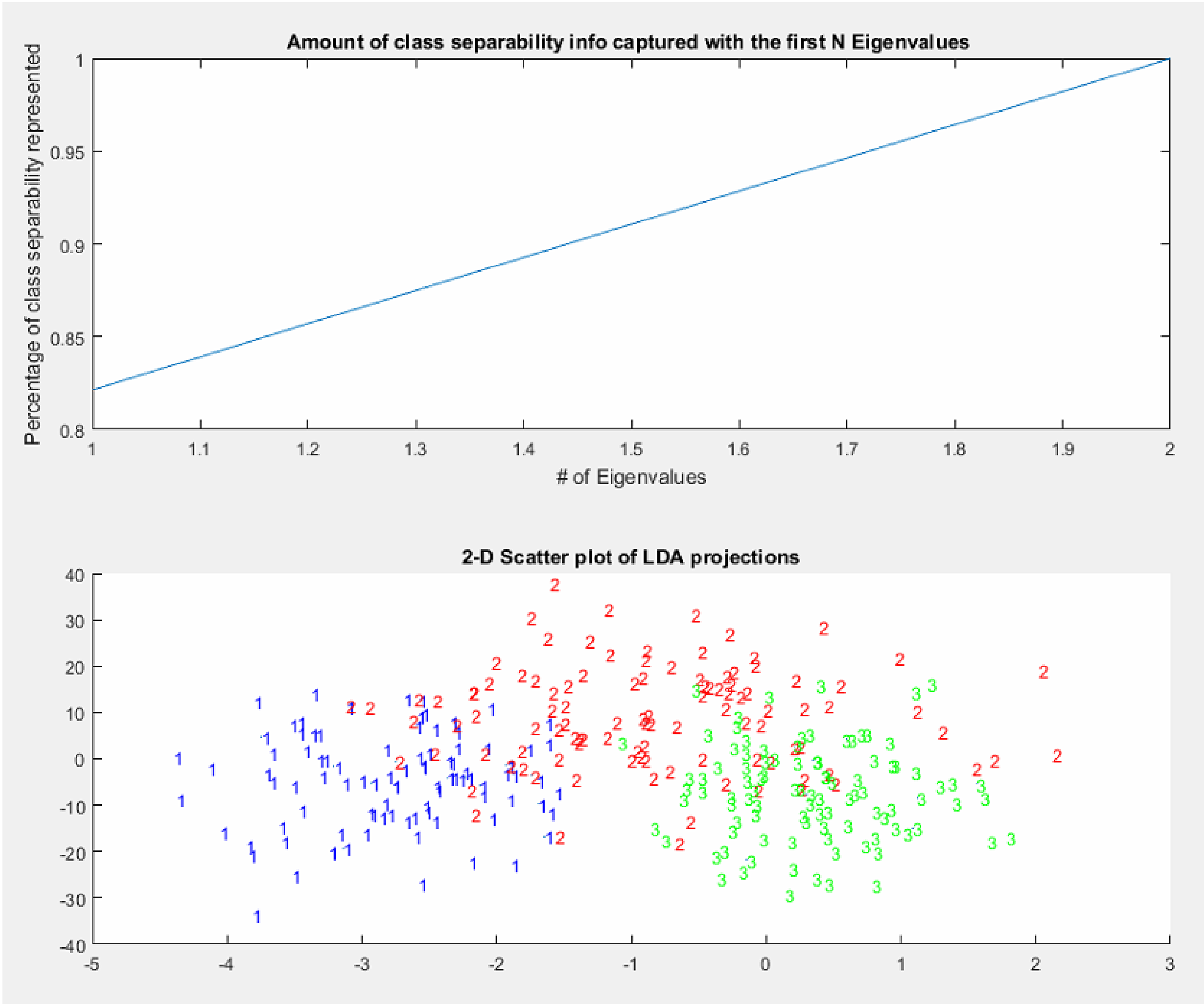

Figure 11. Amount of possible class separability captured with the first X eigenvalues (top), and the scatter plot using the two LDA dimensions.

Once it was examined that LDA offered a better class separation than PCA, it was then computed the average accuracies for LDA and the original feature space on both Quadratic and KNN classifiers. The average was computed on 30 iterations, with the whole training set being split in training and validation sets using a 75%/25% ratio. On the experiments the Quadratic classifier using LDA obtained the best results, around 52%. Table 2 summarizes the results for each combination.

Table 2. Average accuracies for the combinations of classifier and feature space experimented.

	Quadratic classifier	KNN classifier (K=11)
Original feature space	33.6%	37.2%
LDA	52.1%	47.8%

After validating the best approach to be used for the classification task, we then trained our Quadratic classifier on the whole training set and test it on the hw2p6_test.mat dataset. Obtaining an exact 54% of accuracy, really close to what we predicted on the validation phase. Figure 12 contains the transposed assigned labels column vector for the data on the test set.

```
uclab = [
  Columns 1 through 23

     1     2     3     1     3     2     2     2     1     1     3     3     3     2     1     1     3     1     3     2     2     2     1

  Columns 24 through 46

     1     3     1     1     3     1     1     1     3     1     3     1     2     2     2     1     1     2     3     2     3     3     2

  Columns 47 through 69

     1     3     1     3     3     3     1     1     2     2     2     1     1     1     2     1     3     1     3     2     1     2     2

  Columns 70 through 92

     1     2     1     1     3     2     3     2     2     2     2     1     2     1     1     3     2     2     2     3     2     1     3

  Columns 93 through 100

     1     1     2     3     3     1     2     1

];
```

Figure 12. Transposed column vector with the assigned classes for the entries in the test set.