GE 461 PROJECT 2

Doruk Barkın Durak

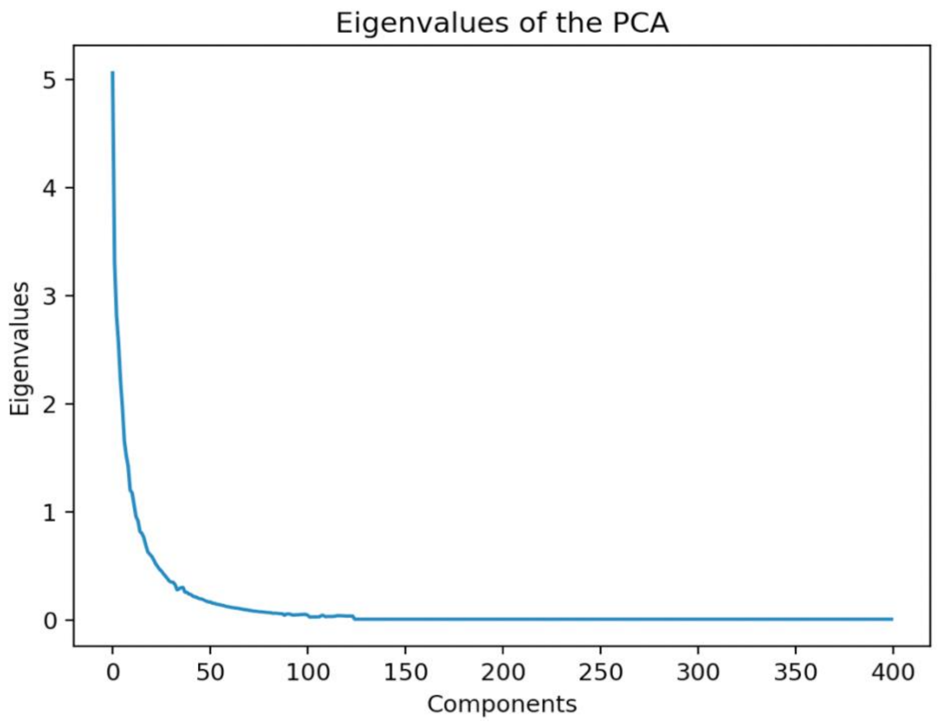
21802255

Industrial Engineering

Question 1:

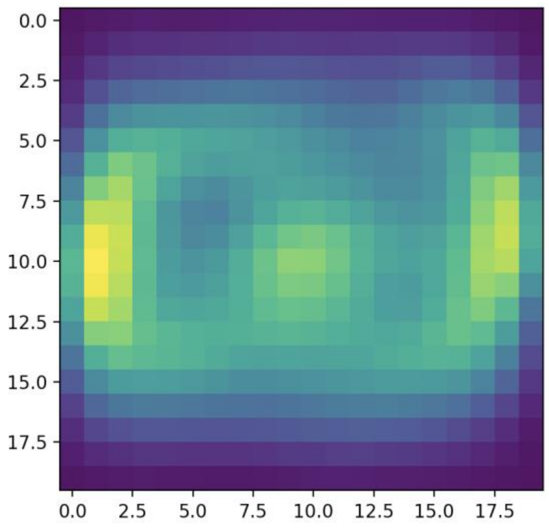
1) Required coding can be found on GE461P2Q1 python file with explanatory notes on code. I used Visual Studio Code for coding. I divided the data to half and completely random using train_test_split function.

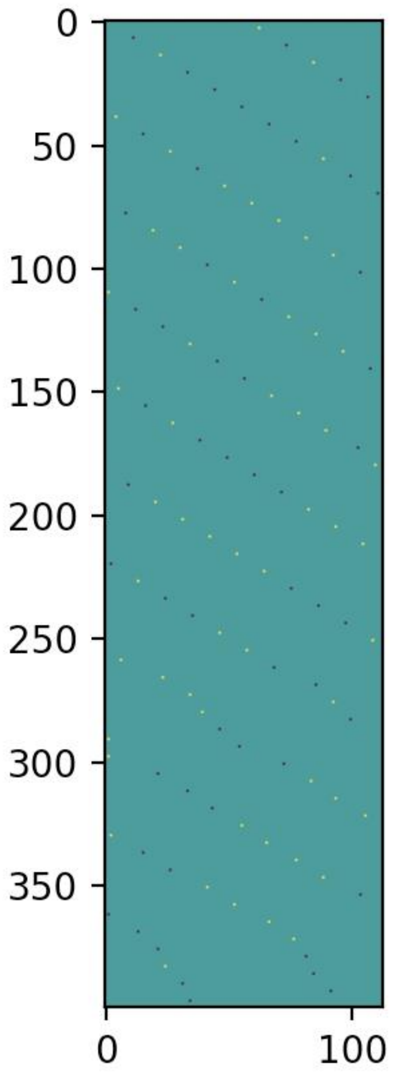
   For 113 components, this is the graph for the eigenvalues of PCA. Looking at this graph,



   Approximately at 50 components, eigenvalues become 0 at data. I would chose a number between 20-50.

2)

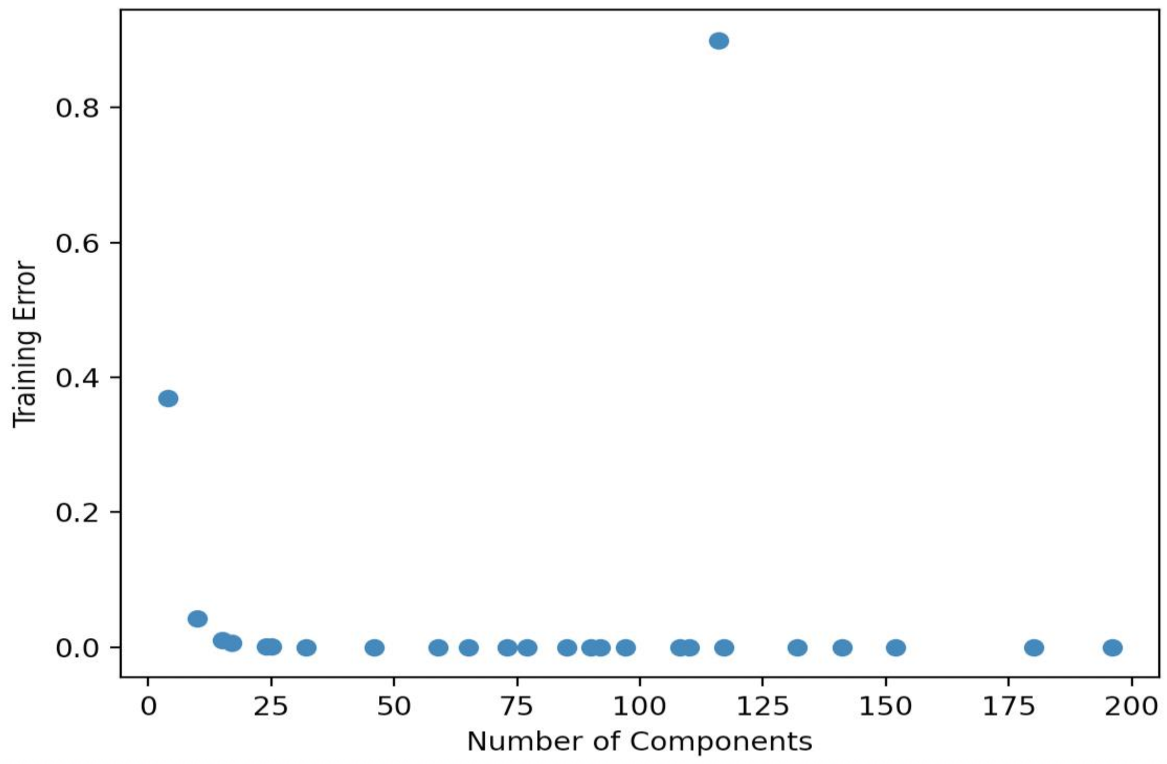


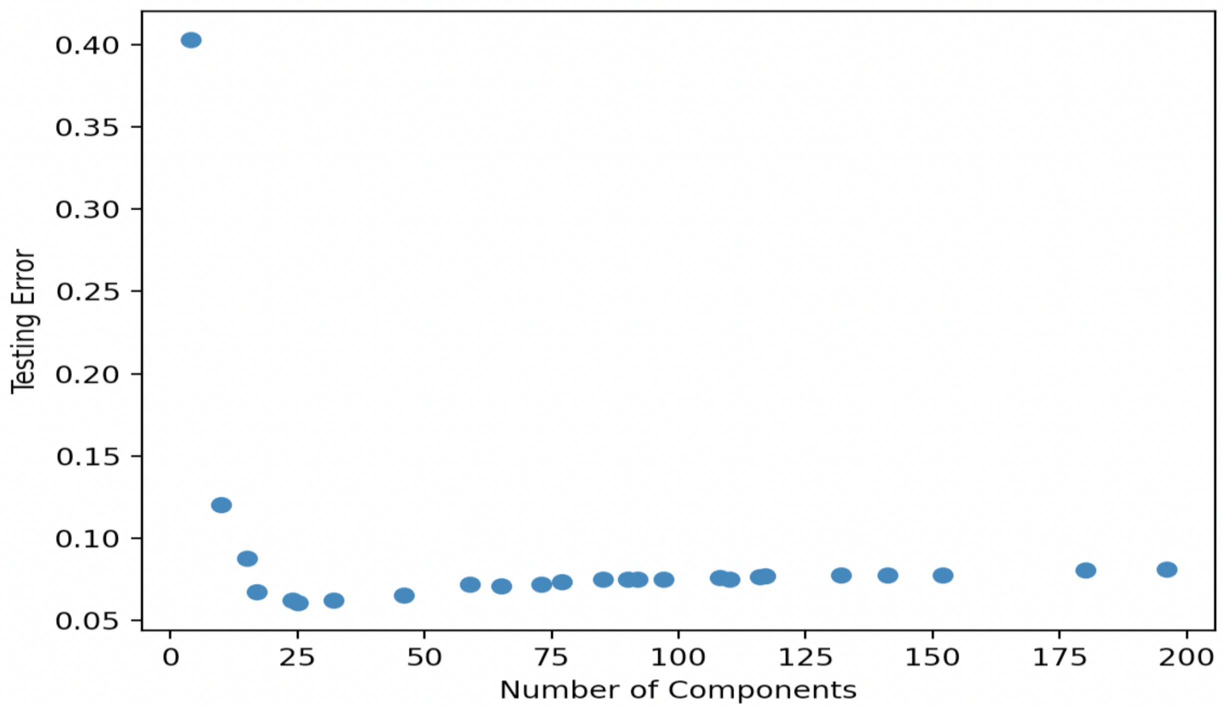


First picture is the sample means for 113 components and second is the eigenvectors for 113 components. From sample means we cannot say a certain thing and from eigenvectors we cannot say a certain thing also.

3) Used numbers and rates can be found excel sheet called ‘önemli data’. Used coding can be found by following the explanatory notes.

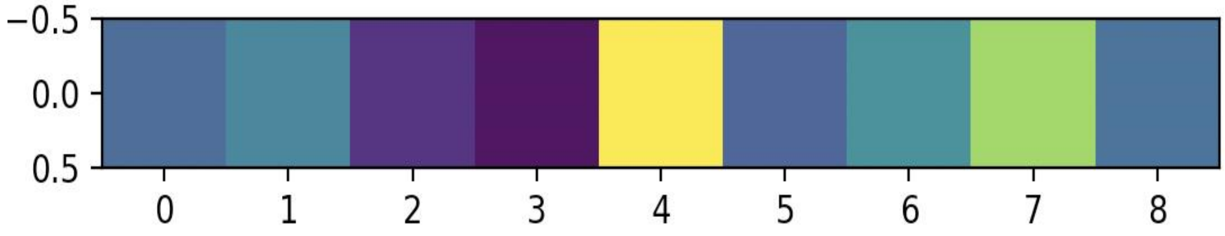
4) By looking at the graphs, training error seems to be stabilizing at 25 while test error stabilizes approximately at 50. This means that around 50 subspaces we should be fine. Charts can be found below.

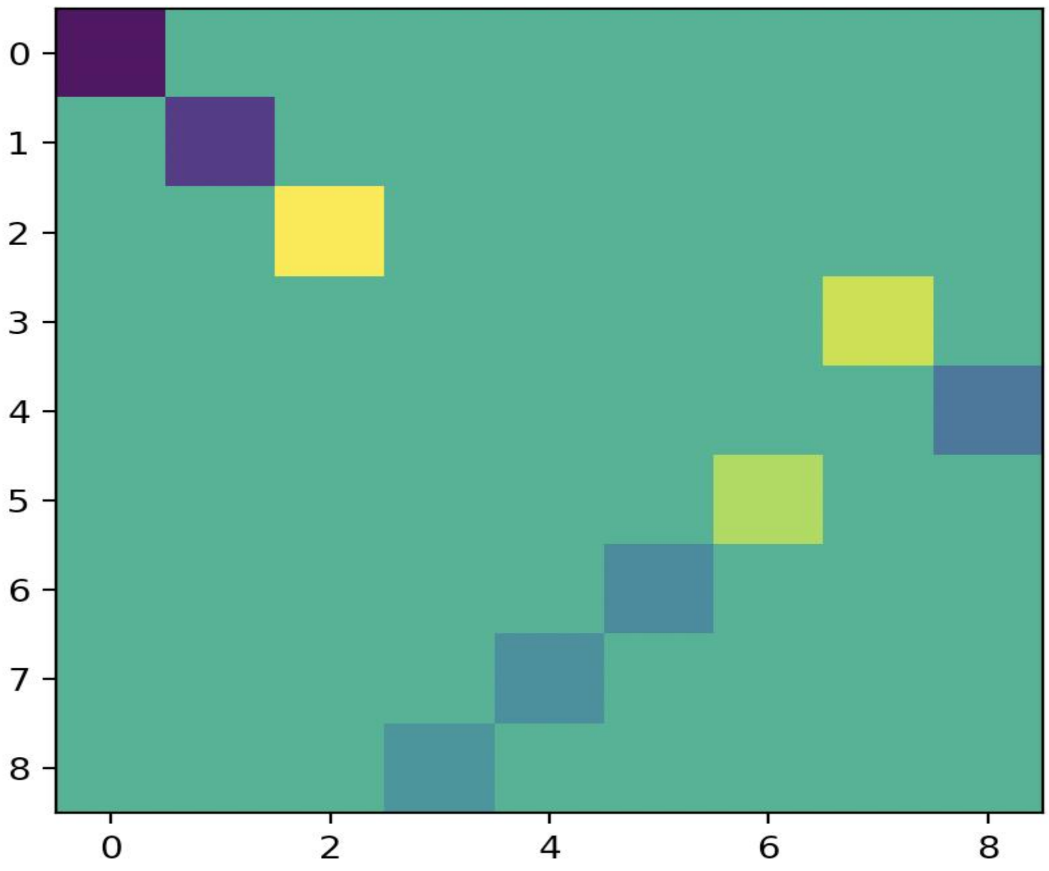




Question 2:

1) For 9 lairs. Sample mean and eigenvector charts are as follows:

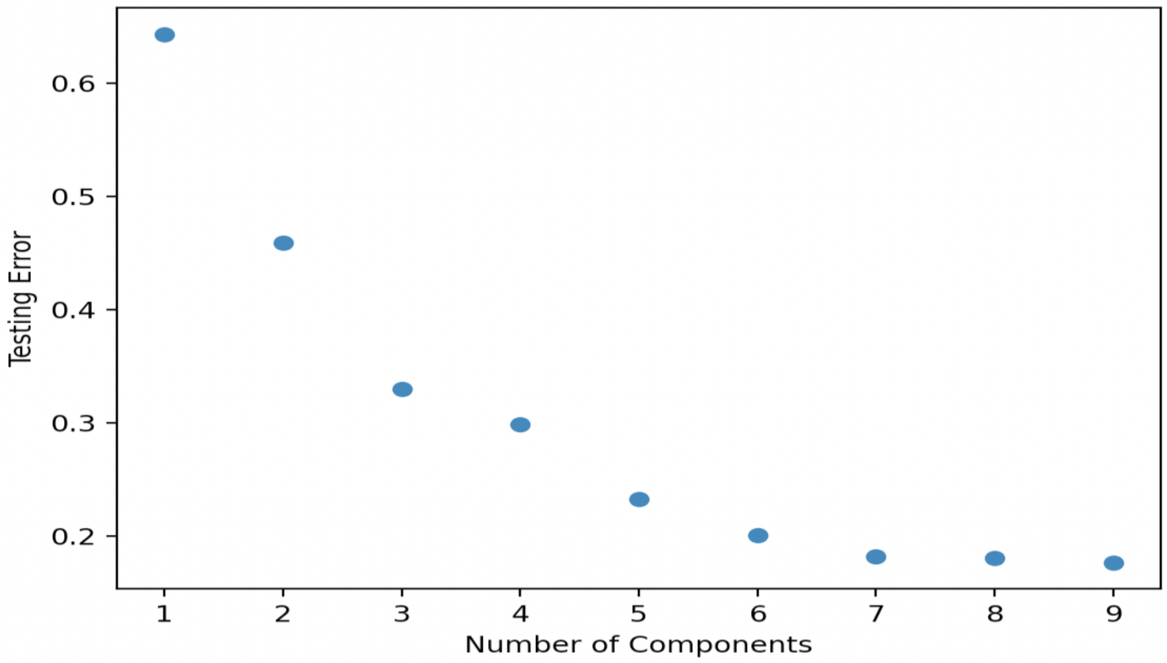




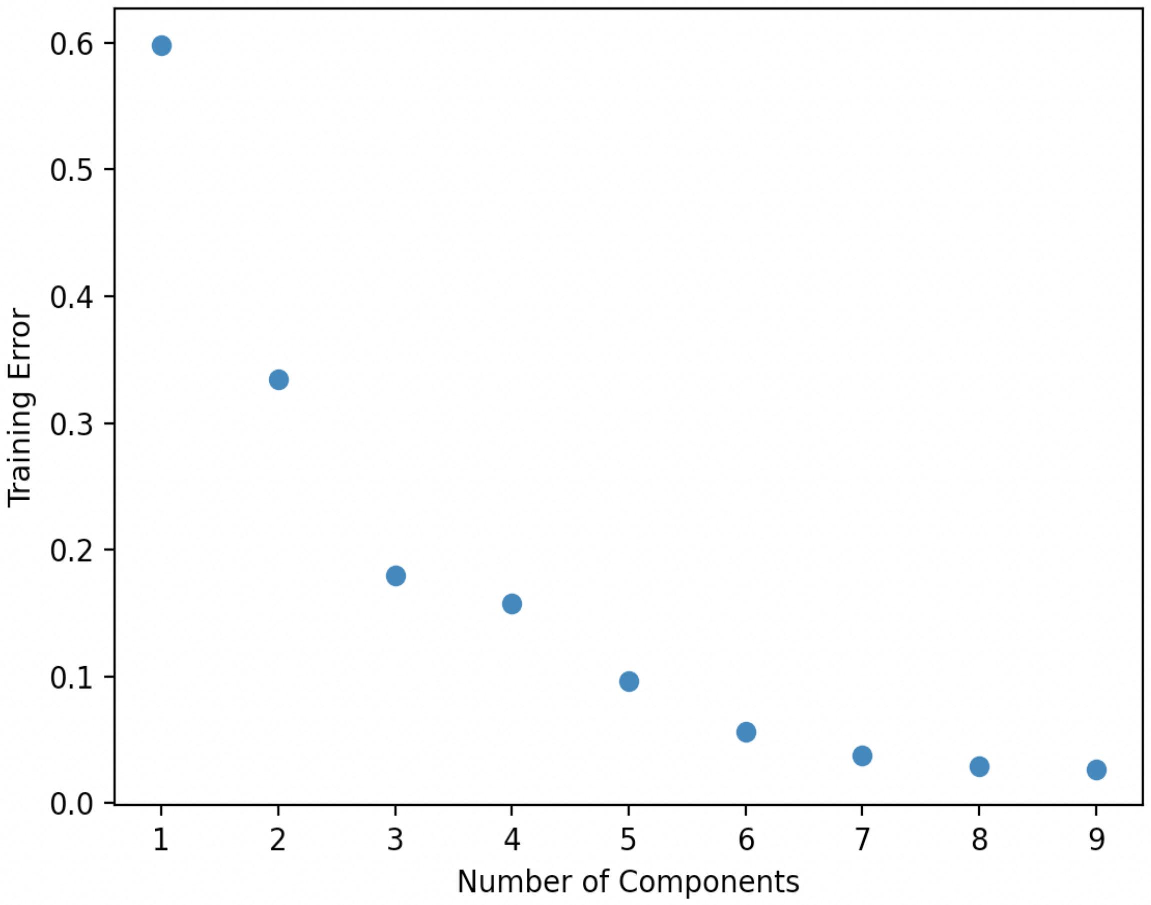
In this graph, I can see a pattern reoccurring at 6th component. My expectation is a stabilization at 6th component and after. Before the 6th component data may not be enough for visualization.

2) Required data can be found on excel file called ‘önemliLDA’. Used measures can be found on coding by reading explanatory notes.

3



Testing Error
0.6
0.5
0.4
0.3
0.2
1
2
3
4
5
6
7
8
9
Number of Components



Training Error
0.6
0.5
0.4
0.3
0.2
0.1
0.0
1
2
3
4
5
6
7
8
9
Number of Components

My expectations were correct because after the 6th component, training and testing data stabilizes at followingly 0.03 and 0.18 percent. Comparing this error rates with PCA method, I can say that using PCA method for this kind of data is better than using PCA method because in PCA we can negate the effects of dimensionality more than LDA method.

References:

1) https://scipy.org ,Python scipy.
2) https://scikit-learn.org/stable/ , Scikit-Learn
3) https://pandas.pydata.org , Pandas library
4) https://matplotlib.org , Matplot library
5) https://numpy.org , Python Numpy library
6) https://seaborn.pydata.org , Seaborn.pydata library
7) https://machinelearningmastery.com/linear-discriminant-analysis-for-dimensionality-reduction-in-python/
8) https://towardsdatascience.com/fishers-linear-discriminant-intuitively-explained-52a1ba79e1bb
9) https://github.com/sthalles/fishers-linear-discriminant/blob/master/LDA.ipynb
10) https://towardsdatascience.com/how-to-split-a-dataset-into-training-and-testing-sets-b146b1649830
11) https://machinelearningmastery.com/train-test-split-for-evaluating-machine-learning-algorithms/
12) https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html
13) https://towardsdatascience.com/pca-using-python-scikit-learn-e653f8989e60
14) https://www.datacamp.com/community/tutorials/principal-component-analysis-in-python
15) https://www.blogforbrains.com/blog/2014/9/6/loading-matlab-mat-data-in-python