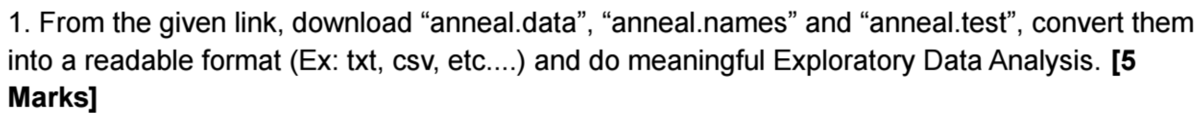
PRML LAB REPORT 7

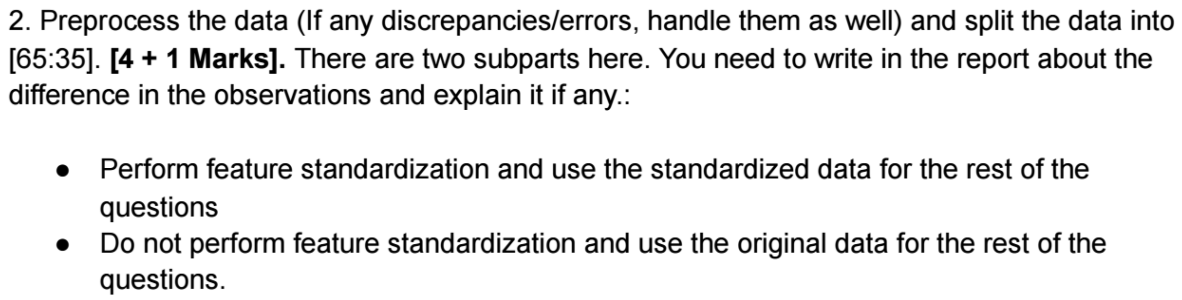
MANISH(B21CS044)

QUESTION 1



1. From the given link, download “anneal.data”, “anneal.names” and “anneal.test”, convert them into a readable format (Ex: txt, csv, etc....) and do meaningful Exploratory Data Analysis. **[5 Marks]**

In this part I have performed naming of the file **anneal.data** named the columns according to the file **anneal.names** and make the data meaningful to use by dropping the columns having most of the entries as **Nan**.



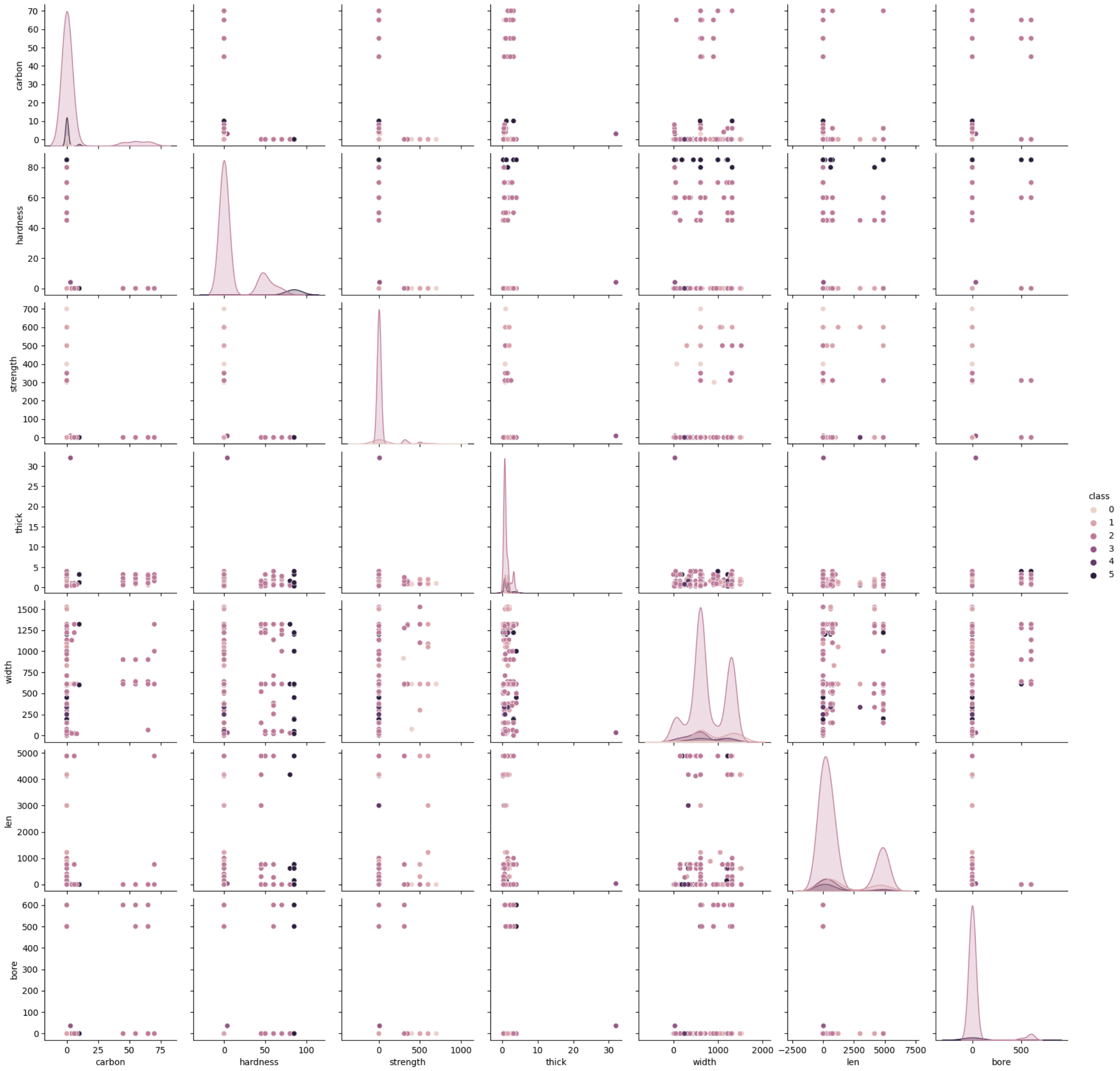
2. Preprocess the data (If any discrepancies/errors, handle them as well) and split the data into [65:35]. **[4 + 1 Marks].** There are two subparts here. You need to write in the report about the difference in the observations and explain it if any.:

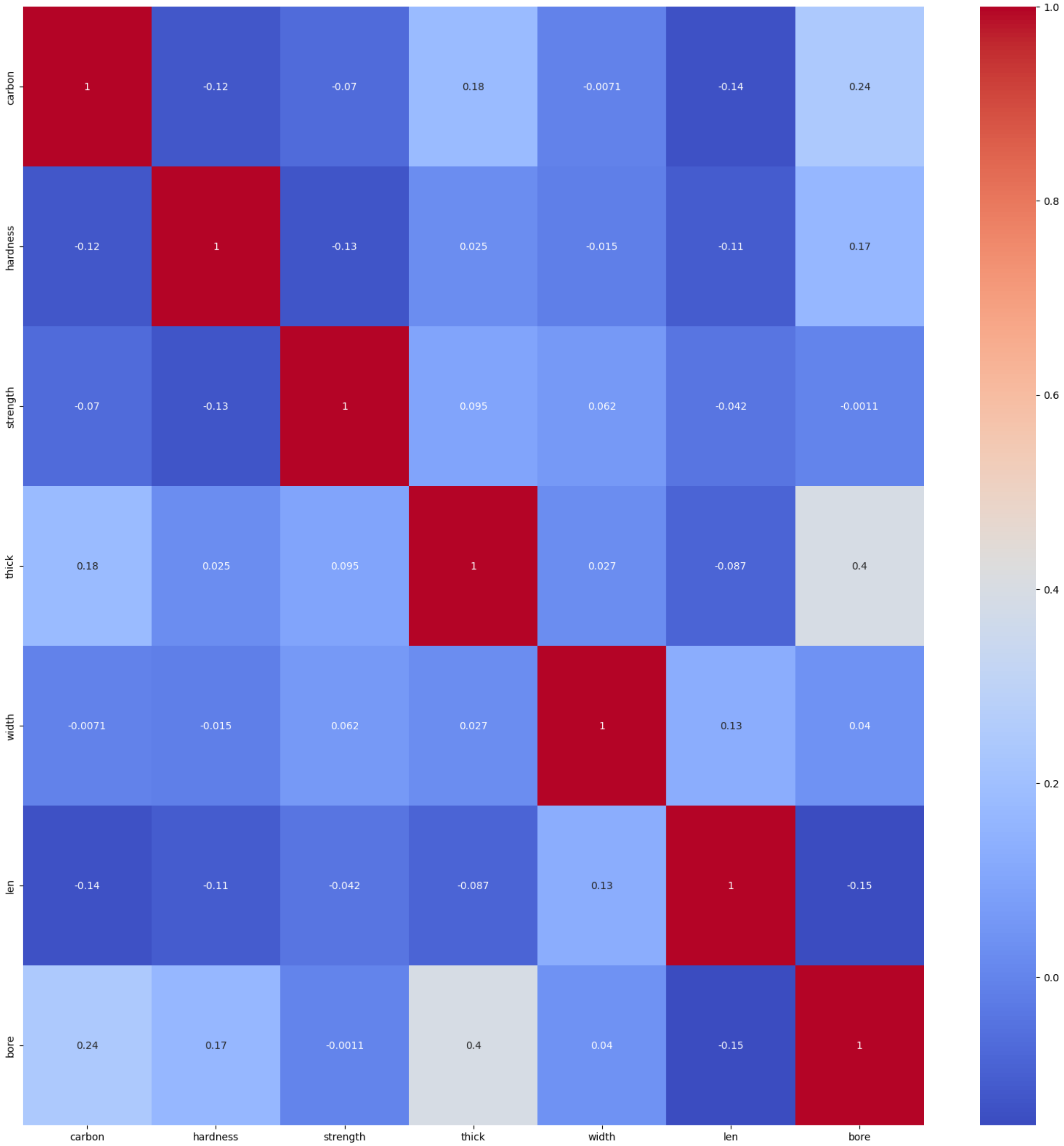
- Perform feature standardization and use the standardized data for the rest of the questions
- Do not perform feature standardization and use the original data for the rest of the questions.

In this I have performed the preprocessing of the data using **Label Encoder, Get dummies and Standard Scalar** for ordinal encoding , categorical encoding and Standardizing the data.

I have kept both datasets one with Standardizing and without Standardizing. Also, performed a train test split for both datasets.
Below given plots are the plots for the dataset without Standardization.

Here are the plots for visualization of the data:


carbon
hardness
strength
thick
width
len
bore
0
10
20
30
40
50
60
70
0
20
40
60
80
0
100
200
300
400
500
600
700
0
5
10
15
20
25
30
0
250
500
750
1000
1250
1500
0
1000
2000
3000
4000
5000
0
100
200
300
400
500
600
0
25
50
75
0
50
100
0
500
1000
0
10
20
30
0
1000
2000
−2500
0
2500
5000
7500
0
500
class
0
1
2
3
4
5



3. Train 2-3 Classification Models (studied and implemented so far out of which one has to be *SVM* classifier) with the proper reasoning of choosing them and showing 5-Fold Cross-Validation Plots as well for comparison. **[5 + 5 Marks]**

In this part for classification I have used 3 classifiers: Decision Tree, SVM and KNN classifiers.

Because the dataset is not following normal distribution and continuous in certain columns that's why I have not used **Naive Bayes**.
Since,**Decision Tree** can work on any type of dataset and performs well on discrete data. I have performed the 5-fold cross validation and Decision Tree is significantly performing better than SVM and KNN as we can see in the graph also. And also, It doesn't show any change on standardizing Data.

Since the **SVM** performs better on the continuous data so it is performing better on Standardized data.

KNN is performing better on the standardized data because it is a distance based algorithm and it is performing better on the data which is in the same scale. And, also **KNN** works better on small Dataset that's why I have used KNN.

Here are the scores for the same:
5-fold score of Decision Tree Classifier without standardization is: 0.9072327044025158

5-fold score of SVM Classifier without standardization is: 0.7619025157232704

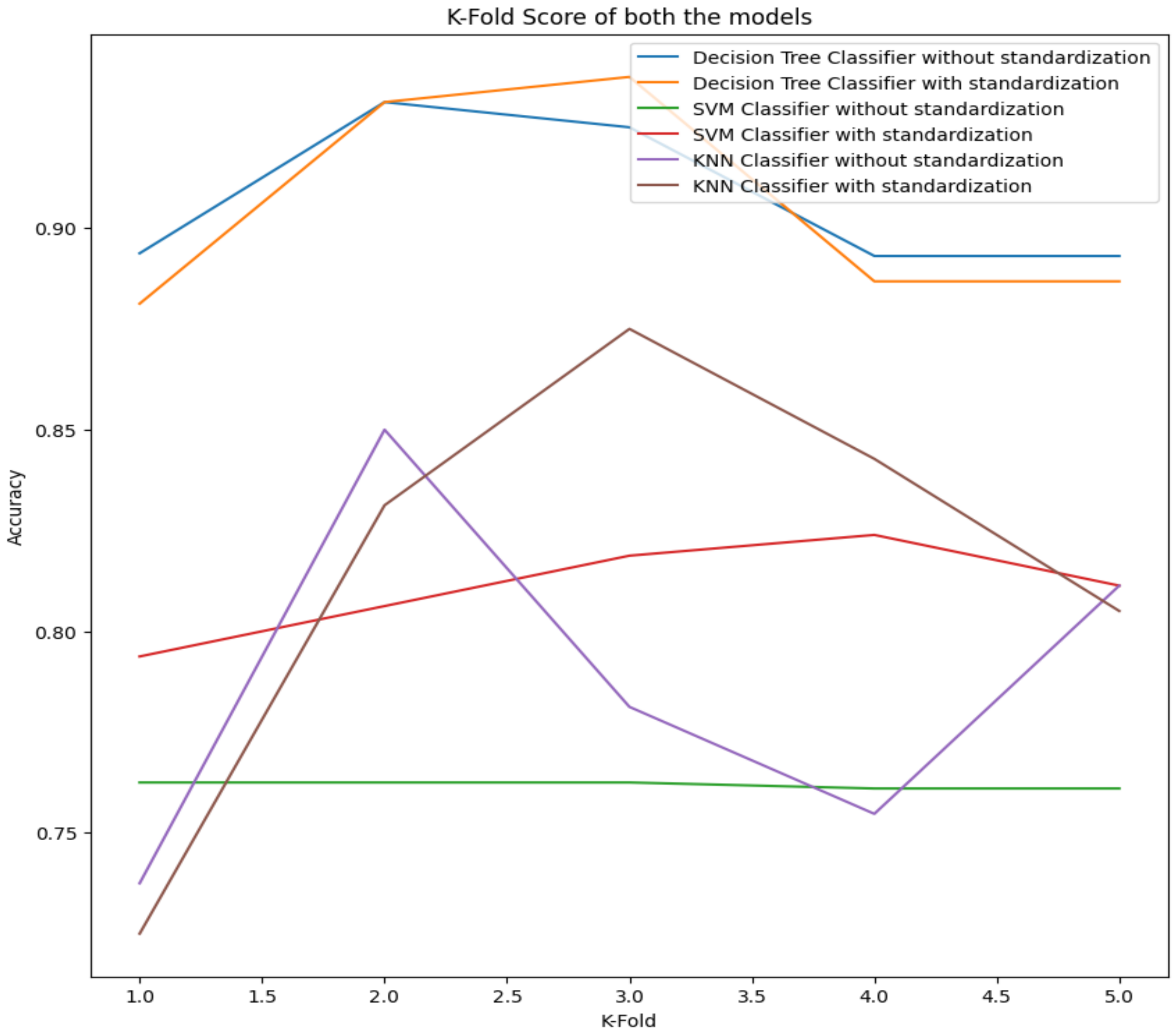
5-fold score of KNN Classifier without standardization is: 0.7869575471698114

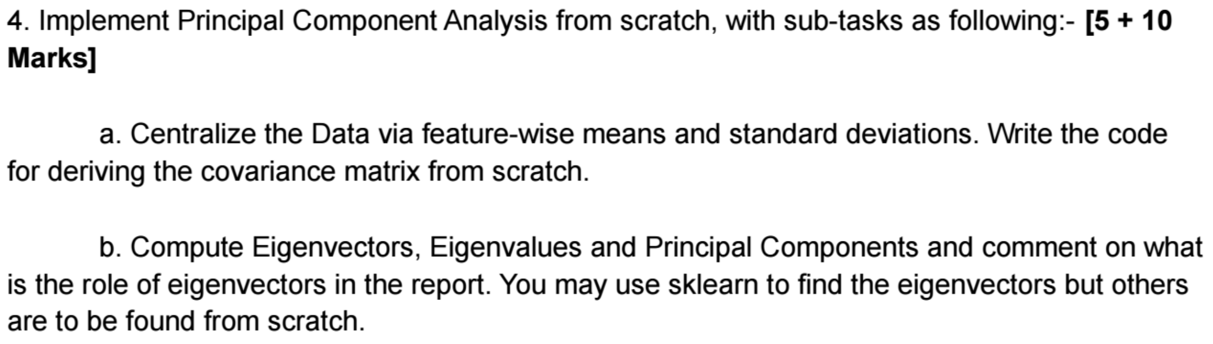
5-fold score of Decision Tree Classifier with standardization is: 0.9047169811320754

5-fold score of SVM Classifier with standardization is: 0.8107940251572326

5-fold score of KNN Classifier with standardization is: 0.8158097484276728

Here is the plot for the Comparison of the K-fold for all classifiers for both Standardized and Non-Standardized Data:




4. Implement Principal Component Analysis from scratch, with sub-tasks as following:- **[5 + 10 Marks]**

a. Centralize the Data via feature-wise means and standard deviations. Write the code for deriving the covariance matrix from scratch.

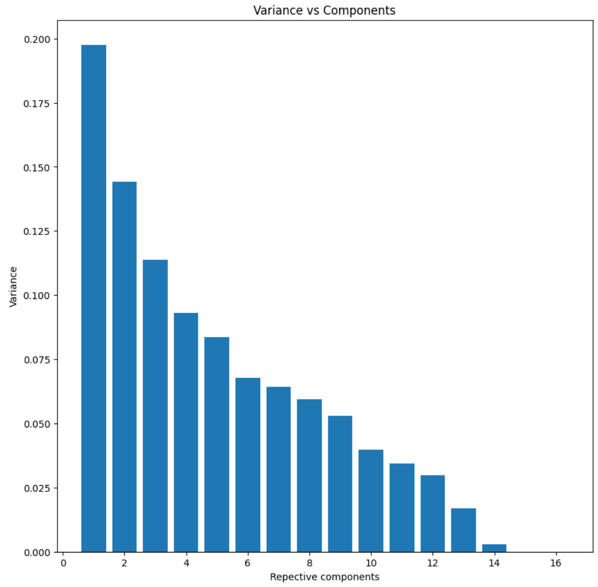
b. Compute Eigenvectors, Eigenvalues and Principal Components and comment on what is the role of eigenvectors in the report. You may use sklearn to find the eigenvectors but others are to be found from scratch.

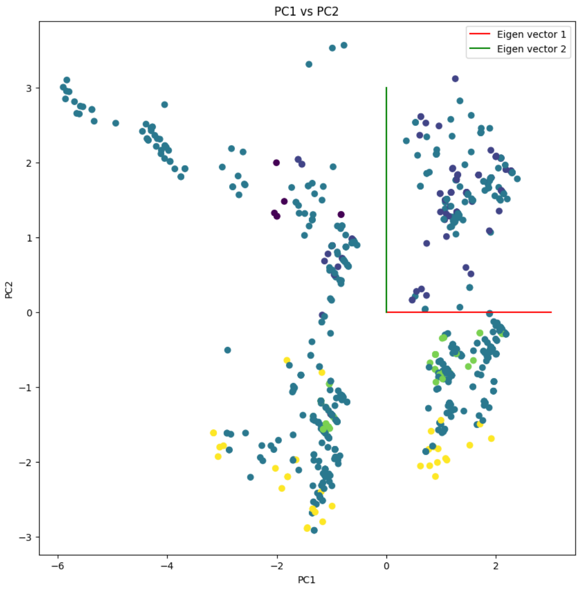
In this part I have implemented the PCA and also implemented the covariance Matrix of the data's function from Scratch. Also, set the default value of **PC's as 2** but the user can give input as it wants.

PCA reduces the dimensionality of the data according to the Variance explained by the Features. So, the role of **eigenvalues** is that it represents how much variance can be explained using the respective **PC.**
The role of **eigenvalues** is that it transforms the data according to the **PC** which explains most variance By projecting the data points using dot product of the **eigenvectors** and the original data.

5. Use the above-made PCA to reduce the data upto a chosen dimension/principal-components. Plot a bar graph to show the change in variance as you increase the no. of components. Along with this, plot a scatter plot to show the direction of the eigenvectors along with the data points(you may choose any 2 features among the reduced dataset). **[2+3 Marks]**

In this part I have performed the **PCA** implemented and Here are the plots for the respective **PC** and **Variance Explained by that PC** and also the plot between the **PC1 , PC2** along with their respective **Eigenvectors:**





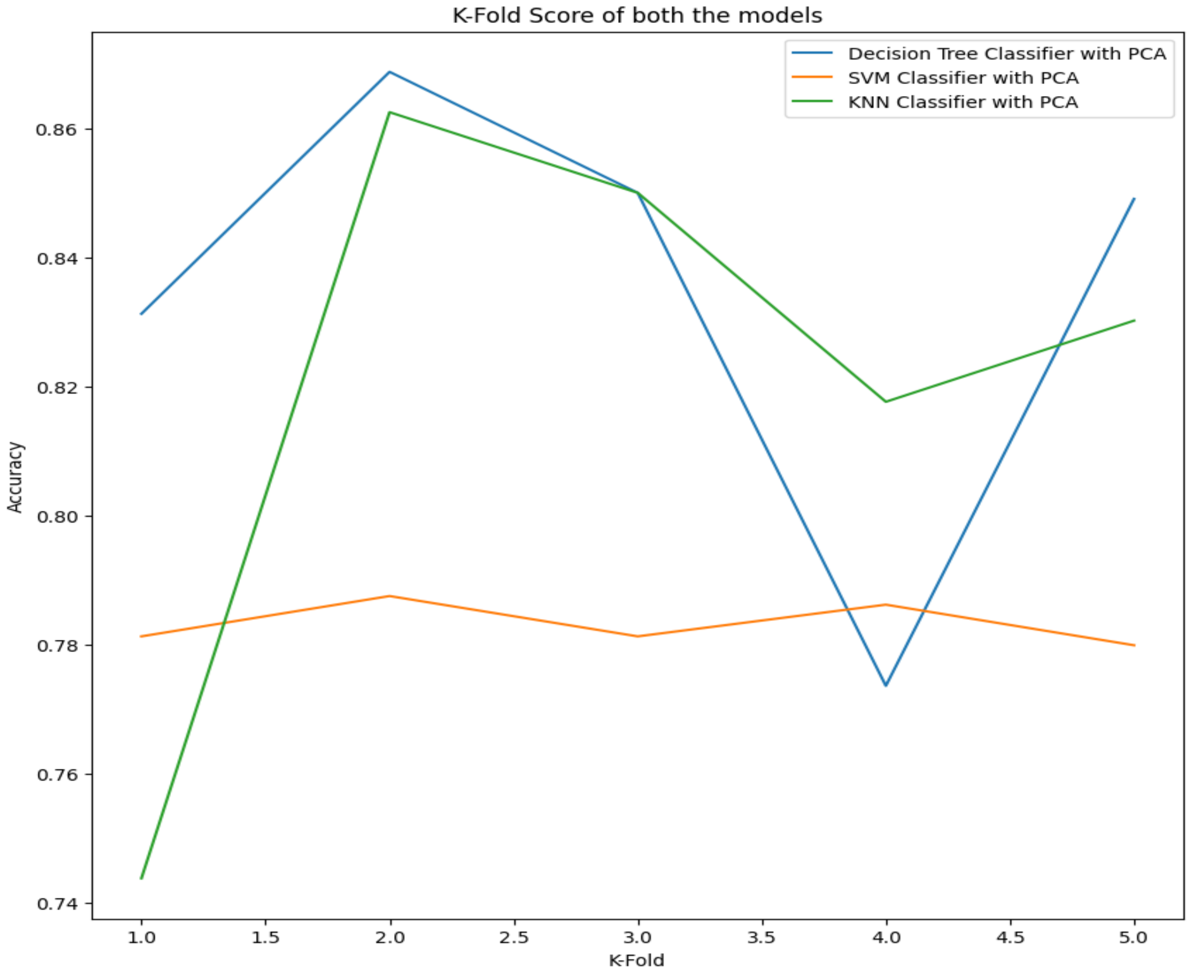
6. Train 2-3 chosen classification models alongside 5-Fold Cross-Validation Plots. **[5 Marks]**

In this part I have plotted the graph of 5-Fold Cross Validation after performing the PCA:

5-fold score of Decision Tree Classifier with PCA is: 0.8345283018867924

5-fold score of SVM Classifier with PCA is: 0.7832075471698113

5-fold score of KNN Classifier with PCA is: 0.820809748427673



6. Show the Test results of Classification Models on both types of datasets (Before and After PCA), via 2-3 Evaluation Metrics of choice (Ex:- Accuracy, Sensitivity, F1-Score, etc.) with the proper reasonings. **[5 + 5 Marks]**

Accuracy score of Decision Tree Classifier before PCA is: 0.9214285714285714

Accuracy score of Decision Tree Classifier after PCA is: 0.8

Accuracy score of SVM Classifier before PCA is: 0.8107142857142857

Accuracy score of SVM Classifier after PCA is: 0.7821428571428571

Accuracy score of KNN Classifier before PCA is: 0.8142857142857143

Accuracy score of KNN Classifier after PCA is: 0.8107142857142857

F1 score of Decision Tree Classifier before PCA is: 0.9184256089546594

F1 score of Decision Tree Classifier after PCA is: 0.803946164408941

F1 score of SVM Classifier before PCA is: 0.7604572254467871

F1 score of SVM Classifier after PCA is: 0.6930287840814157

F1 score of KNN Classifier before PCA is: 0.805140630611219

F1 score of KNN Classifier after PCA is: 0.792405434740756

Precision score of Decision Tree Classifier before PCA is: 0.9185251732271857

Precision score of Decision Tree Classifier after PCA is: 0.8094803944450928

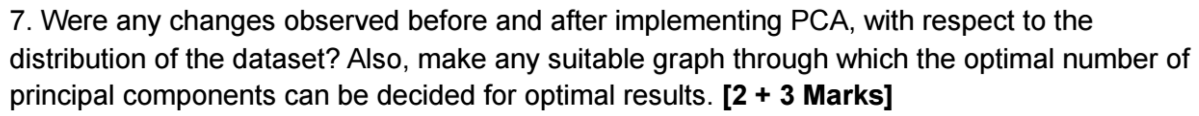
Precision score of SVM Classifier before PCA is: 0.7540368437451317

Precision score of SVM Classifier after PCA is: 0.7335174717368963

Precision score of KNN Classifier before PCA is: 0.8123562998160637

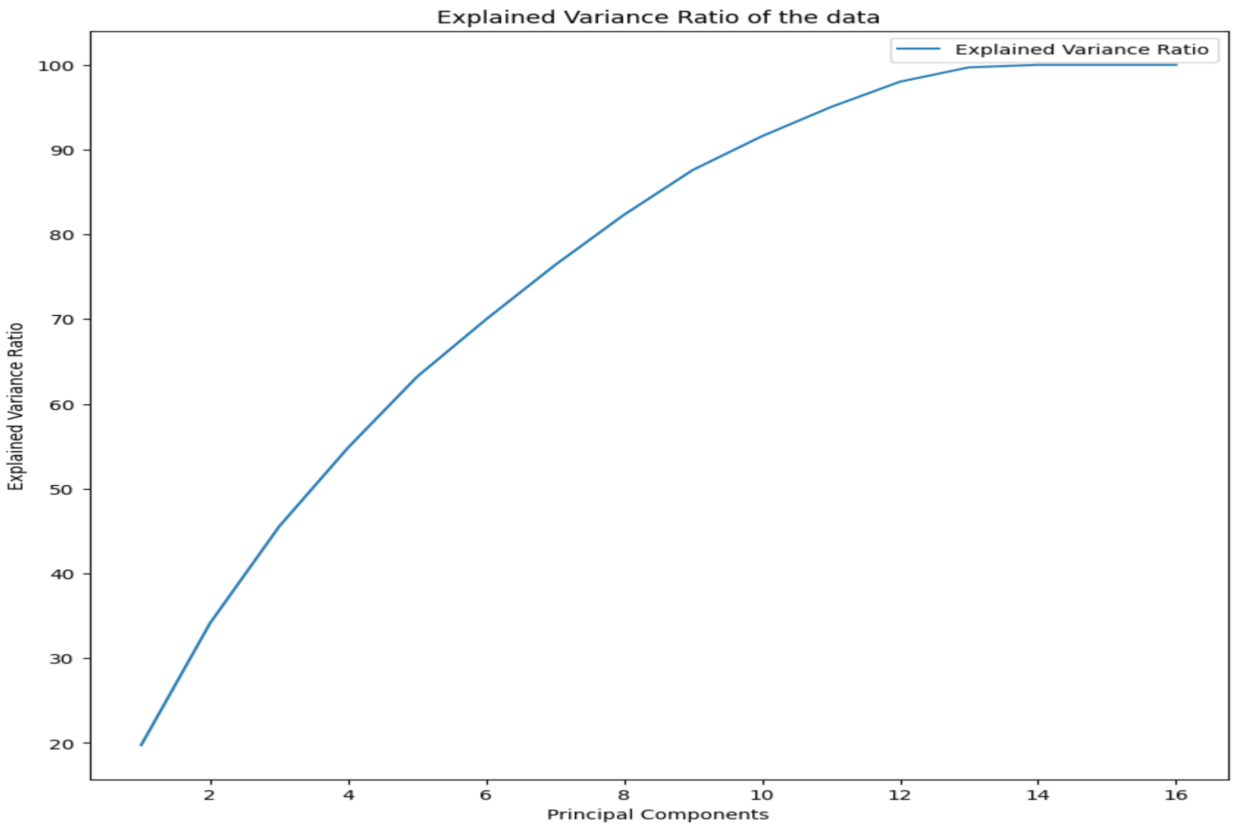
Precision score of KNN Classifier after PCA is: 0.7961747343565525

From These scores we can see that **SVM and KNN** don't change much after performing PCA because after performing PCA the information variance is retained . But **Decision Tree** scores change because PCA decreases the number of features So it will perform less than original data.



7. Were any changes observed before and after implementing PCA, with respect to the distribution of the dataset? Also, make any suitable graph through which the optimal number of principal components can be decided for optimal results. **[2 + 3 Marks]**

Dataset after PCA mostly gives the same information but with less features. Here is the graph for the variance explained and no of PC's. We can select the no of components as how much variance we want to be retained by the data.
I have used the method of cumulative sum of variance explained by the principal components and plotted the graph and found that 90% of the variance is explained by 10 principal components and initially there were 20 principal components and after applying PCA the number of principal components reduced to 10. So we can say that after applying PCA the number of features reduced to 10.



Bonus : Assuming the Naive Bayes assumption, calculate the eigenvector, eigenvalues and principal components. Do part 6 with these new feature vectors and comment on advantages/disadvantages you observed with this assumption.**[5 Marks]**

Here are the values of eigenvalues and eigenvectors:

```
[1.00125471 1.00125471 1.00125471 1.00125471 1.00125471 1.00125471
 1.00125471 1.00125471 1.00125471 1.00125471 1.00125471 1.00125471
 1.00125471 1.00125471 1.00125471 1.00125471]
[[1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]

 [0. 0. 0. 0. 0. 0. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0.]

 [0. 0. 0. 0. 0. 0. 0. 1. 0. 0. 0. 0. 0. 0. 0. 0.]

 [0. 0. 0. 0. 0. 0. 0. 0. 1. 0. 0. 0. 0. 0. 0. 0.]

 [0. 0. 0. 0. 0. 0. 0. 0. 0. 1. 0. 0. 0. 0. 0. 0.]

 [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 1. 0. 0. 0. 0. 0.]

 [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 1. 0. 0. 0. 0.]

 [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 1. 0. 0. 0.]

 [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 1. 0. 0.]

 [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 1. 0.]

 [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 1.]]
```

On assuming Naive Bayes the Covariance matrix becomes **diagonal**. Hence, **eigenvectors** come out to be of the same basis and come out to be **Identity matrix.** Hence , On taking projection the values of the features will not change and So, we can directly select the features having bigger eigenvalues as they will explain more Variance.

Advantages:

Time taken on to transform the dataset will reduce and can directly select the features.

Disadvantages:

It will be less effective when features are not independent but we assume it to be independent. So, the dataset will be less of use.

QUESTION 2:

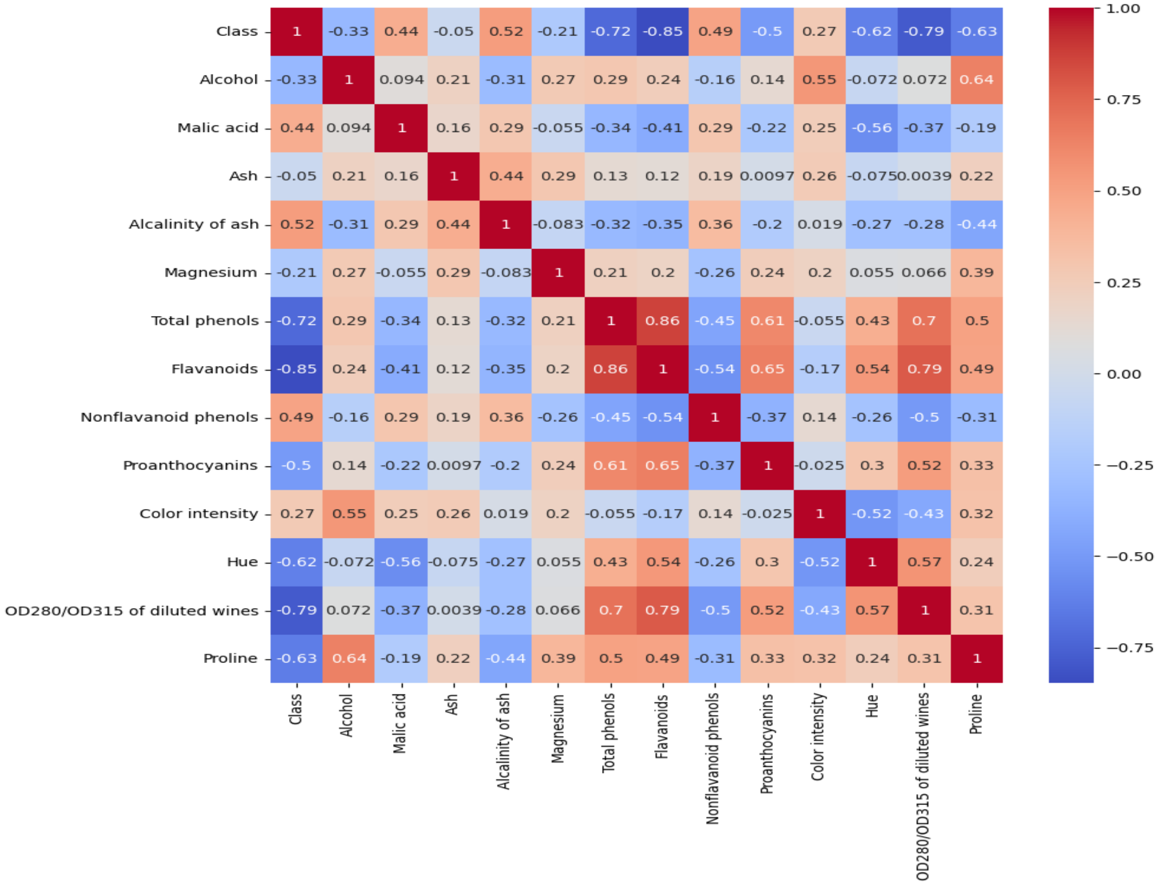
1. Implement Linear Discriminant Analysis from scratch with the following subtasks:-

   a. A function for computing within class and between class scatter matrices.

   b. A function that will automatically select the number of linear discriminants based upon the percentage of variance that needs to be conserved **[5+5 Marks]**

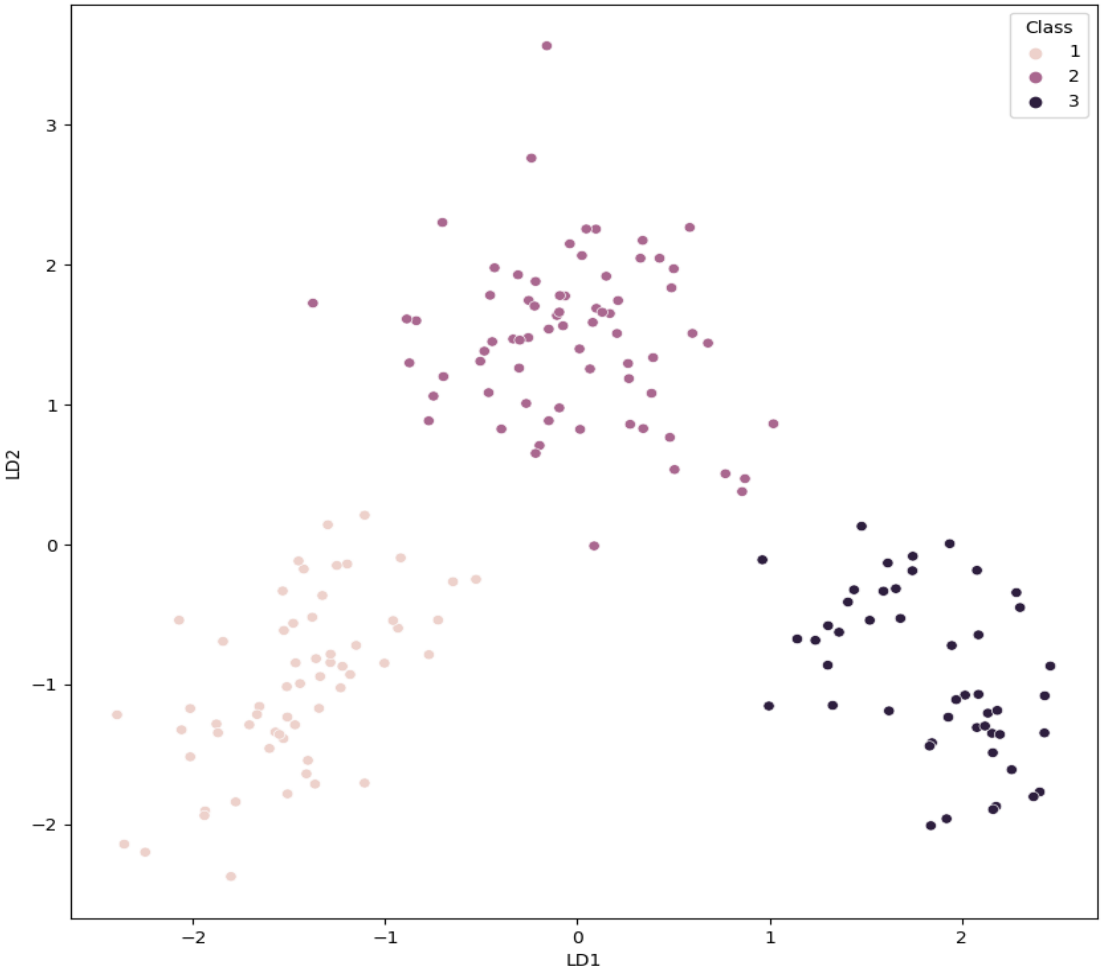
In this part I have performed the preprocessing of the data in which I have performed the Scaling and plotted pair plot and Correlation matrix. And also implemented the Class LDA in which given the variable **Varaince_explained** as the percentage of variance that should be explained.

Here is the plot for Correlation matrix:



2. Vary the variance and identify features that have a high impact on the classification tasks using LDA and visualize the feature space for the same using those linear discriminants.

After the implementation of the LDA the dataset is converted into the dataset that has Two **LD's** components. As, they have almost explained the 99% percentage of the variance.
Here is the plot of both the LD's:



3. Perform PCA on the dataset and compare the results with LDA by using any 2 classification techniques. **[3 Marks]**

In this part I have applied PCA and LDA on the same dataset and then compared the Accuracies of different Classifiers. Classifiers are Decision Tree, SVM and KNN.

Here are the accuracies for the same:

Decision Tree Accuracy with LDA: 1.0

Decision Tree Accuracy with PCA: 0.9166666666666666

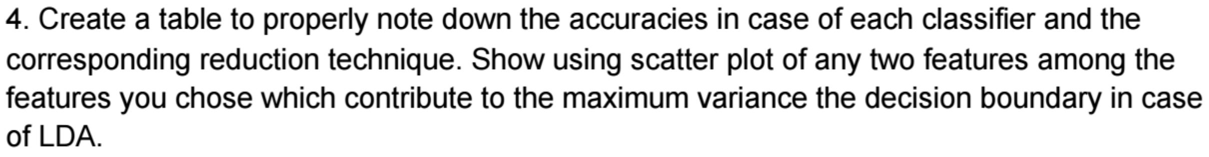
SVC Accuracy with LDA: 1.0

SVC Accuracy with PCA: 0.9444444444444444

KNN Accuracy with LDA: 1.0

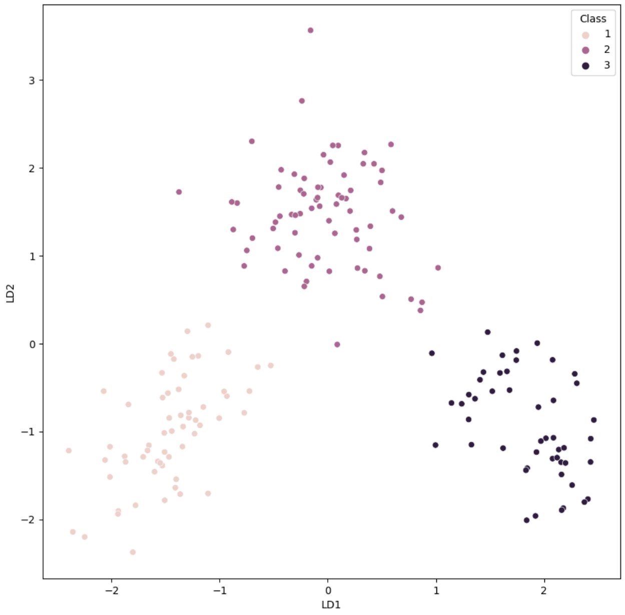
KNN Accuracy with PCA: 0.9444444444444444

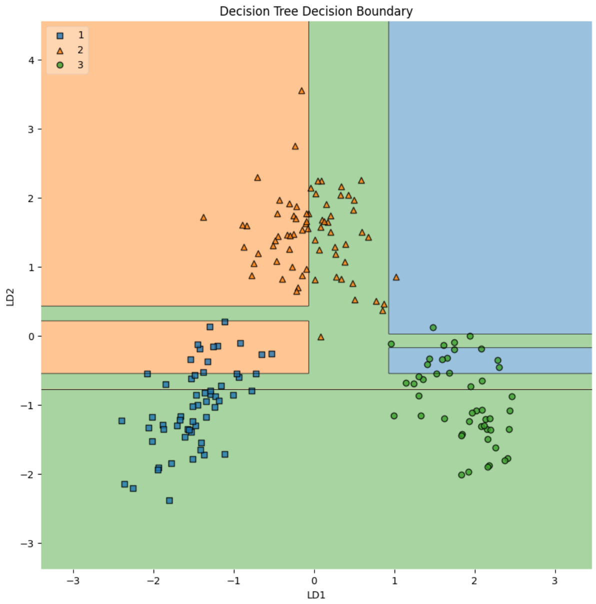
We can clearly see that **LDA is performing better than PCA. It is** also expected because LDA makes Transformation of the data according to Classes. And LDA is performing best because it almost explained 100% of the variance just by two LD's.



4. Create a table to properly note down the accuracies in case of each classifier and the corresponding reduction technique. Show using scatter plot of any two features among the features you chose which contribute to the maximum variance the decision boundary in case of LDA.

The accuracy of each classifier in each case is already explained in the above parts.
Here are the plots of scatter plots and decision boundary:

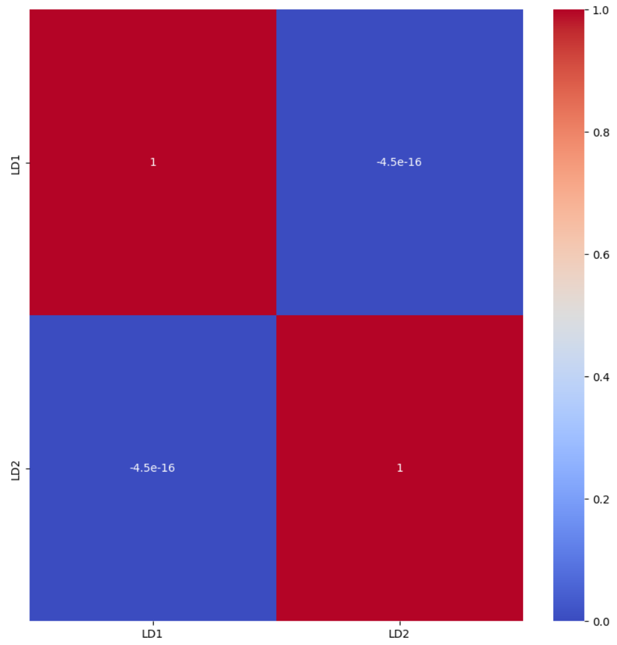


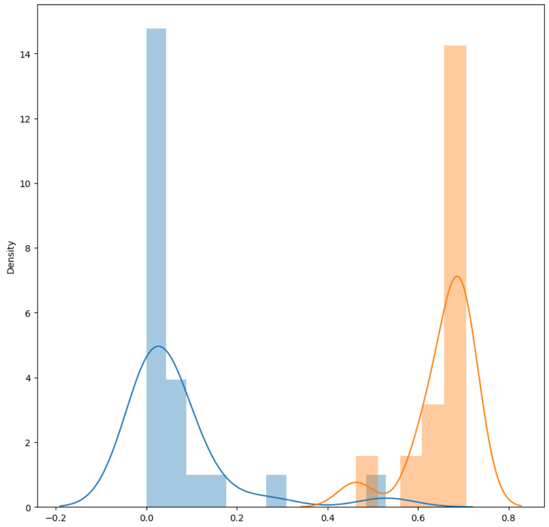


5. Using LDA as a classifier, perform 5-fold cross-validation and plot ROC and compute AUC for each fold from scratch **[10 Marks]**

As we can see from the Correlation Graph also features are independent and Gaussian distributed. So ,we can use Gaussian Naive Bayes Classifiers directly as LDA Classifiers.

Here are the plots for the same:



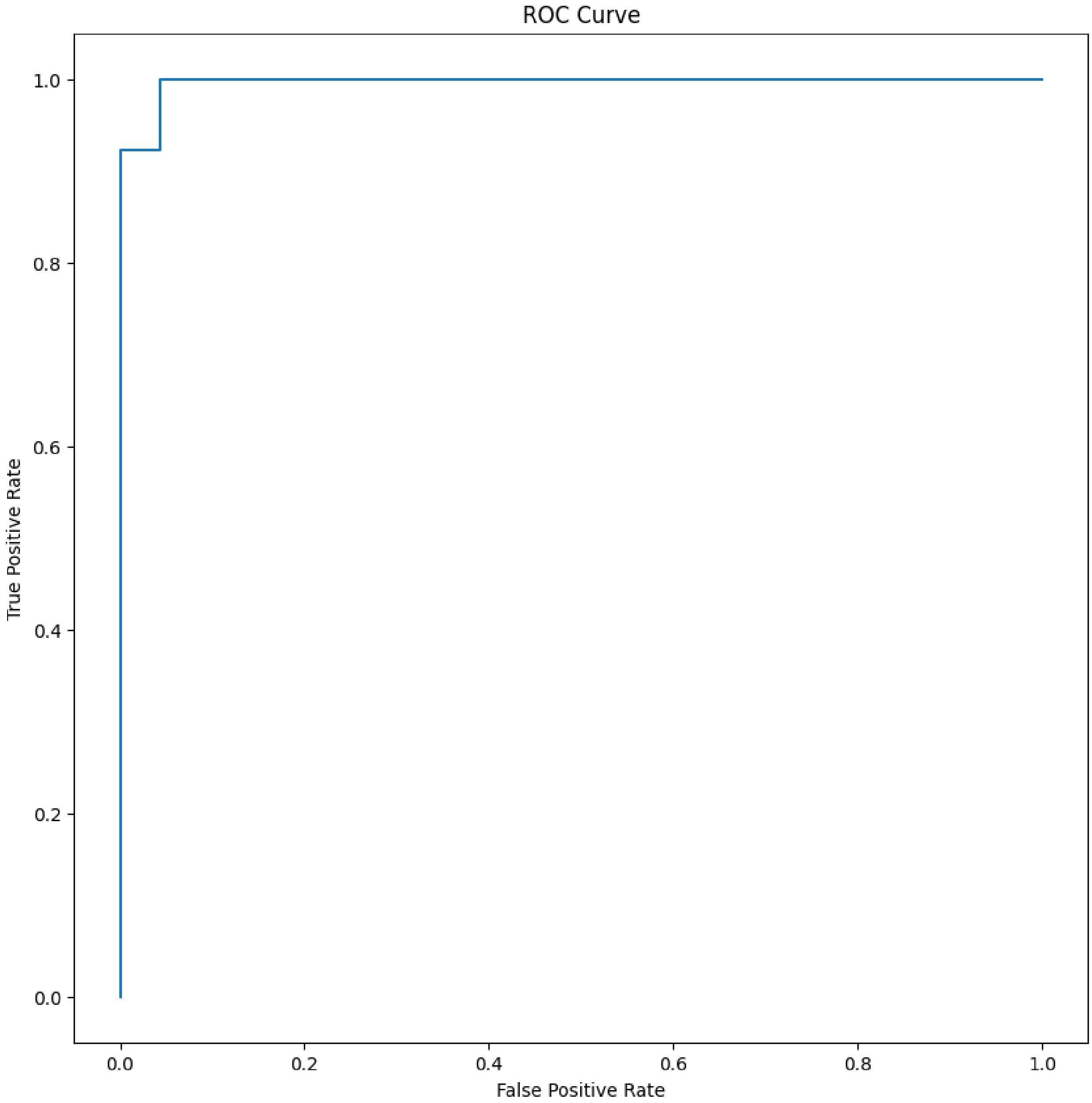


As, we know that we have two **LD's and 3 classes** as output. But roc_curve can be plotted only for **1 Feature and 2 Classes** to calculate the TPR and FPR. So, I have **LD2** and replaced the class 3 with 2 and then trained the data to Gaussian Naive Bayes.
Here are the results for the 5-fold Cross Validation , Roc_curve and Auc score:

5-fold cross validation scores: [1.0, 1.0, 1.0, 1.0, 1.0] this score on training the dataset with LD1 and LD2.

Here is the AUC score of the curve.:

AUC score of data is : 0.9966555183946488


ROC Curve
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
0.0
0.2
0.4
0.6
0.8
1.0