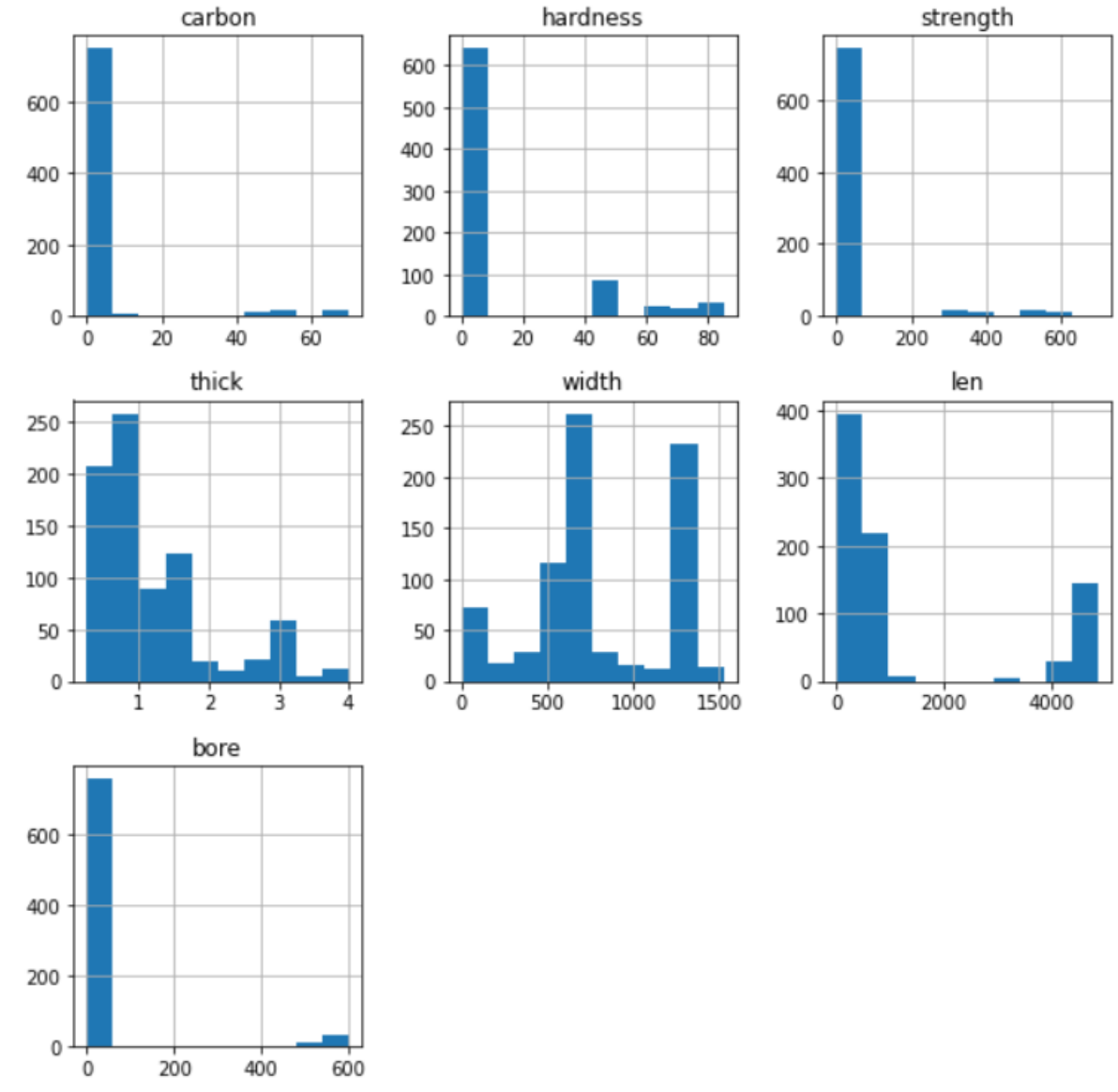
Lab 7 Report

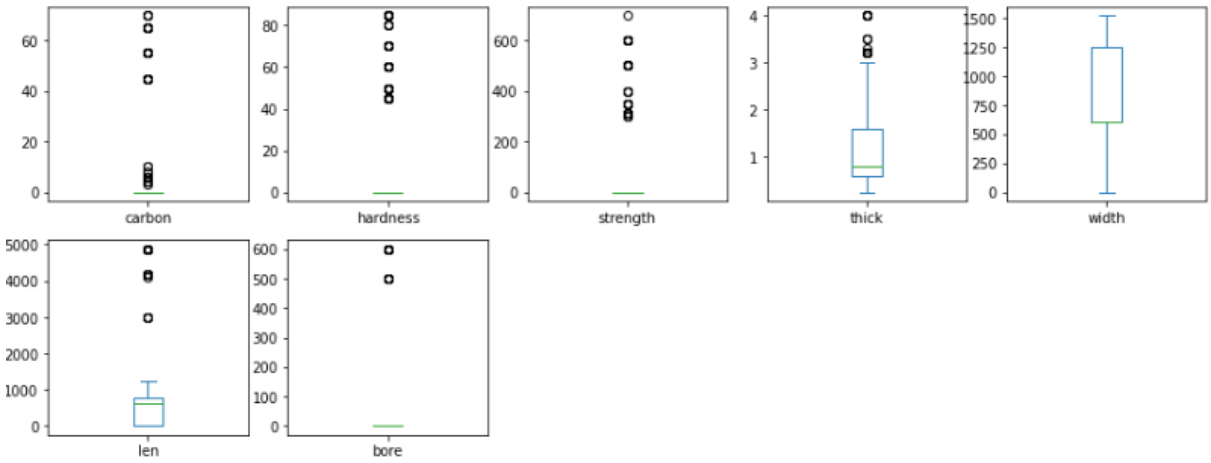
Nakkina Vinay (B21AI023)

Question 1

Subpart 1:

- Getting the dataset and using the head() to print it
- Getting the information of the dataset using info()
- Visualising the dataset using histogram and box plots





Subpart 2:

- Replacing the '?' values in the dataset with nan using np.nan
- Now dropping some unnecessary columns from the dataset

```
dataset1 = dataset1.drop(['family', 'temper_rolling', 'non-
ageing', 'surface-
finish', 'enamelability', 'bc', 'bf', 'bt', 'bw/me', 'bl', 'm','chrom
', 'phos', 'cbond','marvi','exptl', 'ferro','corr', 'blue/bright/varn
/clean', 'lustre','jurofm','s','p', 'oil', 'packing'], axis=1)
```

- Filling the missing values in columns using fillna
- Now performing LabelEncoder to some columns in the dataset
- Now splitting the data into X and y and splitting the data into train and test sets in the ratio of 65:35 using train_test_split
- Standardising the values using StandardScaler

Subpart 3:

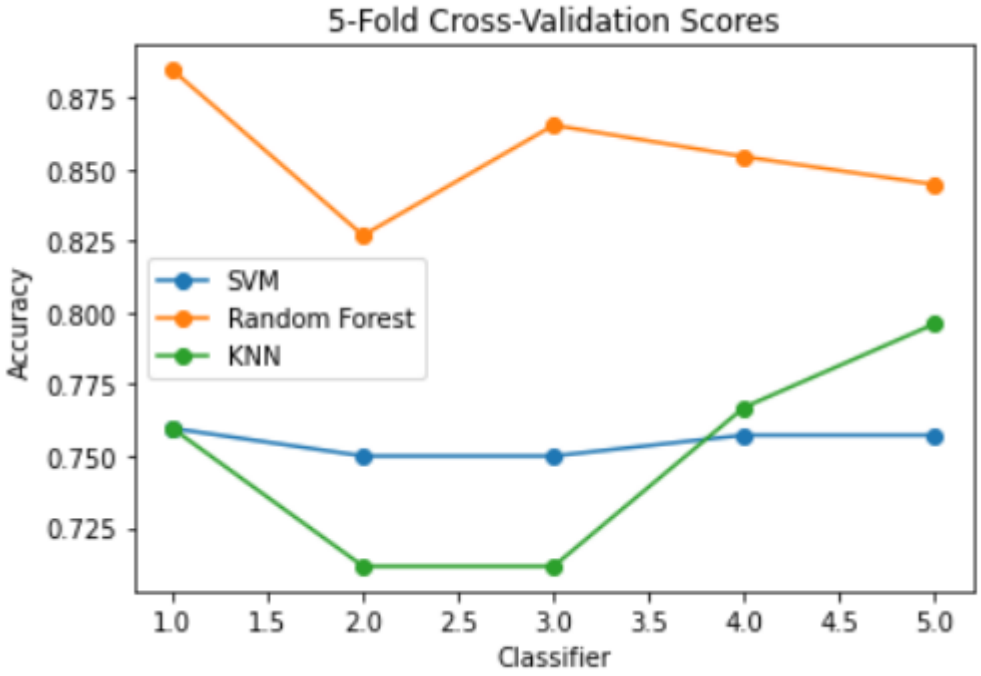
- Defining the three models SVM, Random Forest and KNeighboursClassifier
- Evaluating each model using 5-fold cross validation and storing the cross validation scores of both the standardisied data and non-standardisied data in two list variables
- Printing the mean of cross validation scores

```
SVM:
Mean accuracy: 0.7548
Std deviation: 0.0040
SVM:
Mean accuracy std: 0.8417
Std deviation: 0.0246
```

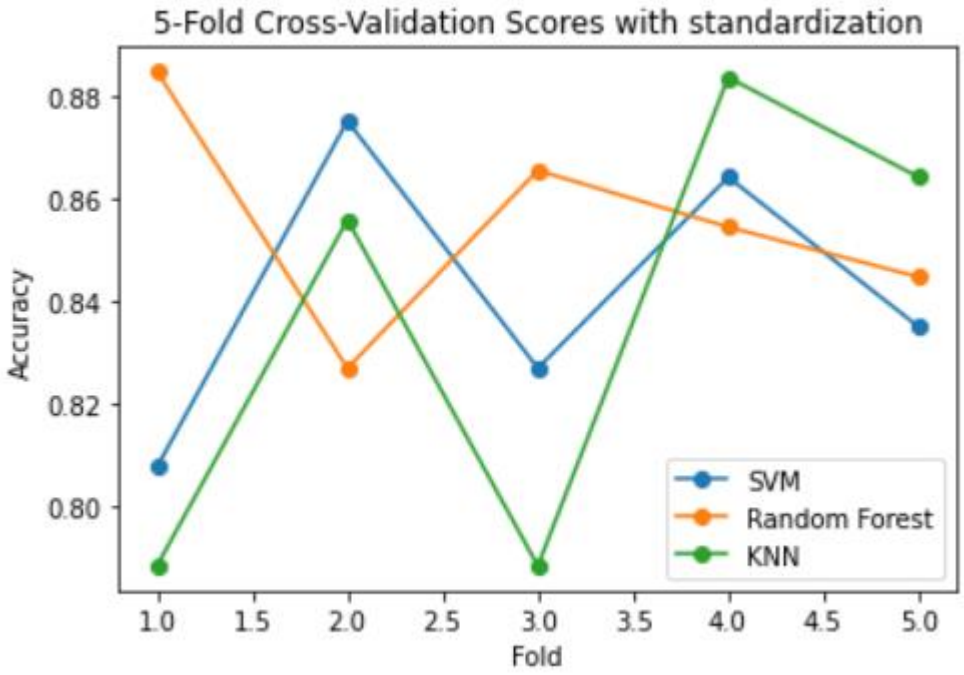
```
Random Forest:
Mean accuracy: 0.8552
Std deviation: 0.0194
Random Forest:
Mean accuracy std: 0.8552
Std deviation: 0.0194
```

```
KNN:
Mean accuracy: 0.7492
Std deviation: 0.0331
KNN:
Mean accuracy std: 0.8361
Std deviation: 0.0399
```

- Plotting the graph of variation of cross validation score when data is not standardised



- Plotting the graph of variation of cross validation score when data is standardised



Subpart 4:

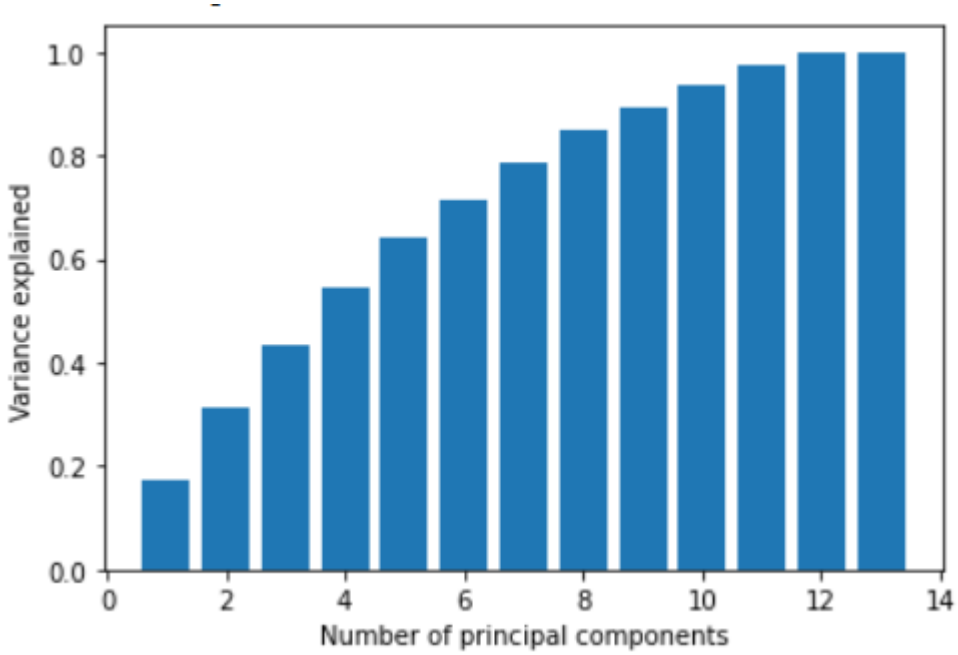
- Making a PCA class from scratch
- __init__ function is used for initialisations
- fit_transform method, we first center the data by subtracting the mean and dividing by the standard deviation. Then, we compute the covariance matrix using the centered data. We then compute the eigenvectors and eigenvalues of the

covariance matrix, sort the eigenvectors by descending eigenvalues, and select the first n_components eigenvectors to compute the principal components. Finally, we transform the data to the new coordinate system defined by the principal components

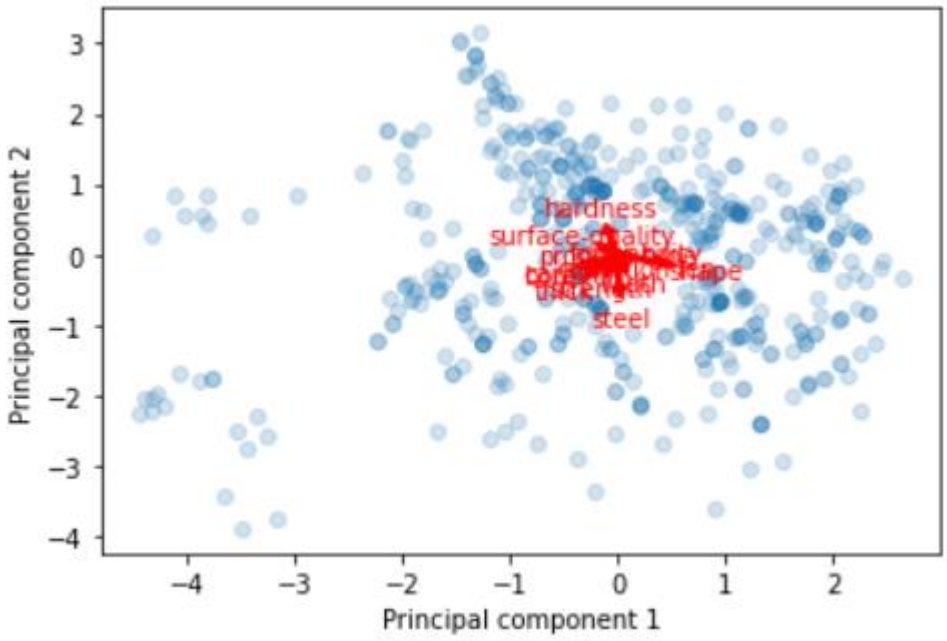
- transform method, we simply center the data and transform it to the new coordinate system defined by the principal components.
- The role of eigenvectors in PCA is to define the principal components, which are the directions of maximum variance in the data.
- The eigenvectors correspond to these directions, and the eigenvalues represent the variance of the data along each eigenvector.
- By selecting the eigenvectors with the highest eigenvalues, we can identify the most important directions in the data and use them to reduce its dimensionality.

Subpart 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.



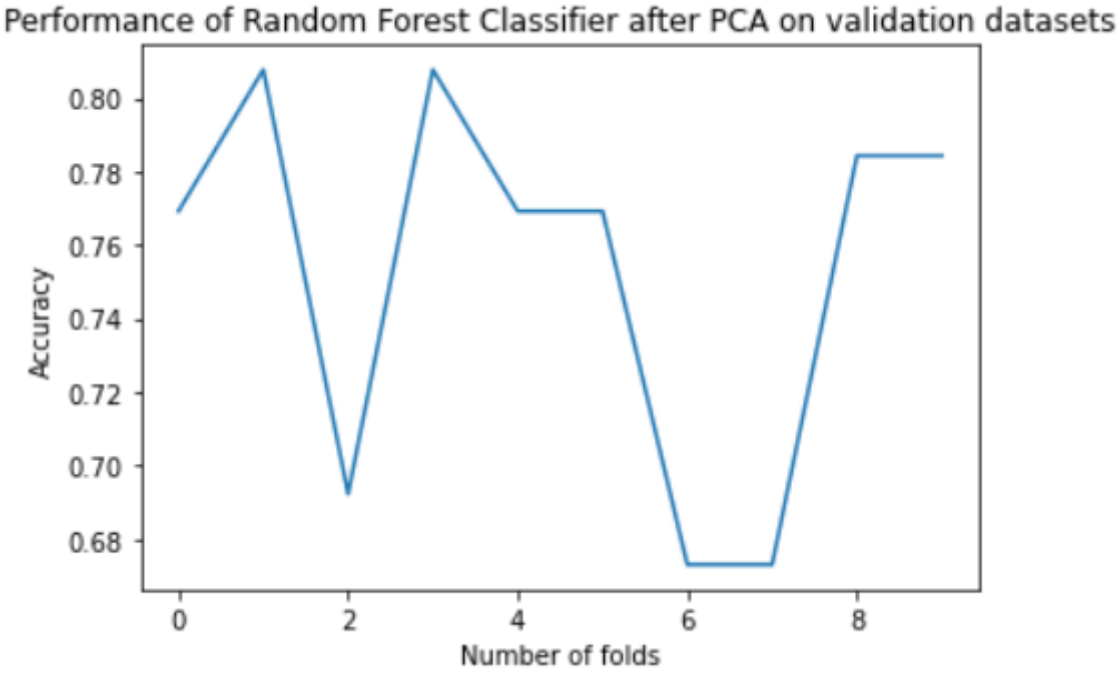
- Plotting a scatter plot to show the direction of the eigenvectors along with the data points

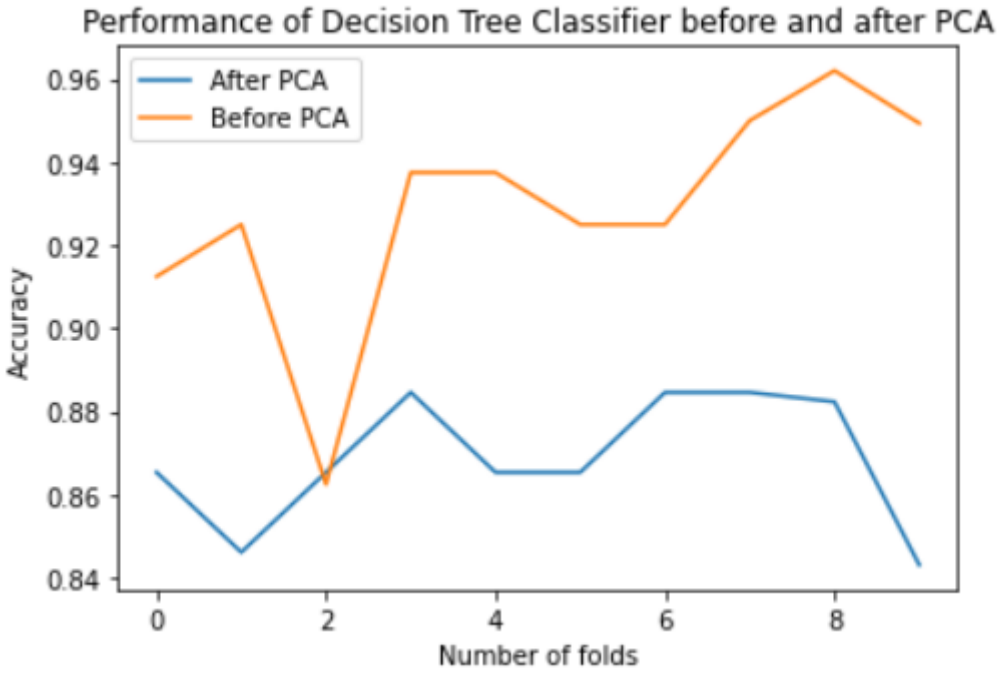


Subpart 6:

- Training Random Forest classification model and Decision Tree classification model and calculating the accuracies for the performance of the classifier before and after PCA and plotting 5-Fold Cross-Validation.

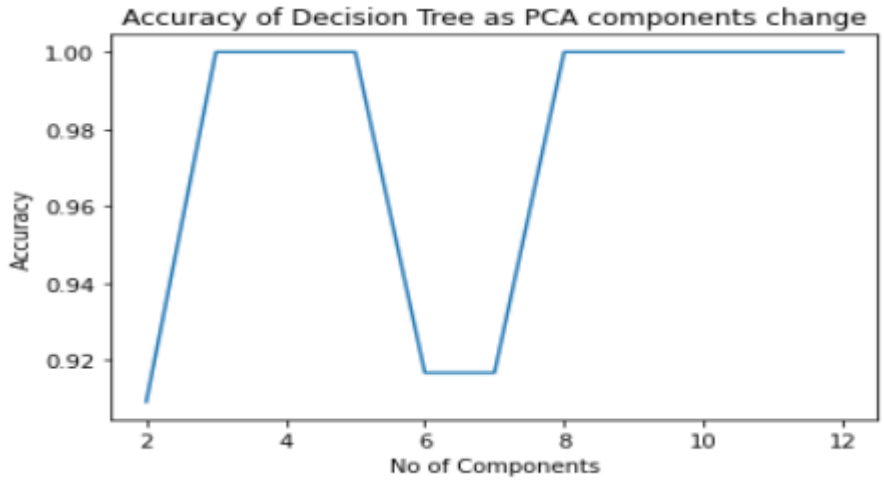
```
Average accuracy of Decision Tree before PCA:  0.9286392405063291
Average Accuracy of Decision Tree after PCA:  0.8687028657616894
```





Subpart 7:

- The find optimum no of components, find_optimum_components() method runs PCA with various numbers of principle components on the feature matrix X and target variable y.
- After training a Random Forest classifier on the smaller dataset, cross-validation calculates accuracy. It retains accuracy values after repeating this technique for various primary component numbers.
- Lastly, it graphs accuracy vs primary component number and provides the appropriate number of components to optimise accuracy.
- The graph may be used to determine the most accurate number of main components for categorization.
- Cross-validation prevents overfitting and improves accuracy estimates.
- This function helps pick the right amount of principle components for a classification problem, improving model performance and minimising feature space dimensionality.



Question 2

- Getting the dataset and performing the info()

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 178 entries, 0 to 177
Data columns (total 14 columns):
 #   Column                        Non-Null Count  Dtype
---  ------                        --------------  -----
 0   Classes                       178 non-null    int64
 1   Alcohol                       178 non-null    float64
 2   Malic acid                    178 non-null    float64
 3   Ash                           178 non-null    float64
 4   Alcalinity of ash             178 non-null    float64
 5   Magnesium                     178 non-null    int64
 6   Total phenols                 178 non-null    float64
 7   Flavanoids                    178 non-null    float64
 8   Nonflavanoid phenols          178 non-null    float64
 9   Proanthocyanins               178 non-null    float64
 10  Color intensity               178 non-null    float64
 11  Hue                           178 non-null    float64
 12  OD280/OD315 of diluted wines  178 non-null    float64
 13  Proline                       178 non-null    int64
dtypes: float64(11), int64(3)
memory usage: 19.6 KB
```

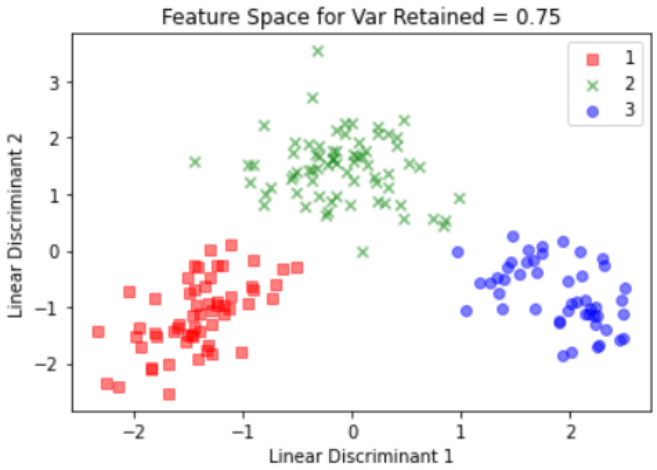
- Splitting the data into X and Y and standardising the data using StandardScaler
- After standardising splitting the data into train and test sets

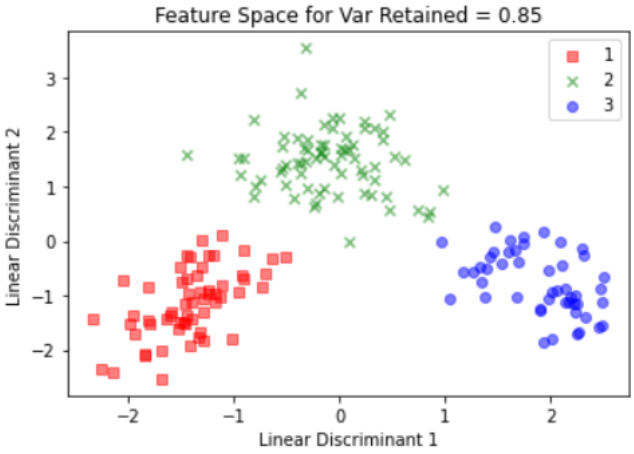
Subpart 1:

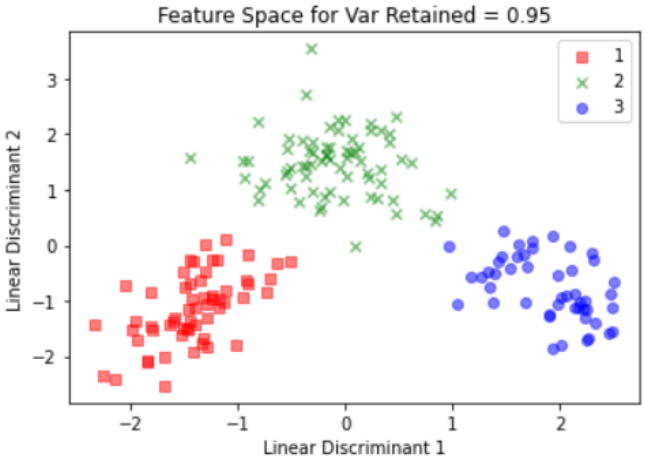
- Implementing LDA algorithm from scratch
- LDA class consists of
    - __init__ for initialisation of different variables being used in the class
    - Scatter_matrices
        - Function to compute within-class and between-class scatter matrices
    - Select_dimensions
        - Function to automatically select number of linear discriminants based upon the percentage of variance that needs to be conserved

Subpart 2:

- Varying the variance and identifying features that have a high impact on the classification tasks using LDA and visualizing the feature space for the same using those linear discriminants.







Subpart 3:

- Performing PCA on the dataset and comparing the results with LDA by using Logistic Regression and SVM and printing the accuracies

```
Logistic Regression with PCA: accuracy = 1.00
Logistic Regression with LDA: accuracy = 1.00
SVM with PCA: accuracy = 0.97
SVM with LDA: accuracy = 1.00
```