FOUNDATIONS OF DATA SCIENCE REPORT

GROUP 63

SUBMITTED BY

VAIBHAV SINGLA 2021MCB1254
SNEHA SHAH 2021MCB1368
T NIKHIL 2021MCB1250
ADITI 2021MCB1227

OVERVIEW

This report includes the following:

1. Problem statement

- Use the given image to compress it to 30, 60, 90% of the original size.
- Use the data 1 and apply PCA and SVD to reduce the dimension to 60, 75 and 90% of the original. Give a detailed EDA on PCA data with 60% of the original dimension.
- Based on Decision Trees, Random Forests, and KNN to classify the Diabetes data. Compare different methods.
- Apply logistic regression and QDA on OJ data. Compare the different methods. Do EDA on all the datasets

2. A brief summary of the code.

3. A brief description of the data set.

4. Findings obtained on running the code and its explanation.

Problem statement 1: Use the given image to compress it to 30, 60, 90% of the original size.

Image compression

Image compression is a fundamental technique in the field of digital media, addressing the need for efficient storage and transmission of images. We compress images by resizing them based on user-defined scale factors. The script utilizes the Pillow library (an updated fork of the Python Imaging Library) to open, resize, and save images in a compressed format.

This is image 1:

Image compression to 30%

Image compression to 60%

Image Compression to 90%

The code defines a function, resize_image, which takes an input image file path, an output path for the resized image, and a scale factor as parameters. The script interacts with the user by prompting them to input the name of the original image file. Subsequently, the function is called three times with varying scale factors (30%, 60%, and 90%) to resize the image accordingly.

The resized images are then saved with distinct filenames. The resizing process inherently compresses the images, as it calculates a new size and resizes the original image accordingly, resulting in smaller file sizes for the compressed images. Any exceptions during this process are caught and printed, providing error handling for the user.

Problem statement 2:**Use the data 1 and apply PCA and SVD to reduce the dimension to 60, 75 and 90% of the original. Give a detailed EDA on PCA data with 60% of the original dimension.**

Principal Component Analysis(PCA) :

Principal component analysis is a statistical method used for dimensionality reduction in data analysis.There various steps involved to do this:

- Let X be a data matrix which we need to reduce.
- Compute the covariance matrix of it,which is $(1/(n-1)) * (Xcent.T * Xcent)$,where Xcent is centered data.
- Find the eigenvalues and eigenvectors of the covariance matrix.Sort the eigenvalues in descending order and corresponding eigenvectors in the same order.
- Variance captured by an eigenvector is the ratio of the corresponding eigenvalue to the sum of all eigenvalues.

After doing the above steps,if the number of principal components needed are k, then the reduced or projected data is $X * E$, in which :

- ➢ X is data matrix
- ➢ E contains k columns which are k eigenvectors corresponding to the first k largest eigenvalues.

The size of reduces matrix is $n * k$

Percentage of variances captured by ith principal component for libras movement is given below:(for first 10):

```
Percentage of variance captured by 0th principal component: 25.096891494722556
Percentage of variance captured by 1th principal component: 22.059426021834433
Percentage of variance captured by 2th principal component: 18.093573650159126
Percentage of variance captured by 3th principal component: 10.480391752302818
Percentage of variance captured by 4th principal component: 6.550805059494029
Percentage of variance captured by 5th principal component: 4.9488327220939015
Percentage of variance captured by 6th principal component: 3.472032677279417
Percentage of variance captured by 7th principal component: 2.9123186163615986
Percentage of variance captured by 8th principal component: 2.053419737979947
Percentage of variance captured by 9th principal component: 1.6262213691302985
```

- We require a minimum of 3 eigenvectors to capture 60%(25.09 + 22.05 +18.09) variance.
- We require a minimum of 4 eigenvectors to capture 60%(25.09 + 22.05 +18.09) variance.

- We require a minimum of 7 eigenvectors to capture 60%(25.09 + 22.05 +18.09 + 10.48+6.55+4.94+3.47) variance.

Singular Value Decomposition(SVD) :

This is also a technique used for data reduction.But this follows a different approach by computing singular value decomposition of data matrix.PCA and SVD are closely related.Below are the following steps I used to perform data reduction through SVD:

- Let the data matrix be X of size $n * p$.
- Calculate the SVD decomposition of X using scipy.linalg.svd() . It returns tuple containing:
    1. Matrix U.
    2. Singular values,S.
    3. Matrix V.

  These 3 satisfy:

  $$X = U * diag(S) * V$$
- The variance explained is the sum of squares of singular values.

After doing the above steps, if the required number of vectors are k,then the data reduces is $U[:,:k] * matS[:k,:k] * V[:k,:]$ .Here matS is diagonal matrix with diagonal values as values in S.

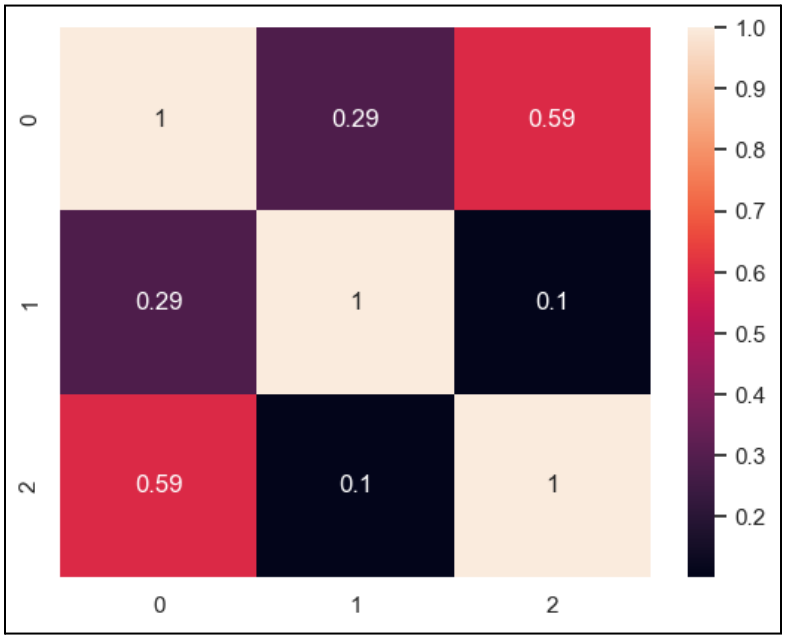
Percentage of variances explained by first 10 singular values are below:

```
Percentage of variance explained by 0th component: 90.86140374212084
Percentage of variance explained by 1th component: 2.6161877399527773
Percentage of variance explained by 2th component: 1.9394992927121342
Percentage of variance explained by 3th component: 1.9154176297048742
Percentage of variance explained by 4th component: 0.7387300124087858
Percentage of variance explained by 5th component: 0.5408433009848994
Percentage of variance explained by 6th component: 0.38519828264281375
Percentage of variance explained by 7th component: 0.31031164220589447
Percentage of variance explained by 8th component: 0.22646955019623
Percentage of variance explained by 9th component: 0.17505763347030406
```

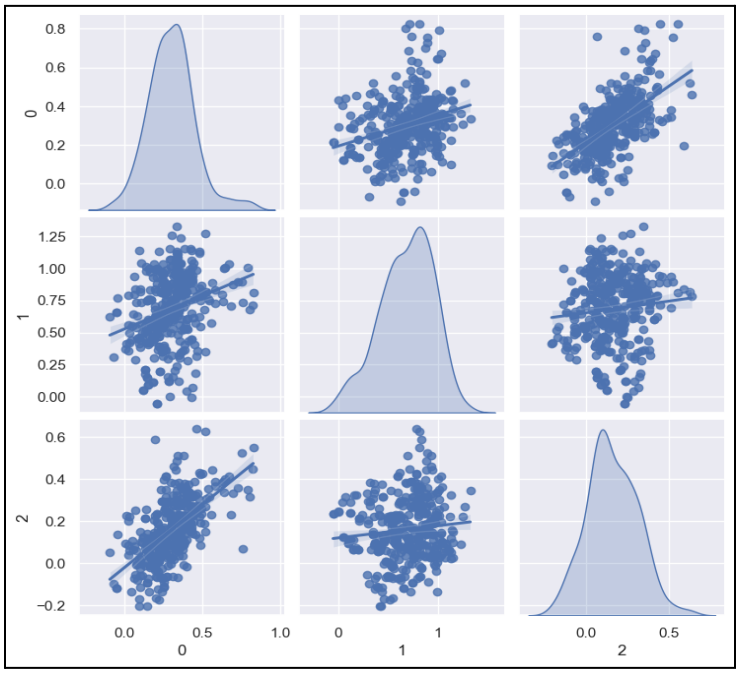
As the variance explained by the first singular value is approximately 91%,we will use just one singular value to capture 60%,75% and 90%.

Exploratory Data Analysis:

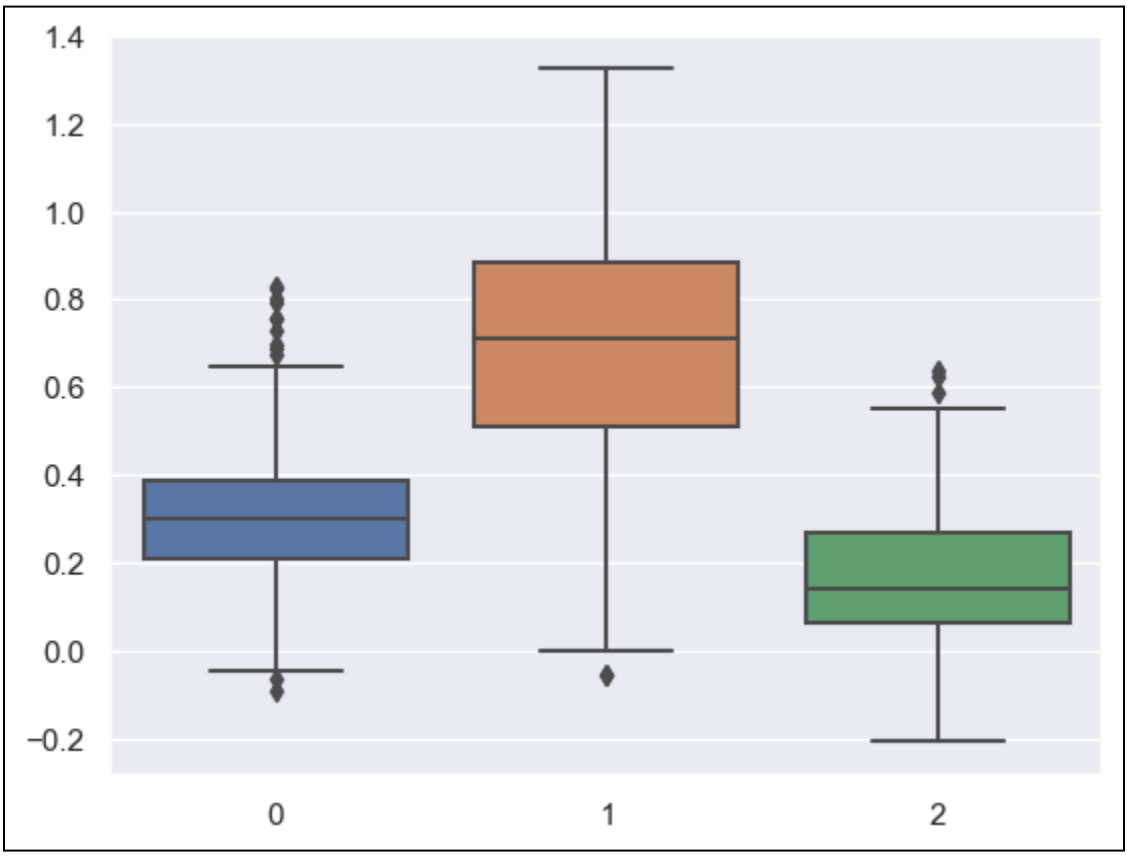
We get the data which is of size $n * p$ which is different from that of PCA.
To get the correlations of data columns after projection and visualize them, we used the heatmap of correlation matrix:



To understand how exactly each column of projected data is correlated to each other and the approximate distribution of each columns,we used pairplot:



To know the outliers, we used the box plot.The number of outliers have decreased significantly from 1st component to second component.This might be due to fact that component with high eigenvalue captures more variance gradually spreading the data more.



Reference:

- [jonathan_medium]

Problem Statement 3: Based on Decision Trees, Random Forests and KNN to classify the Diabetes data. Compare different methods.

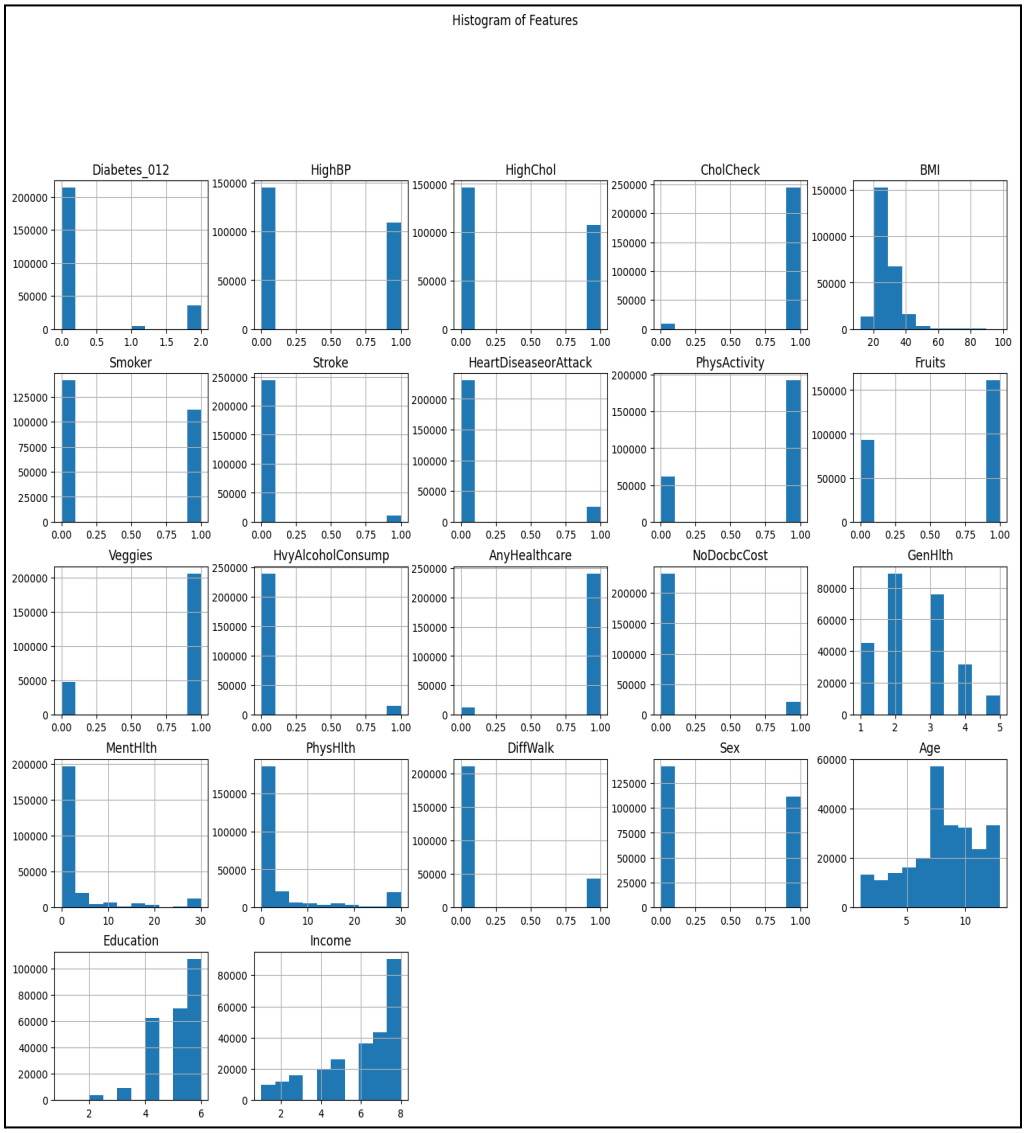
Exploratory Data Analysis (EDA)

Summary of the Data:

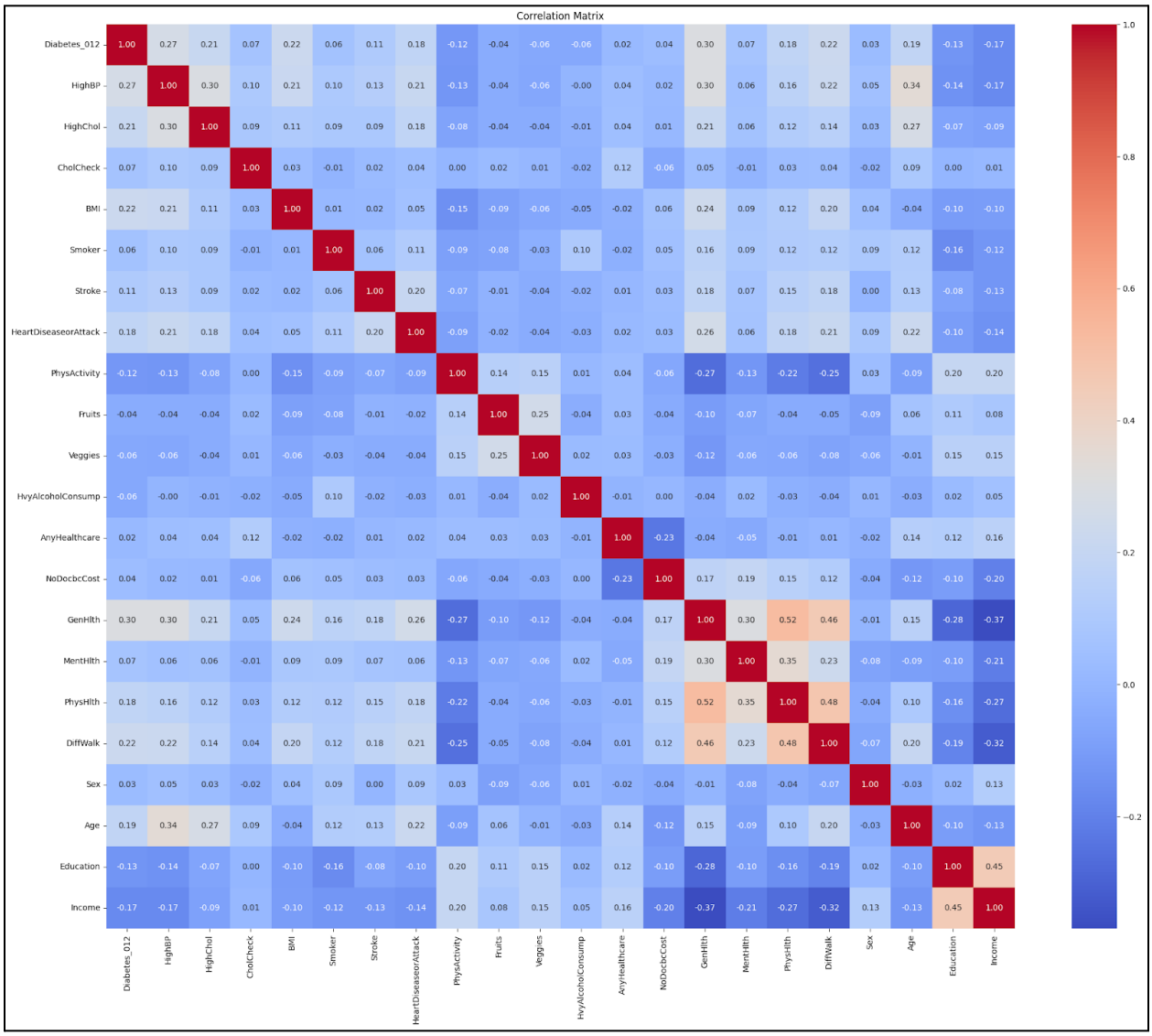
There are no null values in the data, this mean we need not fill any null values in data.
All the data values are of the data type “float64”.

Other information about the data can be found in the code outputs.
Informations like: Mean, Standard Deviation, Min Value, 25%, 50%, 75%, Max Value

The histogram of Features for the data is attached below:



The Correlation Matrix for the data is:



We split the data into the ratio 80/20, 20% of the data values are being used for the testing data while the other 80% is used to train the model.

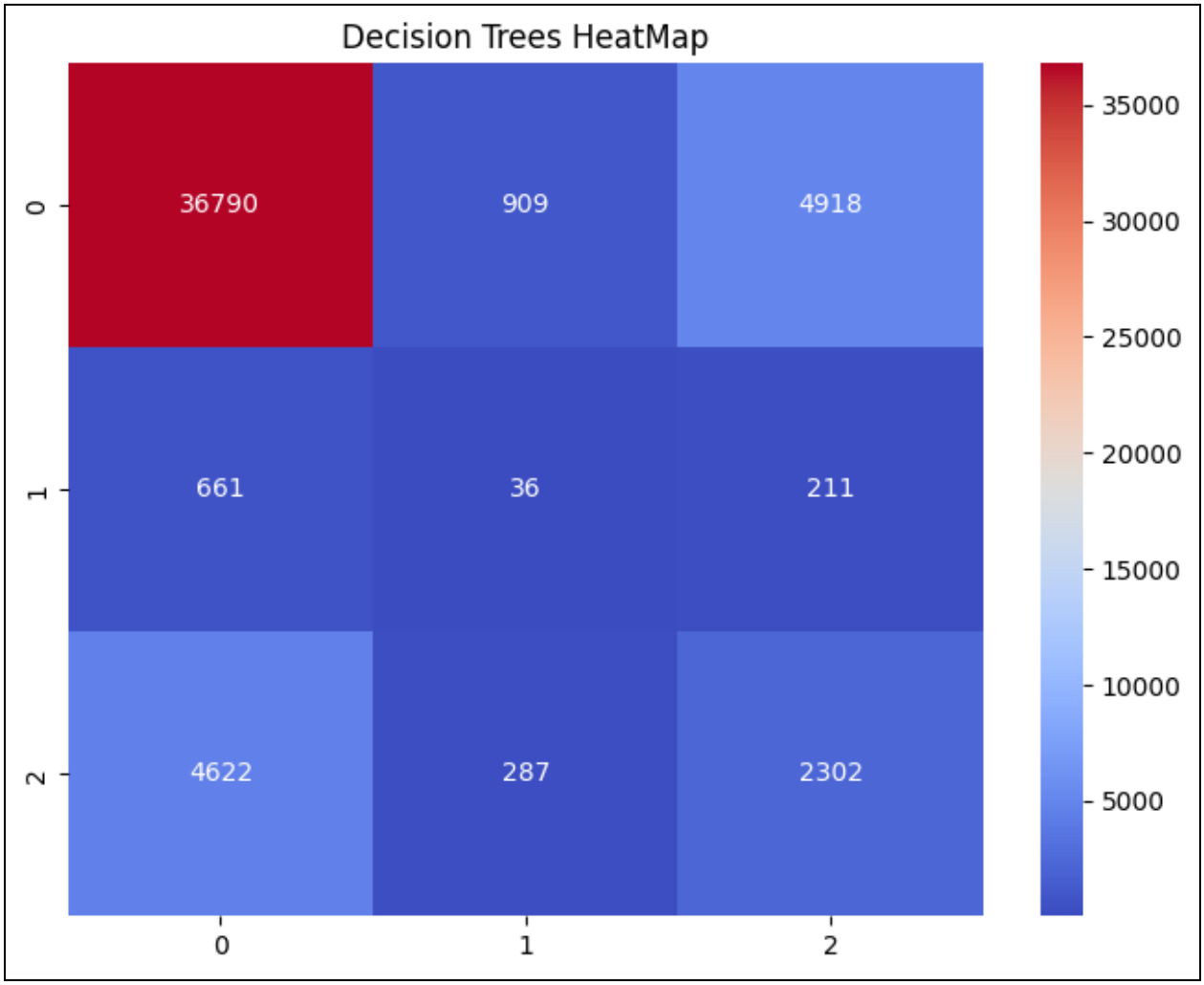
We use the data to train three types of Machine Learning Models namely:

Decision Tree:

Decision Tree Classification is a machine learning algorithm used for predictive modeling. It recursively partitions a dataset into subsets based on feature values, creating a tree-like structure. At each node, the algorithm selects the feature that best separates the data, resulting in decision nodes. This process continues until a

stopping criterion is met or a predefined depth is reached. The final nodes, or leaves, represent predicted class labels. Decision trees are intuitive, easy to interpret, and effective for classification tasks, capturing complex decision boundaries. However, they may be prone to overfitting, addressed by techniques like pruning or ensemble methods like Random Forests.

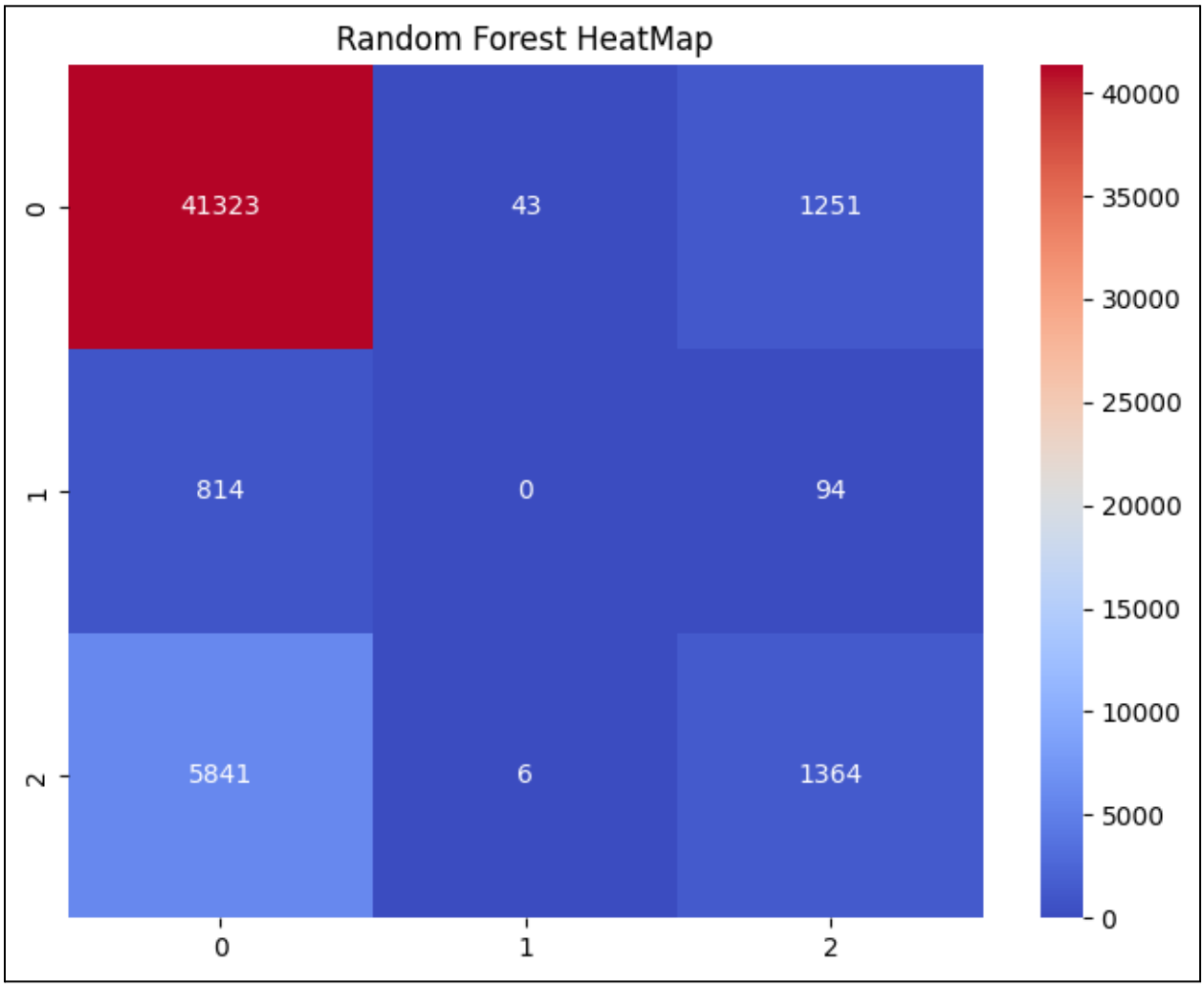
The Heatmap of predictions done by the Decision Tree Classifier is attached below.



Random Forest:

Random Forest is an ensemble learning method in machine learning that builds multiple decision trees and combines their predictions. It creates a diverse set of trees by training each on a random subset of the data and features. The algorithm then aggregates their predictions through voting (classification) or averaging (regression) to enhance accuracy and reduce overfitting. Random Forest is robust, handles high-dimensional data well, and is less prone to overfitting than individual decision trees. Its versatility and ability to handle diverse datasets make it a popular choice for various classification and regression tasks in practice.

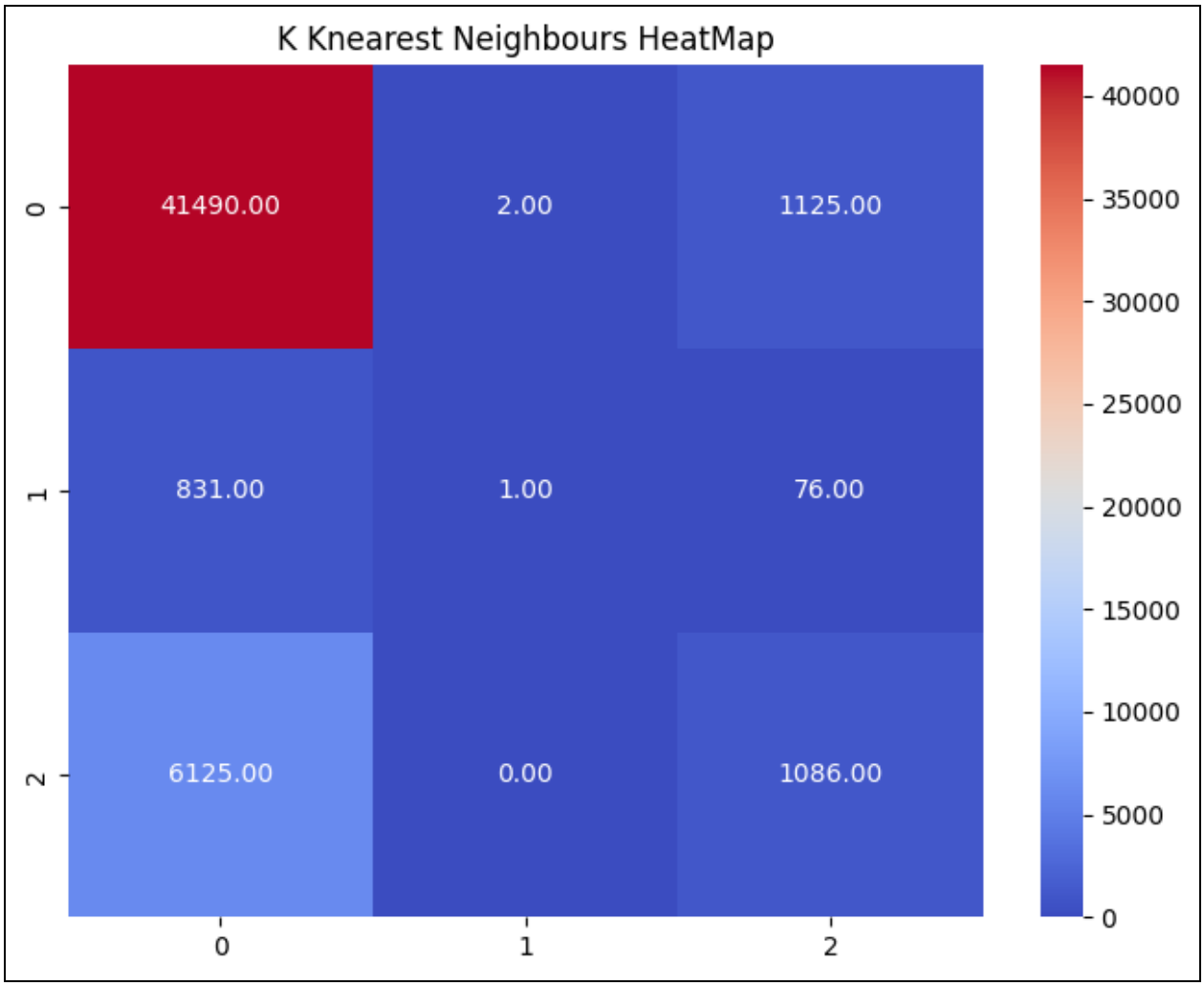
The Heatmap of predictions done by the Random Forest Classifier is attached below.



K Nearest Neighbours:

k-Nearest Neighbors (k-NN) is a simple yet powerful machine learning algorithm for classification and regression tasks. It classifies a data point based on the majority class of its k nearest neighbors in the feature space. The algorithm measures distance, often using Euclidean distance, to identify the nearest neighbors. The value of k, a hyperparameter, influences the model's sensitivity to noise and smoothness of decision boundaries. While k-NN is intuitive and easy to implement, it may be sensitive to outliers and computationally expensive for large datasets. It's particularly effective in applications with well-defined local structures and is widely employed in pattern recognition and recommendation systems.

The Heatmap of predictions done by the K Nearest Neighbors Classifier is attached below.


K Knearest Neighbours HeatMap
41490.00
2.00
1125.00
831.00
1.00
76.00
6125.00
0.00
1086.00
0
1
2
40000
35000
30000
25000
20000
15000
10000
5000
0

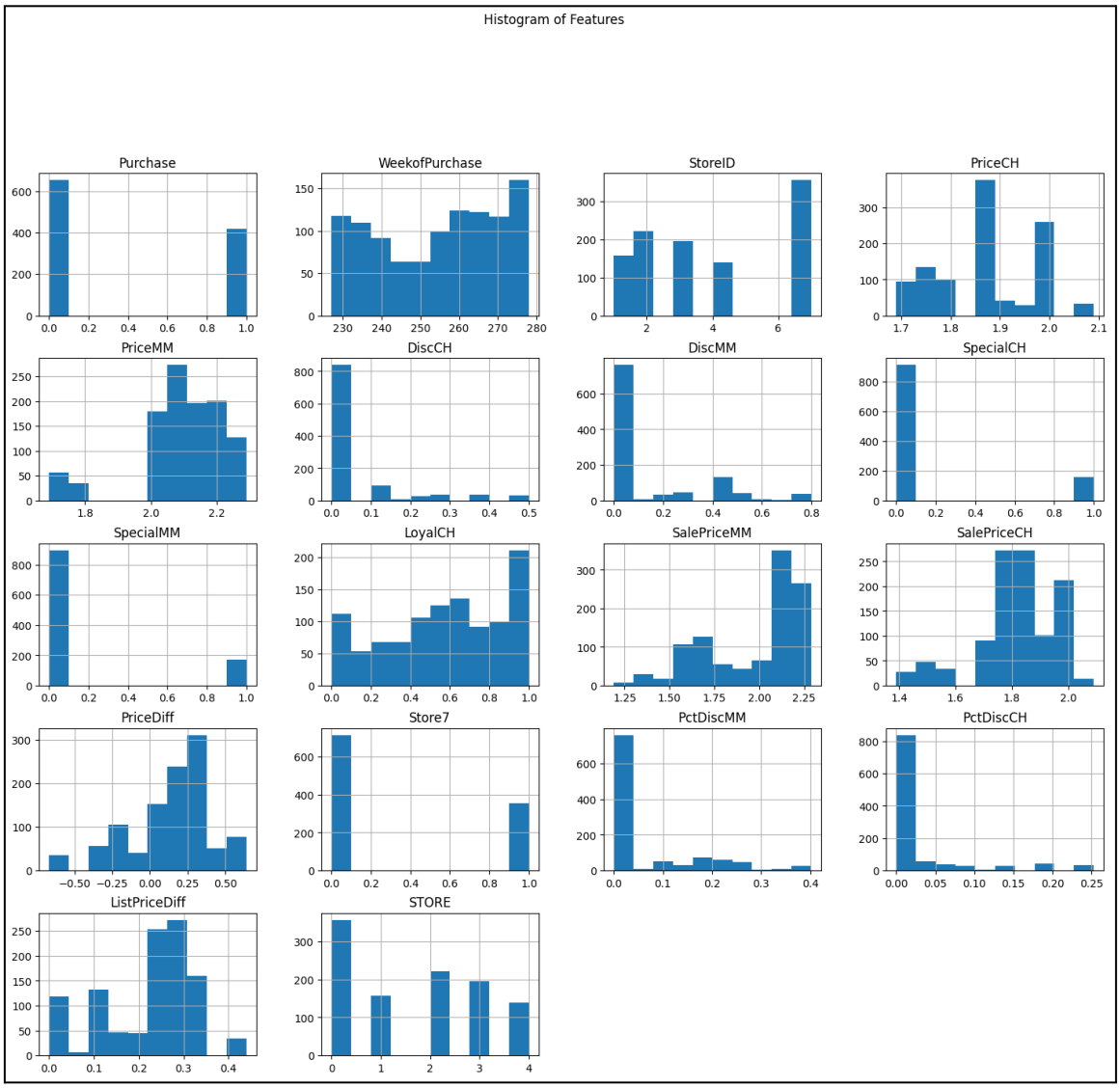
Problem Statement 4: **Apply logistic regression and QDA on OJ data. Compare the different methods. Do EDA on all the datasets**

Exploratory Data Analysis (EDA)

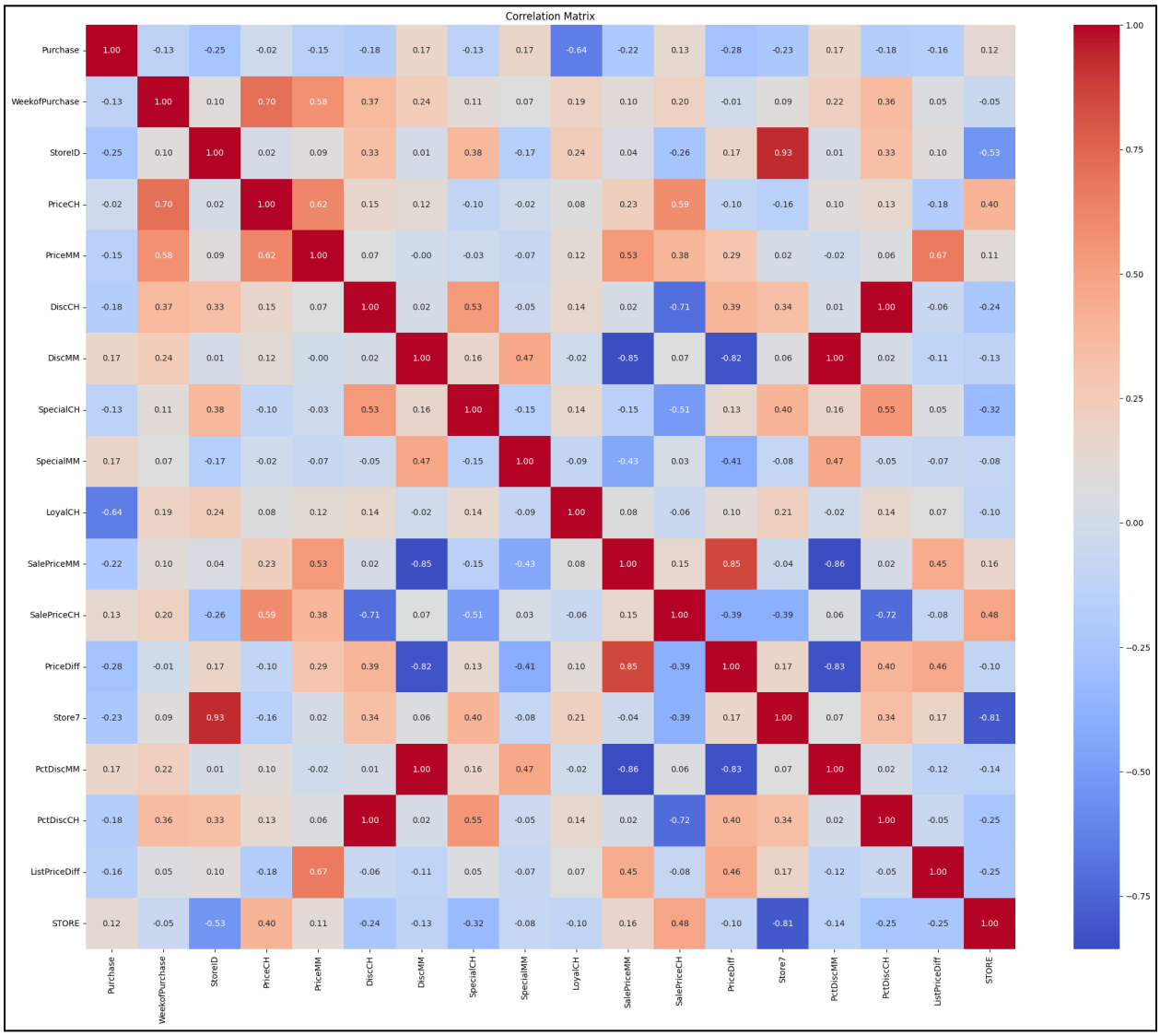
Summary of the Data:
There are no null values in the data, which means we need not fill in any null values in the data.
However, the data values in the dataset were of various types.
So we had to change them to a single data type, i.e. ‘float64’

Other information about the data can be found in the code outputs.
Informations like:
Mean
Standard Deviation
Min Value
25%
50%
75%
Max Value

The histogram of features for the data is attached below:



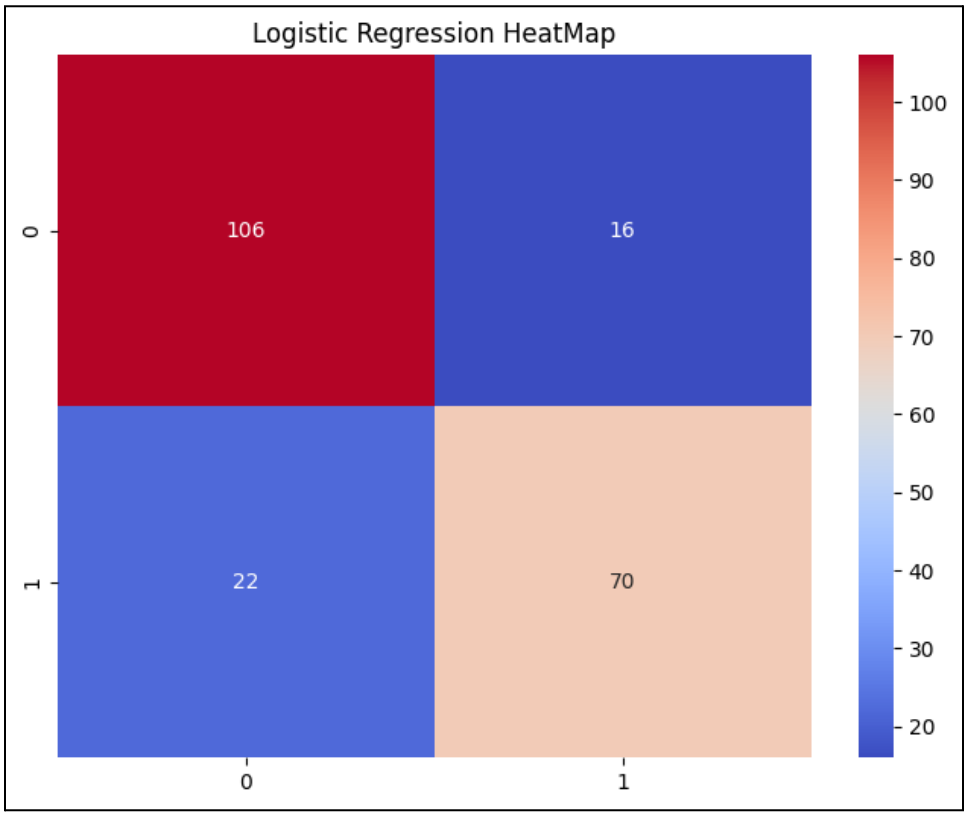
The Correlation Matrix for the data is:



- As we can see there is a high correlation between some of the columns, so we use PCA to reduce the dimension of our data.
- We drop some columns from the data to reduce the dimension and it also results in better performance by the prediction models.
- The explained variance ratio for the dataset is as follows,
  So seeing the data we set the ratio cap at 1e-8 for the data and cut off the columns that do not satisfy the above ratio.

We split the data into the ratio 80/20, 20% of the data values are used for the testing data while the other 80% is used to train the model.

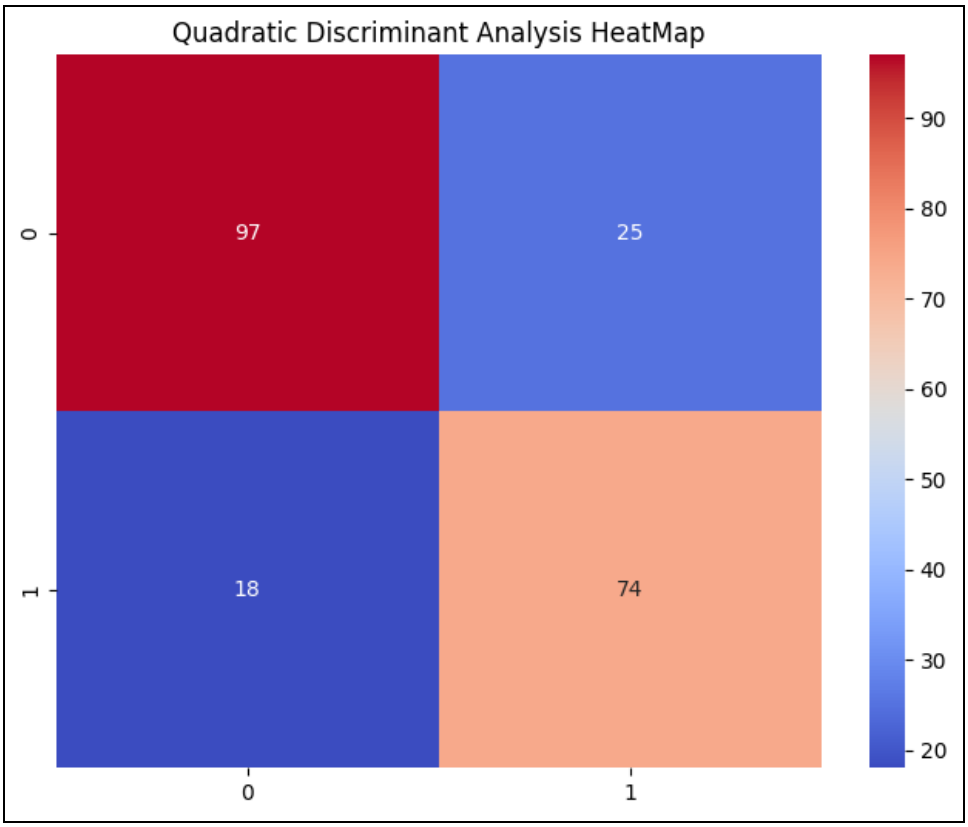
Logistic regression:



Logistic regression was performed, and the following results were obtained:

a) The confusion matrix for logistic regression is
   [[106 16]
   [22 70]]
b) Accuracy 0.822429906542056
c) Precision: 0.8220325744403391
d) Recall: 0.822429906542056
e) F1 Score: 0.8215679932794289

QDA(Quadratic Discriminant Analysis)



Quadratic Discriminant Analysis was performed, and the following results were obtained:

a) The confusion matrix for QDA is :

[[97 25]

[18 74]]

b) The accuracy for QDA is 0.7990654205607477

c) Precision: 0.8022057224007652

d) Recall: 0.7990654205607477

e) F1 Score: 0.7997800795240039

Comparison between Logistic Regression and Quadratic Discriminant Analysis

Model	Precision	Recall	F1 score	Accuracy (%)
Logistic regression	0.8220	0.8224	0.8215	82.24 %
Quadratic Discriminant Analysis	0.8022	0.7991	0.7998	79.90 %

On comparing, it could be seen that Logistic regression performs better.