Heart Attack Possibility Prediction
Milestone: Project Report

Group 18
Student 1 Karan Paresh
Student 2 Tianyu Yang

857-294-7660 (Tel of Karan Paresh)
617-368-0332 (Tel of Tianyu Yang)

samani.k@northeastern.edu
yang.tianyu@northeastern.edu

Percentage of Effort Contributed by Student 1: 50%
Percentage of Effort Contributed by Student2: 50%
Signature of Student 1: Karan Paresh
Signature of Student 2: Tianyu Yang
Submission Date: 12/13/2021

Problem setting:

The dataset contains 14 attributes which are responsible for predicting whether the patient may suffer from a heart disease or not. The "target" attribute indicates the presence of heart disease in the patient. It is a discrete value 0 = no/less chance of heart attack and 1 = more chance of heart attack. When the prediction made is accurate, we can not only avoid false diagnosis but also save human resources. Making the correct predictions can solve a lot of trouble. Applying machine learning algorithms for the medical prediction, we will be able to extricate plenty of human resource because we do not need the complex diagnosis procedure in medical sector. The algorithms that will be put into test are Logistic Regression, SVM, Naïve Bayes, Random Forest, ANN to output a binary number 1 or 0.

Problem definition: -

Heart disease can be treated right and its symptoms can be condensed with an accurate predicted result and thereby reducing the surgery cost and other medication expenses. The main objective of the project is to predict the possibilities of heart disease explicitly with few records and attributes. Decisions are generally in the data sets and databases which therefore assists the practitioners in making better decisions and in recommending the required treatment.

Data mining holds great latent in medical sector which empowers the health care industry to use the data and analytics to pin down errors and reduce the overall medical lobby spending.

Data source: -

The dataset contains 14 attributes which contributes in predicting the target variable, the target variable refers to the presence of heart disease in the patient. It is a discrete valued 0=no/less chance of heart attack and 1=more chance of heart attack.

URL: https://www.kaggle.com/nareshbhat/health-care-data-set-on-heart-attack-possibility

Data description: -

1. Age:- The age of the people from the dataset.

2. Sex:- The sex of a person and it has has only 2 possible values in this dataset: 1 - Male and 0 - Female.

3. cp(Chest pain type):- This column defines the chest pain severity in scale of 0-4.

— Value 0: asymptomatic

— Value 1: atypical angina

— Value 2: non-anginal pain

— Value 3: typical angina

4. trestbps(Resting blood pressure):-
An healthy has has a blood pressure of 80/120 mmHg , person is at high risk if it is above 180mmHg and lesser than 50mmHg.It has continuous values.

5. chol(Serum cholestrol):- The normal cholestrol level of a healthy person is between 125-200 mm/dl and it is considerable till 240 mm/dl and anything greater than this value will cause higher risk of heart attack and this feature has a continuous value.

6. fbs(Fasting blood sugar):- This feature has only two unique values - 1 if FBS is > 120 mg/dl otherwise 0.If the blood sugar is higher than the value mentioned the person is at high risk of getting heart attack.

7. restecg:- resting electrocardiographic results

— Value 0: normal

— Value 1: showing probable or definite left ventricular hypertrophy by Estes' criteria

— Value 2: having ST-T wave abnormality (T wave inversions and/or ST elevation or depression of > 0.05 mV)

8. thalach:- Maximum heart rate achieved and the normal heart rate of a person should be below 100 bpm anything above that is risky and person might tend to have a heart attack.

9. exang(exercise induced angina):- This feature has two values (1 = yes; 0 = no) means if a person faces angina due to exercise than the value is 1 else it is 0.

10. oldpeak(ST depression induced by exercise):- If the range is lesser than 1.5 its a high risk and if it is greter than 1.5 the person is at low risk of getting a heart attack.

11. slope:- the slope of the peak exercise ST segment

0: downsloping;

1: flat;

2: upsloping

when the value is 0 and 2 the possibility of heart attack is high and when it is 1 the chances of getting heart attack is less.

12. ca(number of major vessels colored by flouropsy):- This column has a discrete value from [0-3].

13. thal:- A blood disorder called thalassemia, it has discrete values
Value 1: normal blood flow
Value 2: fixed defect (no blood flow in some part of the heart)
Value 3: reversible defect (a blood flow is observed but it is not normal)

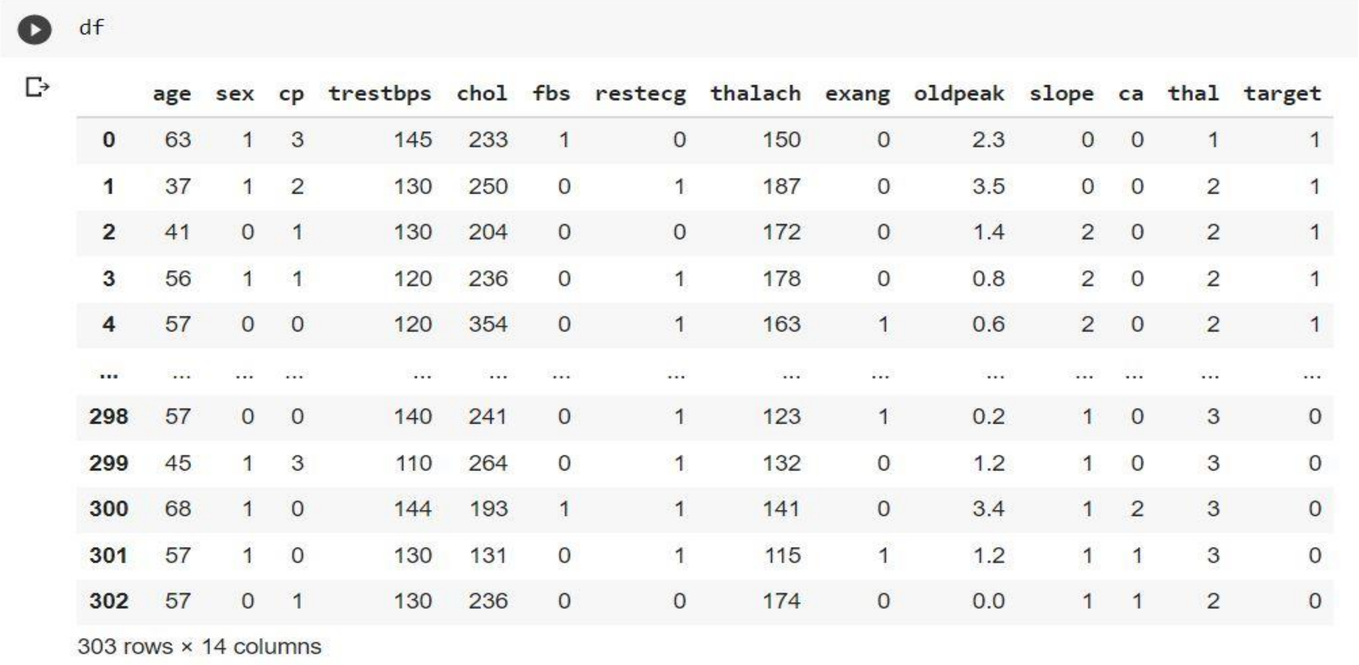
14. target:- This column has a discrete values
0 = less chance of heart attack, 1= more chance of heart attack.

Data description: -

The dataframe has 14 variables and 303 records, the data has both continuous and discrete variables.

Discrete variables = sex, cp (chest pain type), fbs (fasting blood sugar), restecg, exang (exercise induced angina), slope, ca (major vessels colour by flouropsy) , thal, target

Continuous variables = age, trestbps (resting blood pressure), chol (serum cholesterol), thalach, oldpeak (ST depression induced by angina).



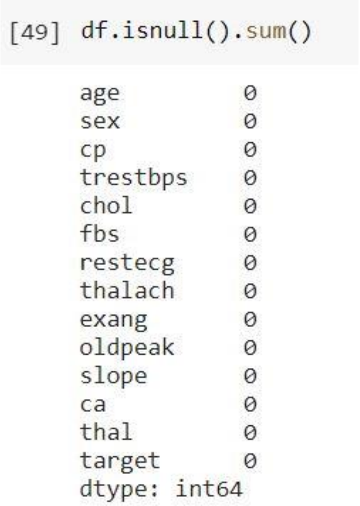
```
df
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	63	1	3	145	233	1	0	150	0	2.3	0	0	1	1
1	37	1	2	130	250	0	1	187	0	3.5	0	0	2	1
2	41	0	1	130	204	0	0	172	0	1.4	2	0	2	1
3	56	1	1	120	236	0	1	178	0	0.8	2	0	2	1
4	57	0	0	120	354	0	1	163	1	0.6	2	0	2	1
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
298	57	0	0	140	241	0	1	123	1	0.2	1	0	3	0
299	45	1	3	110	264	0	1	132	0	1.2	1	0	3	0
300	68	1	0	144	193	1	1	141	0	3.4	1	2	3	0
301	57	1	0	130	131	0	1	115	1	1.2	1	1	3	0
302	57	0	1	130	236	0	0	174	0	0.0	1	1	2	0

303 rows × 14 columns

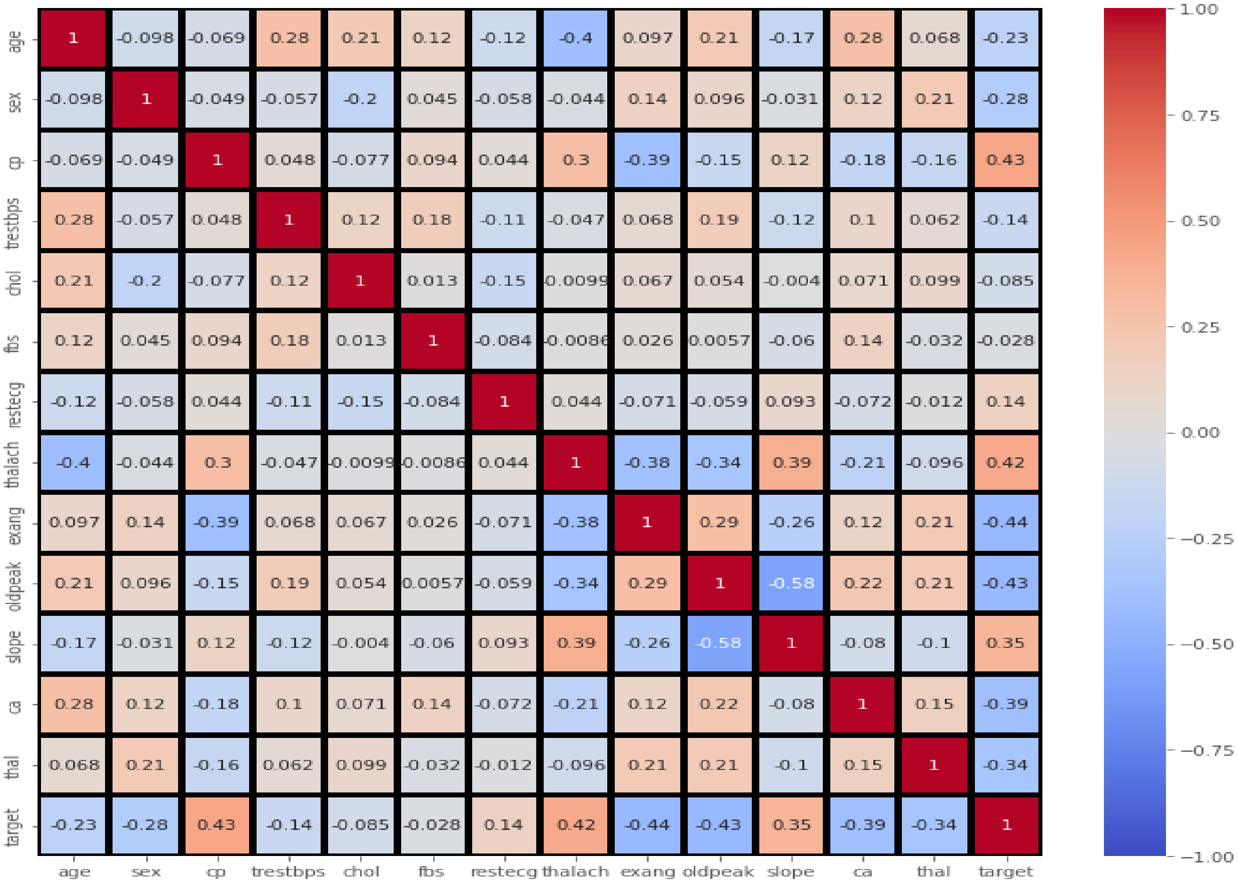
Null Values: -

The heart attack dataset contains zero null values and the data imputation on the null values aren't required. The label encoding is also not required for any of the features.



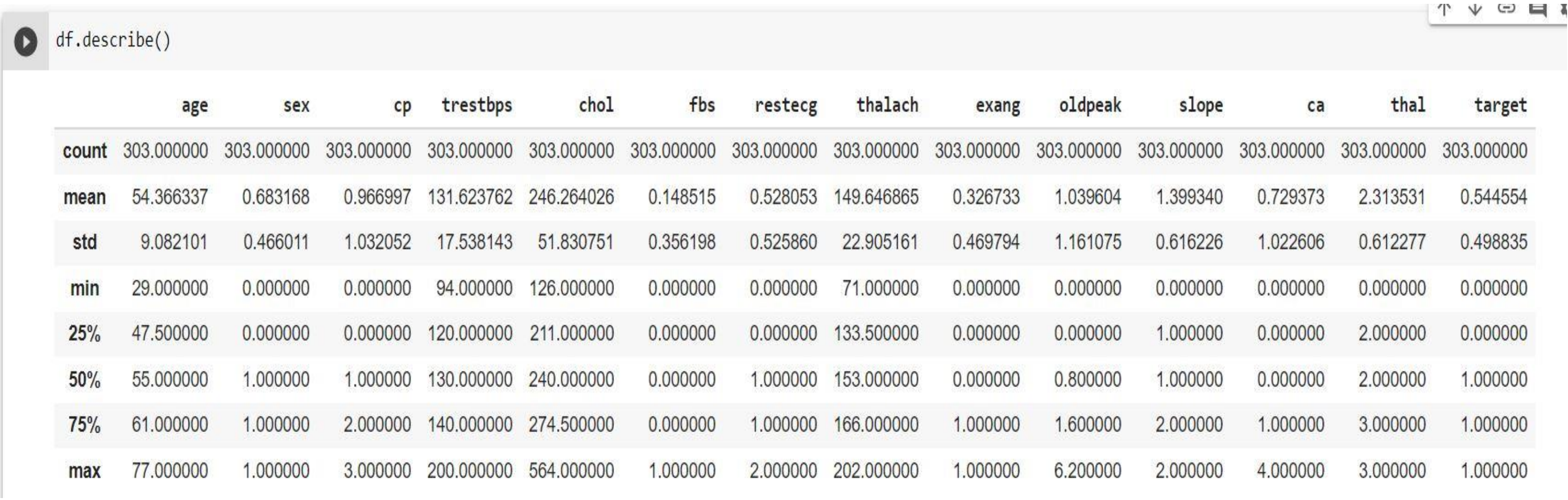
```
[49] df.isnull().sum()

     age         0
     sex         0
     cp          0
     trestbps    0
     chol        0
     fbs         0
     restecg     0
     thalach     0
     exang       0
     oldpeak     0
     slope       0
     ca          0
     thal        0
     target      0
     dtype: int64
```

Feature correlation matrix: -



We can see there is a positive correlation between chest pain (cp) & target (our predictor). This makes sense since, the greater amount of chest pain results in a greater chance of having heart disease. Cp (chest pain), is a ordinal feature with 4 values. In addition, we see a negative correlation between exercise induced angina (exang) & our predictor. This makes sense because when you exercise, your heart requires more blood, but narrowed arteries slow down blood flow. This matrix shows the correlation of every variable with each other and there by we can decide on how many features are needed to predict the target variable. The feature selection can also be done using backward and forward selection.

Summary of the data: -



```
df.describe()
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
count	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000
mean	54.366337	0.683168	0.966997	131.623762	246.264026	0.148515	0.528053	149.646865	0.326733	1.039604	1.399340	0.729373	2.313531	0.544554
std	9.082101	0.466011	1.032052	17.538143	51.830751	0.356198	0.525860	22.905161	0.469794	1.161075	0.616226	1.022606	0.612277	0.498835
min	29.000000	0.000000	0.000000	94.000000	126.000000	0.000000	0.000000	71.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	47.500000	0.000000	0.000000	120.000000	211.000000	0.000000	0.000000	133.500000	0.000000	0.000000	1.000000	0.000000	2.000000	0.000000
50%	55.000000	1.000000	1.000000	130.000000	240.000000	0.000000	1.000000	153.000000	0.000000	0.800000	1.000000	0.000000	2.000000	1.000000
75%	61.000000	1.000000	2.000000	140.000000	274.500000	0.000000	1.000000	166.000000	1.000000	1.600000	2.000000	1.000000	3.000000	1.000000
max	77.000000	1.000000	3.000000	200.000000	564.000000	1.000000	2.000000	202.000000	1.000000	6.200000	2.000000	4.000000	3.000000	1.000000

Initial insights of the data: -

Figure 1: -

The pie chart represents the percentage of people who may suffer from the heart disease and percentage of those who may not. The result shows that 45.5% people do not suffer from heart attack and 54.5% people suffer from heart attack.

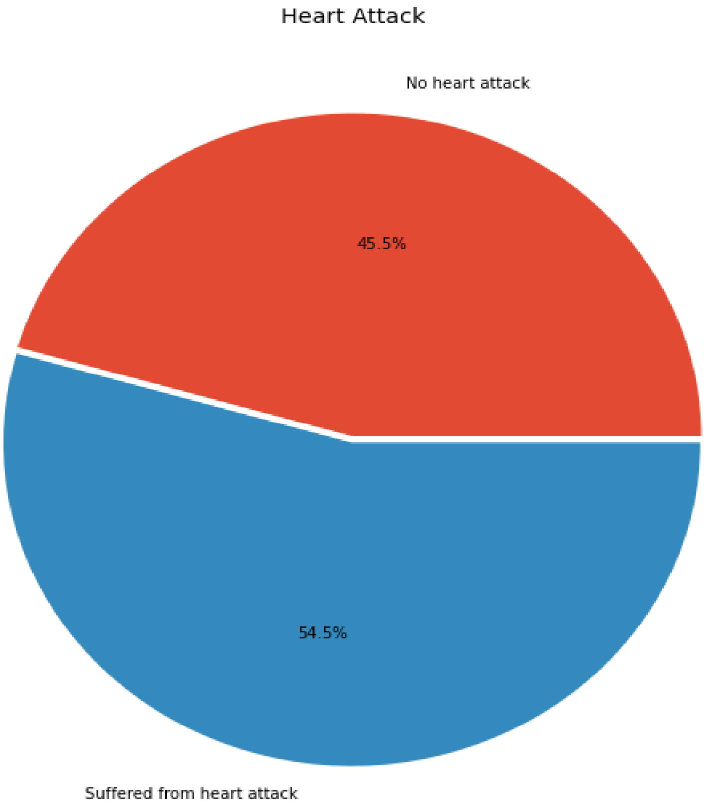


Figure: 1

Figure 2: -

The displot shows that the Heart attack is very common in the seniors which is composed of age group 60 and above and common among adults which belong to the age group of 41 to 60, but it's rare among the age group of 19 to 40 and very rare among the age group of 0 to 18.

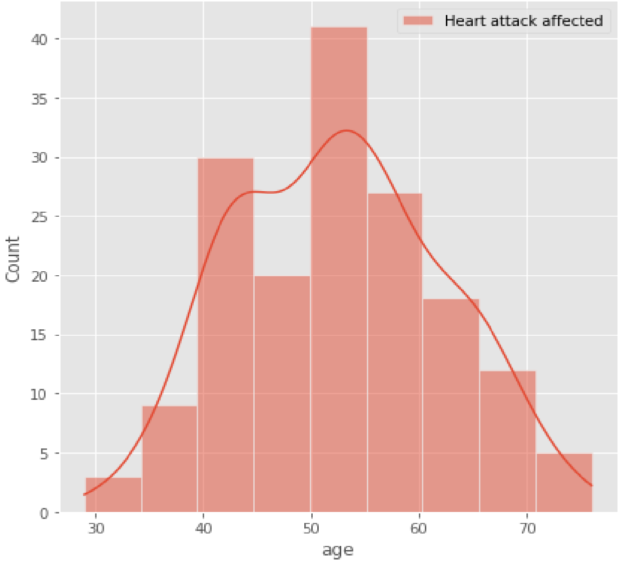


Figure: 2

Figure 3: -

The boxplot shows that the heartrate (thalach) is gradually higher for those who are suffering from the heart disease and on the other hand as the severity of chest pain (cp) increases the mean heartrate is also increasing.

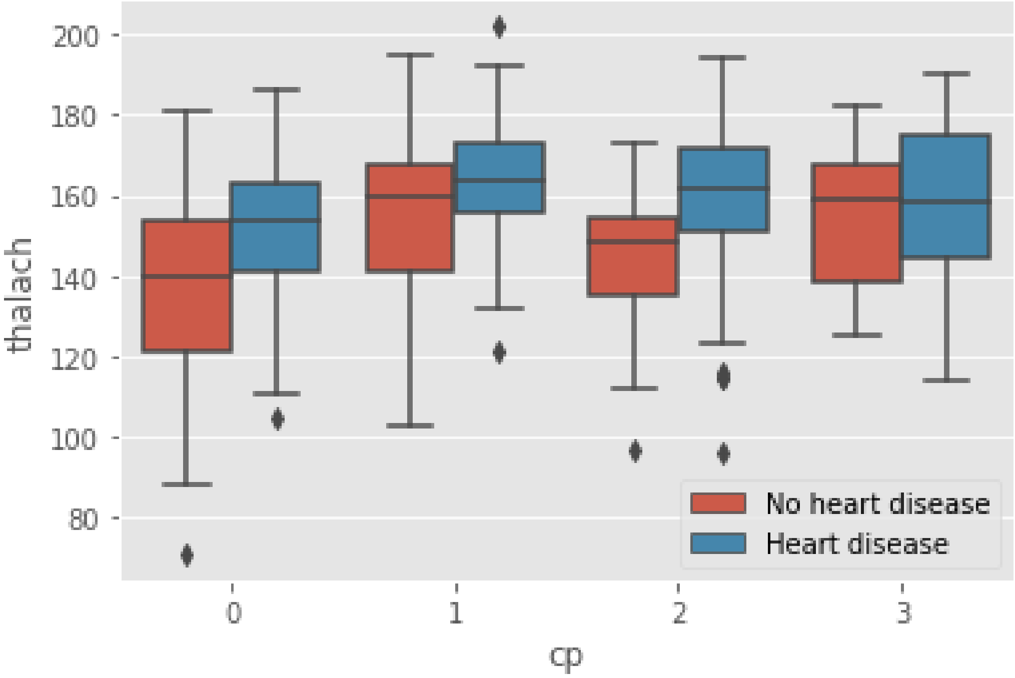


Figure: 3

Figure 4:

Visual illustrates the age distribution of male and female and the mean age of both the genders is almost same and the kernel density is higher for the age groups between 50 to 70 years.

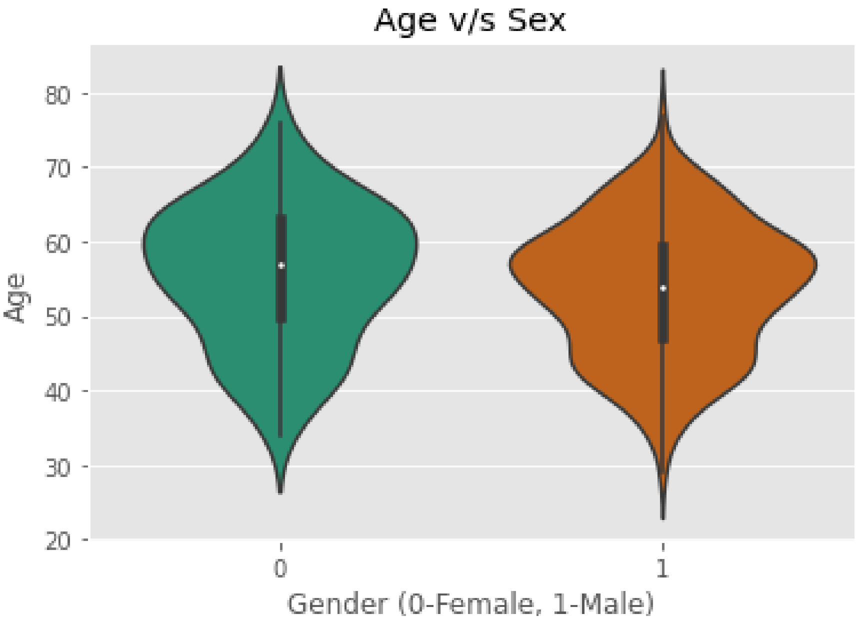


Figure:4

Data exploration and visualization

Countplots:

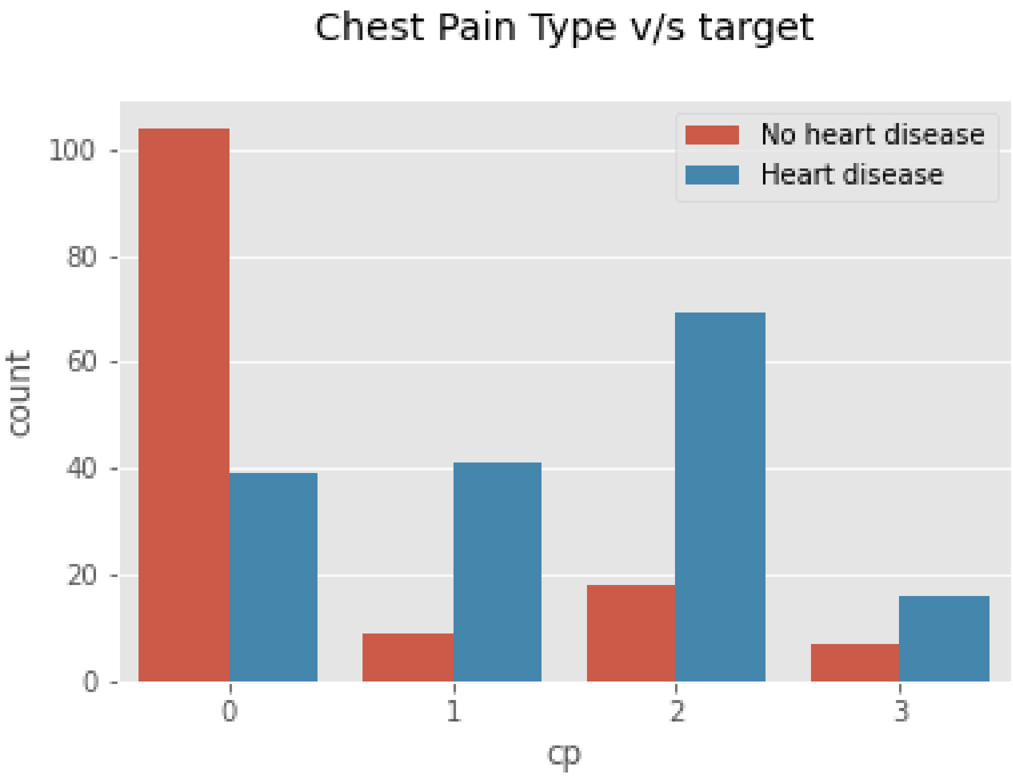


Figure 5:

There are four types of chest pain, asymptomatic, atypical angina, nonanginal pain and typical angina. Most of the Heart disease patients are found to have atypical anginal and non anginal chest pain and very few have typical angina. As the severity increases the number of people who are suffering from heart disease are more compared to those who are not suffering from heart disease. These group of people might show symptoms like indigestion, flu or a strained chest muscle.Heart attack, involves, blockage of blood flow to your heart and possible damage to the heart muscle.

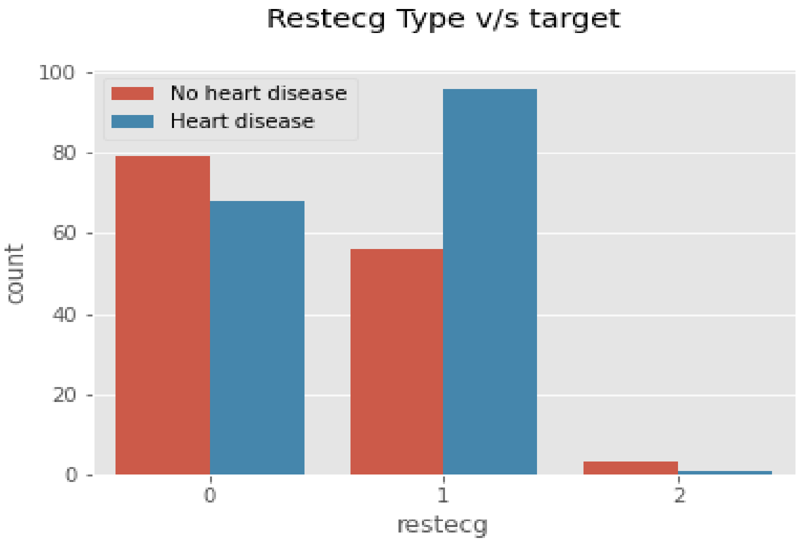


Figure 6:

There are three types of restecg states Value 0: normal, Value 1: showing probable or definite left ventricular hypertrophy by Estes' criteria, Value 2: having ST-T wave abnormality (T wave inversions and/or ST elevation or depression of > 0.05 mV).As the severity increases the number of people who are suffering from heart disease are more compared to those who are not suffering from heart disease.The result shows that people who are suffering from the heart disease have higher ecg value and maximum people has 'value' = 1 and very few suffering from heart disease has 'value'=2.

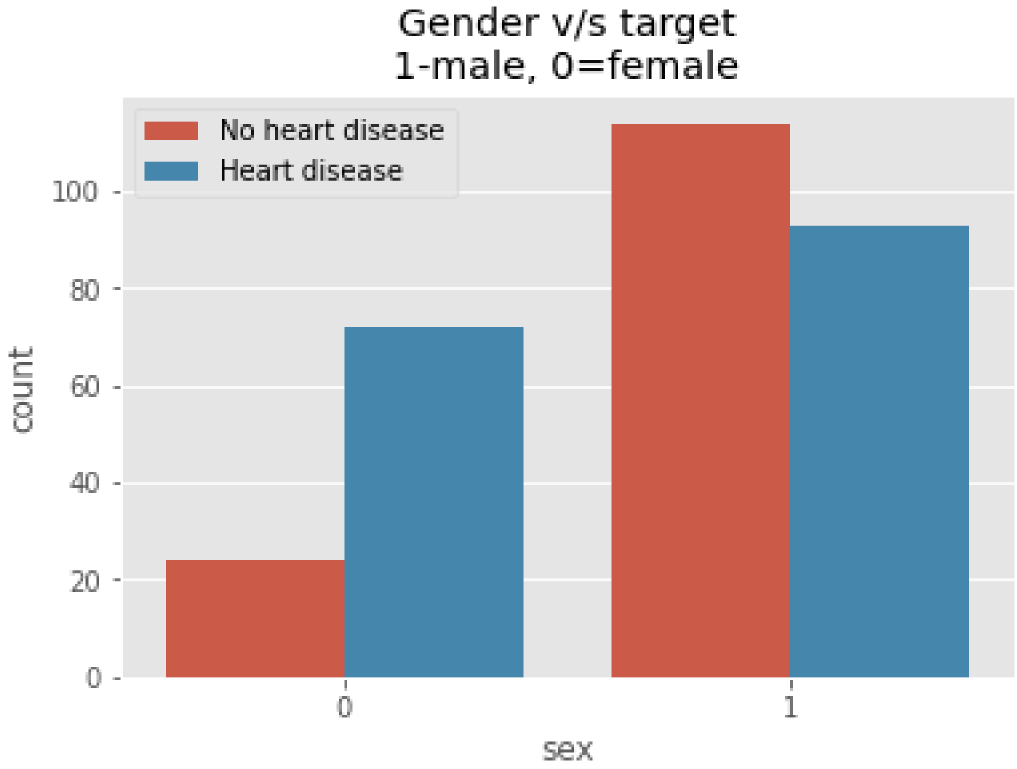


figure 7:

According to this dataset males are more susceptible to get Heart Disease than females. Men experience heart attacks more than women. Sudden Heart Attacks are experienced by men between 70% — 89%. Woman may experience a heart attack with no chest pressure at all, they usually experience nausea or vomiting which are often confused with acid reflux or the flu.

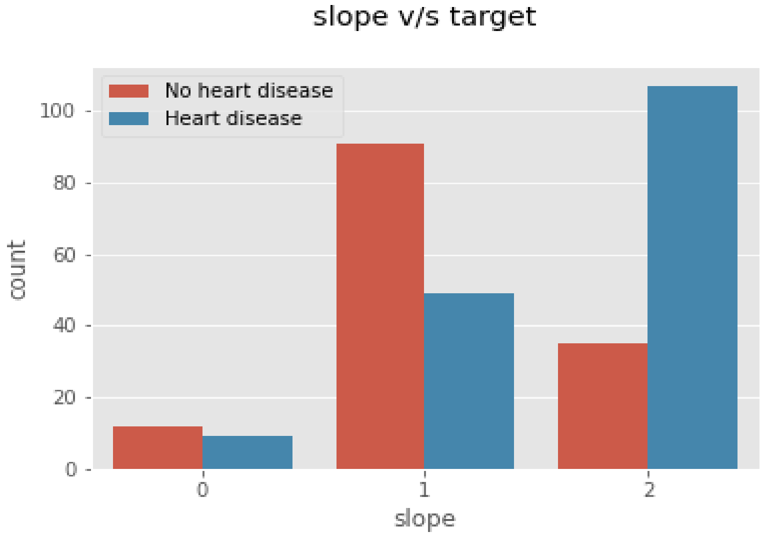


Figure 8:

The slope of the peak exercise ST segment has three values 0: down-sloping; 1: flat; 2: upsloping, when the value is 0 and 2 the possibility of heart attack is high and when it is 1 the chances of getting heart attack is less. The graph shows the people who are suffering from the heart disease has the slope value of either 0 or 2 and maximum people who do not suffer from the heart disease has the slope value 1.

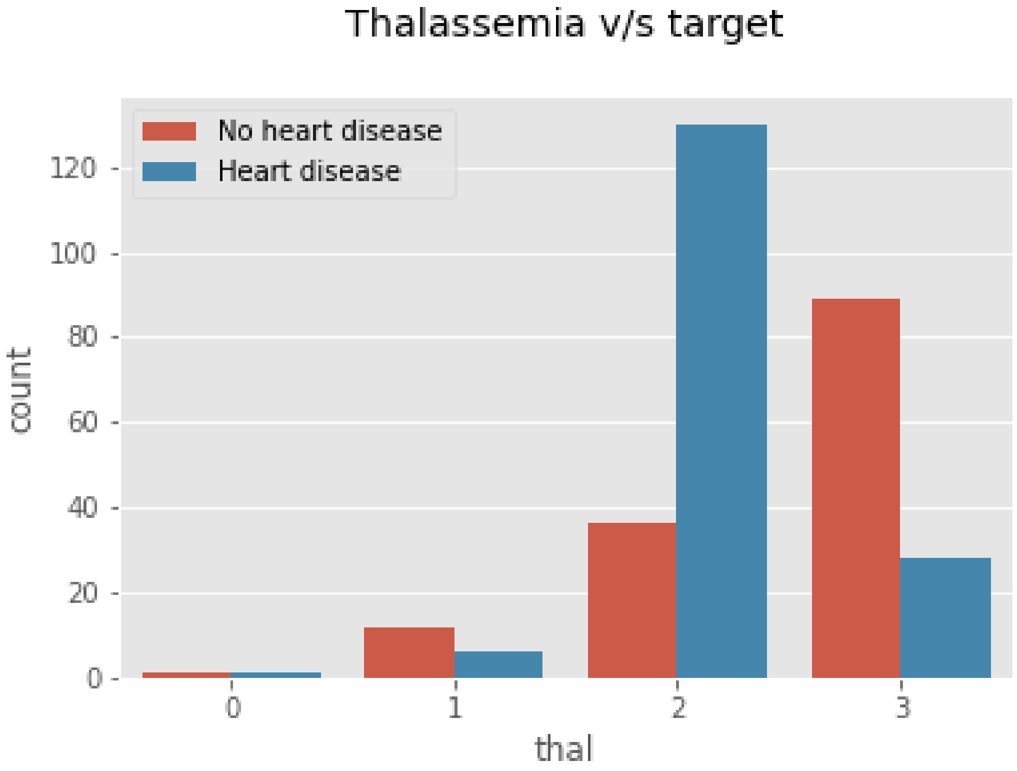


Figure 9:

A blood disorder called thalassemia, it has discrete values Value 1: normal blood flow Value 2: fixed defect (no blood flow in some part of the heart) Value 3: reversible defect (a blood flow is observed but it is not normal).The graph shows that people who are suffering from heart disease has a defect value 2 and 3 and those arent suffering from heart disease has a value 1 and 3.

Jointplots:

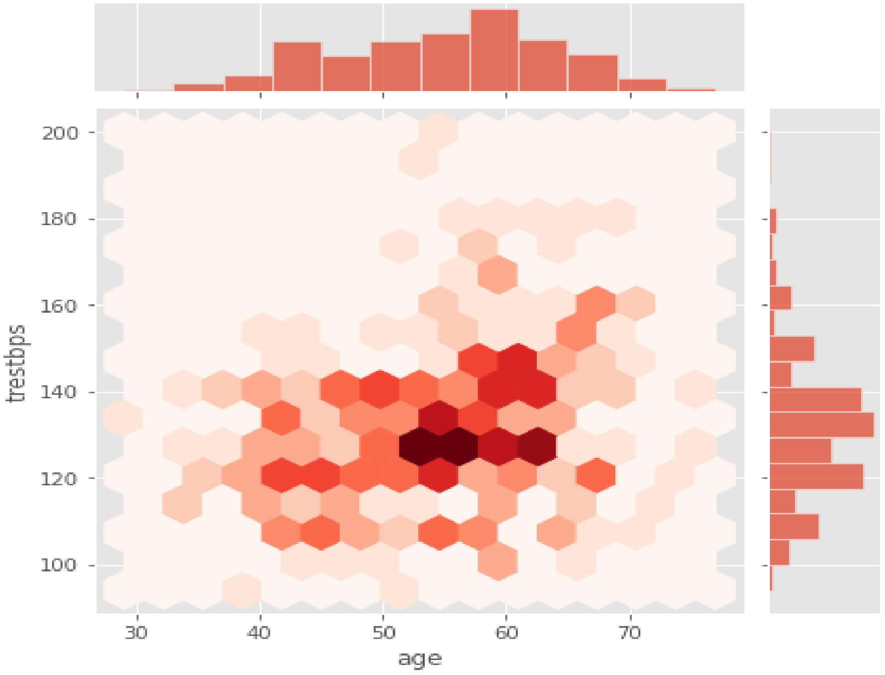


Figure 10:

Joint plots in seaborn helps us to understand the trend seen among two features. As observed from the above plot we can see that most of the Heart diseased patients in their age of upper 50s or lower 60s tend to have trestbps higher than 120mmHg.

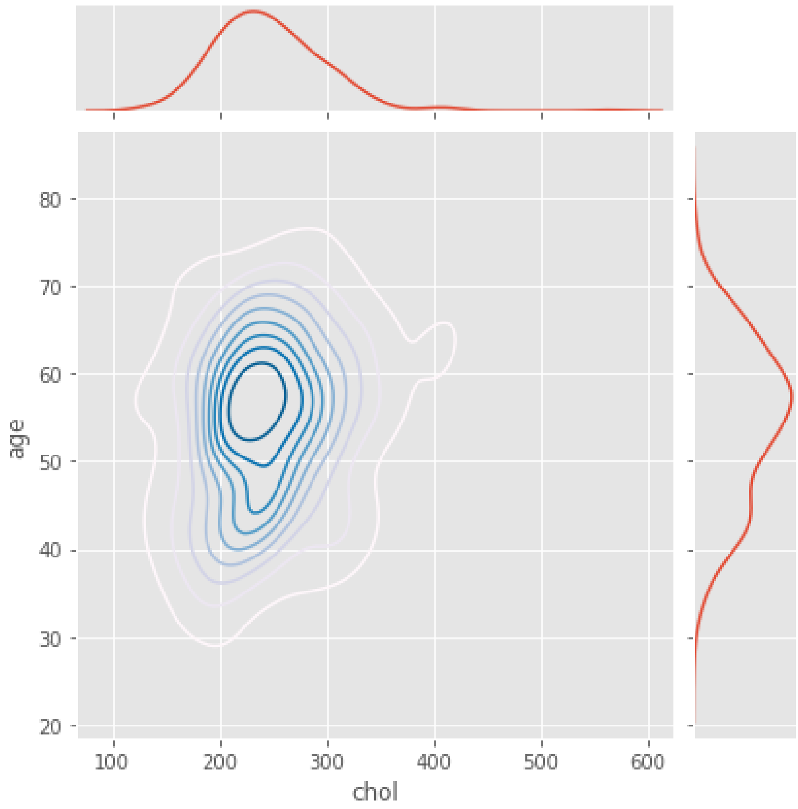


Figure 11:

Joint plots in seaborn helps us to understand the trend seen among two features. As observed from the above plot we can see that most of the Heart diseased patients in their age of upper 50s or lower 60s tend to have Cholesterol between 200mg/dl to 300mg/dl.

Boxplot:

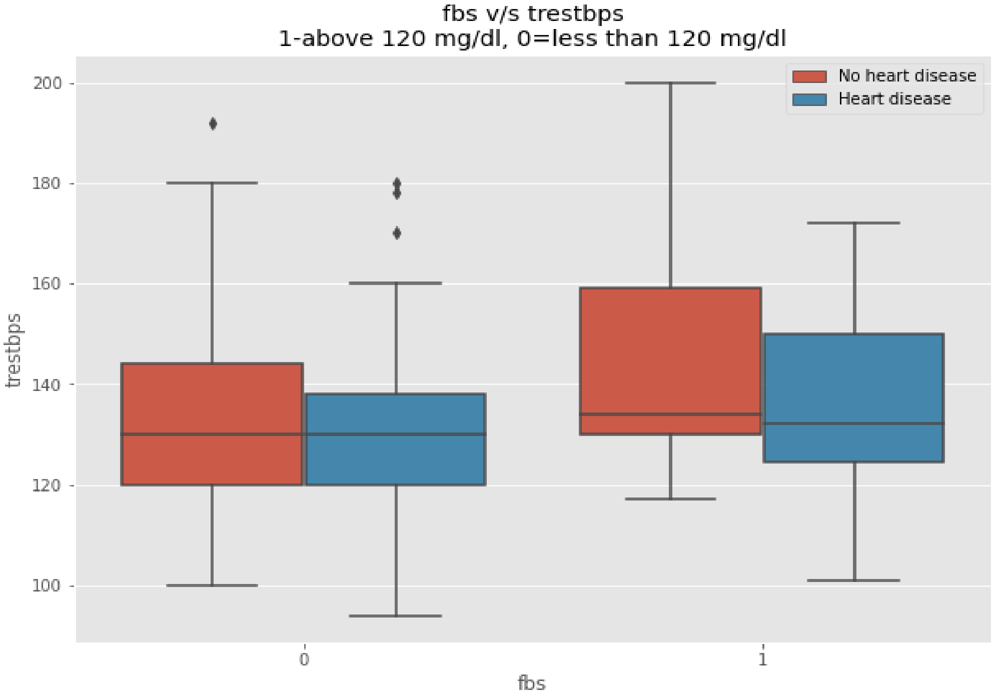


Figure 12:

The boxplot shows that the people who are diabetic (fasting blood sugar) has higher resting blood pressure which concludes that these group of people tend to have higher chances of having heart disease.

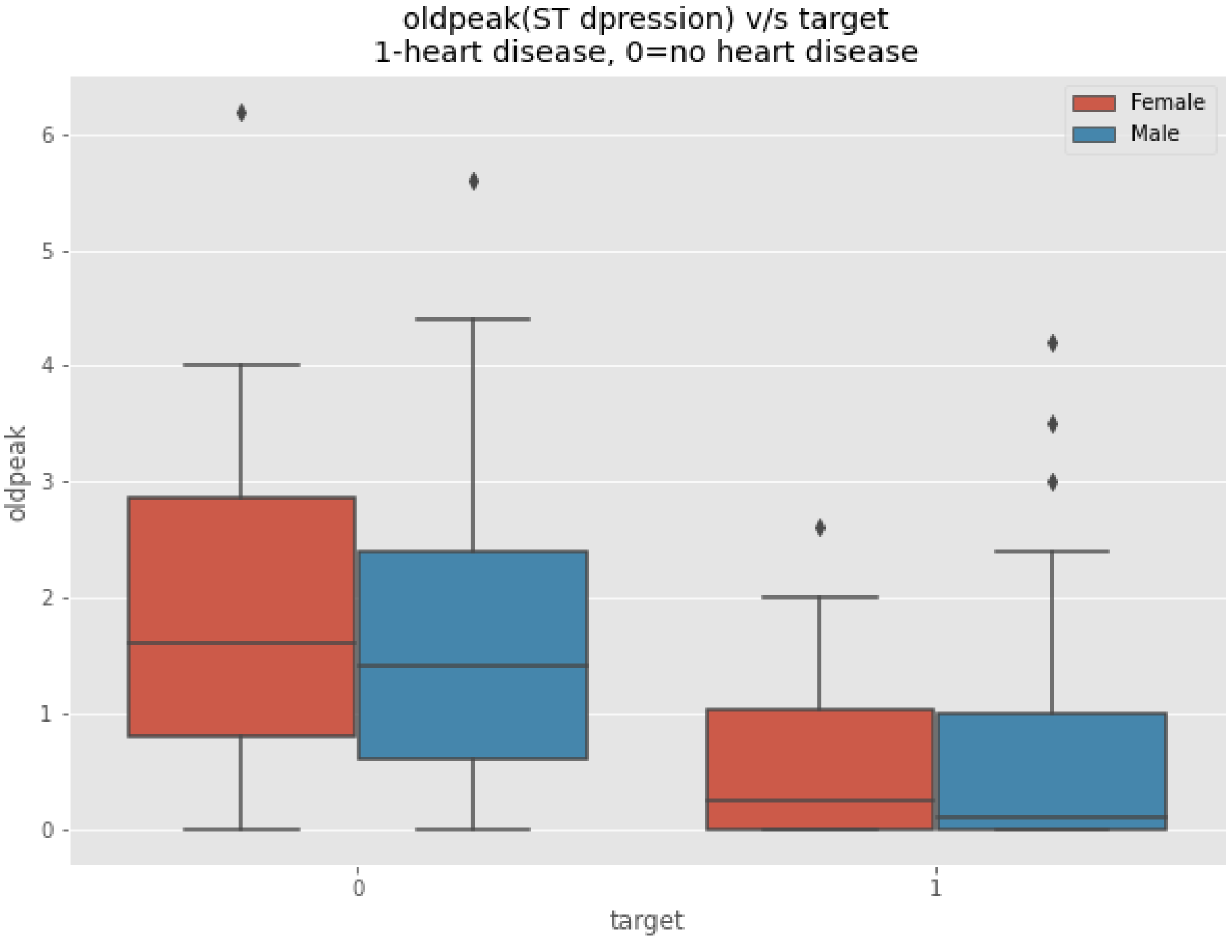


Figure 13:

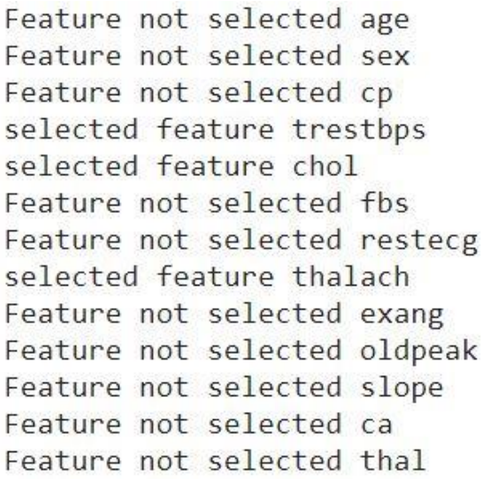
The boxplot shows that low ST Depression (oldpeak) yields people at greater risk for heart disease. While a high ST depression is considered normal & healthy.If the range is lesser than 1.5 its a high risk and if it is greater than 1.5 the person is at low risk of getting a heart attack.

Model selection and implementation

Feature selection: -

1. Lasso L1 penalty feature selection: -

The L1 penalty feature selection is used to determine the features that are needed that is the once that influence the target variable the most are selected.The features whose coefficients shrank to zero are removed and the once which didn't are selected.

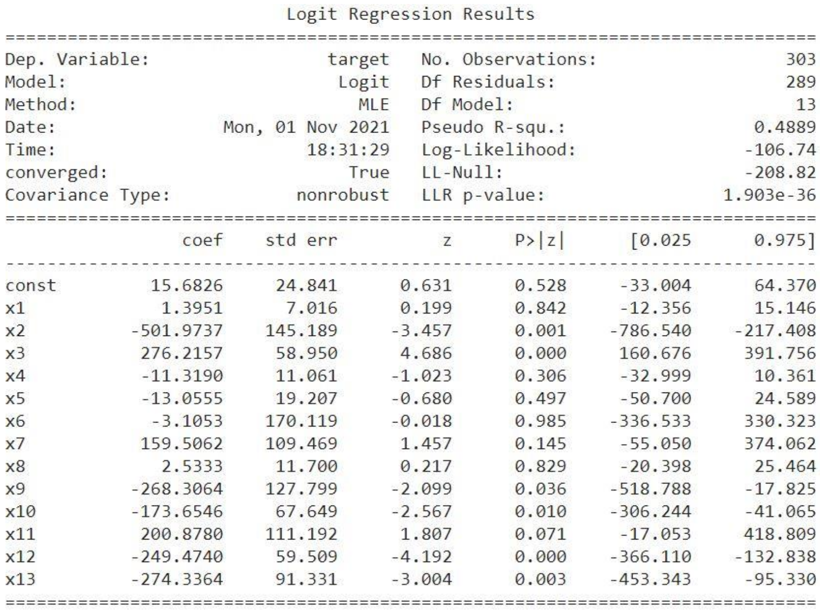


```
Feature not selected age
Feature not selected sex
Feature not selected cp
selected feature trestbps
selected feature chol
Feature not selected fbs
Feature not selected restecg
selected feature thalach
Feature not selected exang
Feature not selected oldpeak
Feature not selected slope
Feature not selected ca
Feature not selected thal
```

The features which are selcted by the L1 penalty are "trestbps", "chol" and "thalach".

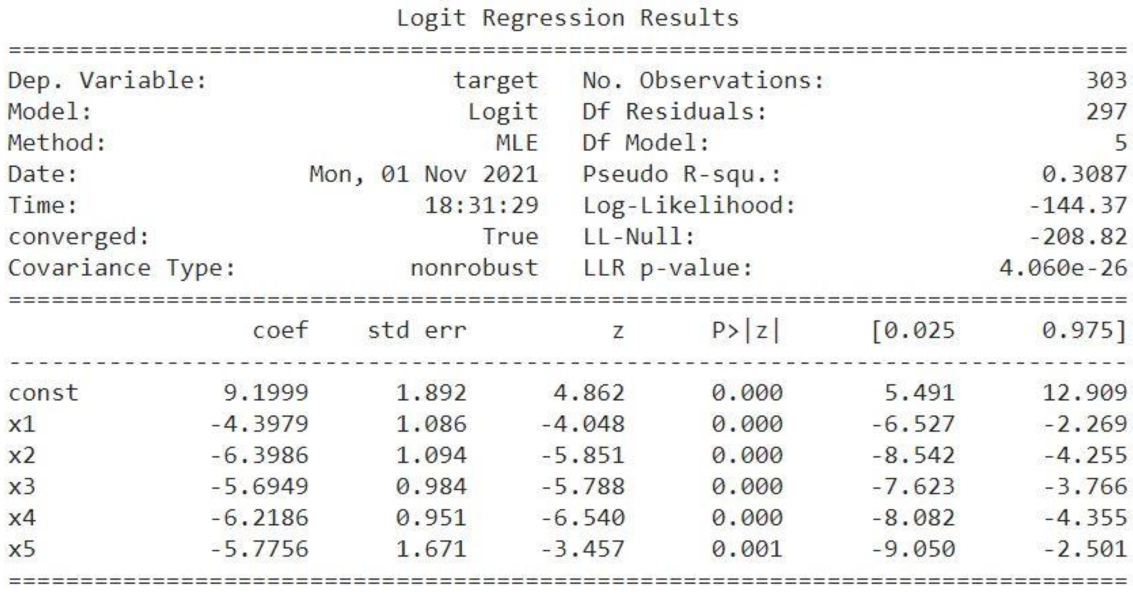
2. Backward feature selection: -

We use the other method known as backward feature selection, the selection of a feature her depends on its p-value. In this method at first all the features are selected and the p-value for every feature is noted, if the p-value of any variable greater than that of "0.05" that feature removed from the group and features with p-value lesser than "0.05" are kept and tested again and we continue until we reach a point where all the feature has p-value lesser than "0.05", once we achieve this we stop and select those features.



```
                           Logit Regression Results                           
==============================================================================
Dep. Variable:                 target   No. Observations:                  303
Model:                          Logit   Df Residuals:                      289
Method:                           MLE   Df Model:                           13
Date:                Mon, 01 Nov 2021   Pseudo R-squ.:                  0.4889
Time:                        18:31:29   Log-Likelihood:                -106.74
converged:                       True   LL-Null:                       -208.82
Covariance Type:            nonrobust   LLR p-value:                 1.903e-36
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const         15.6826     24.841      0.631      0.528     -33.004      64.370
x1             1.3951      7.016      0.199      0.842     -12.356      15.146
x2          -501.9737    145.189     -3.457      0.001    -786.540    -217.408
x3           276.2157     58.950      4.686      0.000     160.676     391.756
x4           -11.3190     11.061     -1.023      0.306     -32.999      10.361
x5           -13.0555     19.207     -0.680      0.497     -50.700      24.589
x6            -3.1053    170.119     -0.018      0.985    -336.533     330.323
x7           159.5062    109.469      1.457      0.145     -55.050     374.062
x8             2.5333     11.700      0.217      0.829     -20.398      25.464
x9          -268.3064    127.799     -2.099      0.036    -518.788     -17.825
x10         -173.6546     67.649     -2.567      0.010    -306.244     -41.065
x11          200.8780    111.192      1.807      0.071     -17.053     418.809
x12         -249.4740     59.509     -4.192      0.000    -366.110    -132.838
x13         -274.3364     91.331     -3.004      0.003    -453.343     -95.330
==============================================================================
```

Final result,



```
                           Logit Regression Results
==============================================================================
Dep. Variable:                 target   No. Observations:                  303
Model:                          Logit   Df Residuals:                      297
Method:                           MLE   Df Model:                            5
Date:                Mon, 01 Nov 2021   Pseudo R-squ.:                  0.3087
Time:                        18:31:29   Log-Likelihood:                -144.37
converged:                       True   LL-Null:                       -208.82
Covariance Type:            nonrobust   LLR p-value:                 4.060e-26
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const          9.1999      1.892      4.862      0.000       5.491      12.909
x1            -4.3979      1.086     -4.048      0.000      -6.527      -2.269
x2            -6.3986      1.094     -5.851      0.000      -8.542      -4.255
x3            -5.6949      0.984     -5.788      0.000      -7.623      -3.766
x4            -6.2186      0.951     -6.540      0.000      -8.082      -4.355
x5            -5.7756      1.671     -3.457      0.001      -9.050      -2.501
==============================================================================
```

The Selected features are 'sex', 'exang', 'oldpeak', 'ca' and 'thal'. There is a difference when we use a different method for feature selection. There is no particular method it all depends on the model and the data, so we need to try different methods. We can also use "ridge regression", "forward selection" and etc.

Model selection: -

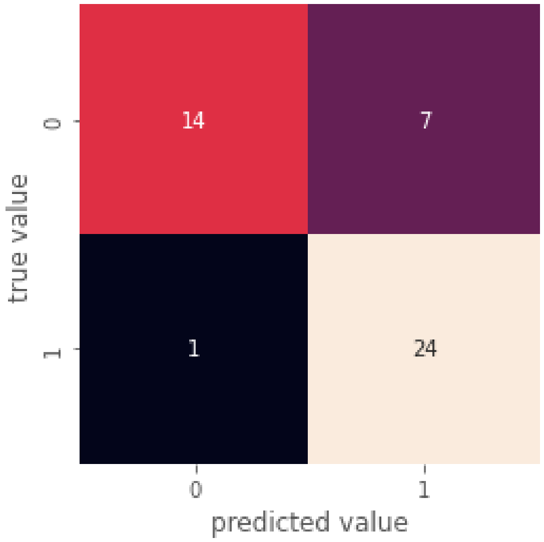
1. Gaussian Naïve Bayes: -

We use the gaussian naive bayes model to classify the class whether the patient will suffer from "Heart attack" or not. We use this model because every feature in this data is independent of each other and its relation to the target variable is also independent. The model is able to predict the results with 82.6% accuracy which is relatively decent.

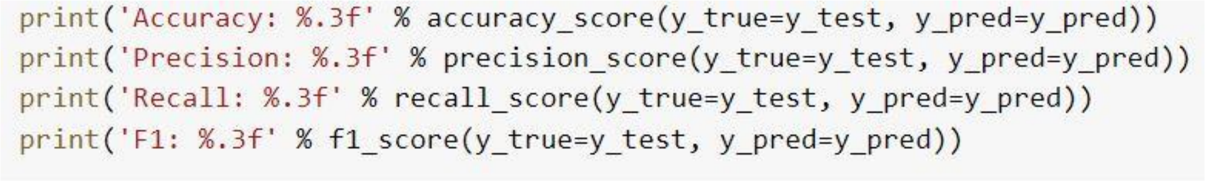
Evaluation: -

Confusion matrix and accuracy score: -

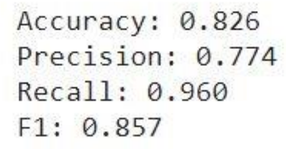
Target classes = 1- High chances Heart attack, 0- Less chances of heart attack.



Accuracy Scores: -



```
print('Accuracy: %.3f' % accuracy_score(y_true=y_test, y_pred=y_pred))
print('Precision: %.3f' % precision_score(y_true=y_test, y_pred=y_pred))
print('Recall: %.3f' % recall_score(y_true=y_test, y_pred=y_pred))
print('F1: %.3f' % f1_score(y_true=y_test, y_pred=y_pred))
```



```
Accuracy: 0.826
Precision: 0.774
Recall: 0.960
F1: 0.857
```

Accuracy = (TP+TN)/(TP+TN+FP+FN)

Precision = (TP)/(TP+FP)

Recall = (TP)/(TP+FN)

F1 = (2*Recall*Precison)/(Recall+Precision).

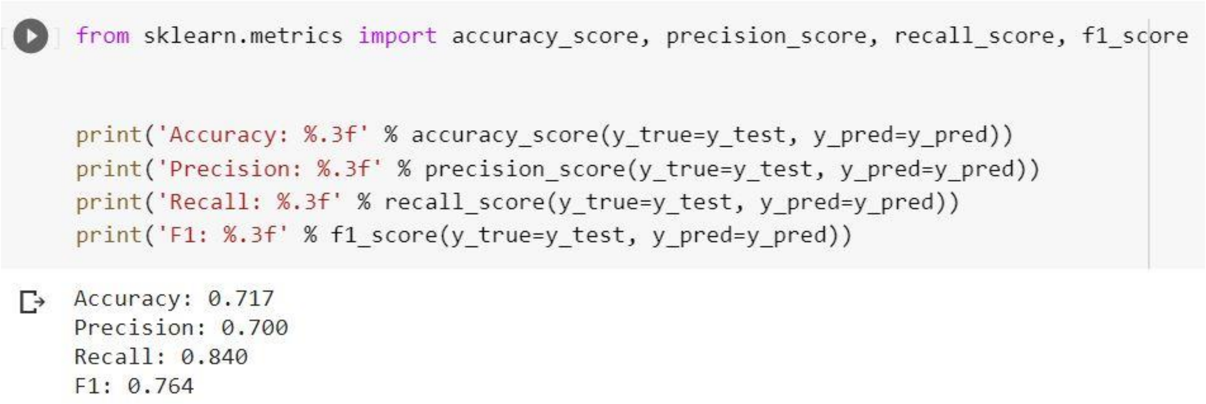
The model provides us with a decent accuracy score in distinguishing the class of a patient which helps in determining the chances of heart attack without going through the medical test by just using the values of the data of a patient.

Performance Evaluation and Interpretation

Modeling: -

1. Logistic Regression: -

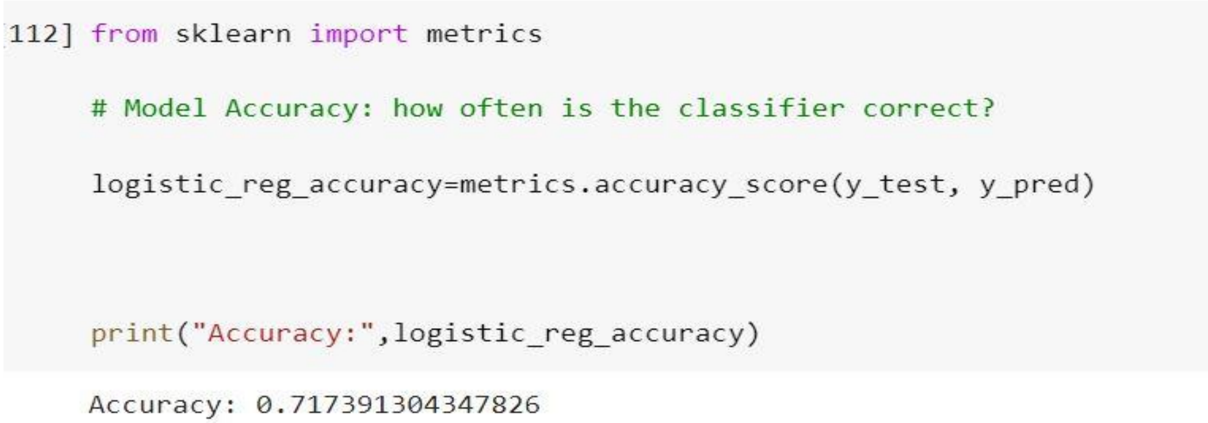
The first model which we are using is the logistic regression model.Using the metrics we get the accuracy, precision, Recall and F1-score .



```
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score


print('Accuracy: %.3f' % accuracy_score(y_true=y_test, y_pred=y_pred))
print('Precision: %.3f' % precision_score(y_true=y_test, y_pred=y_pred))
print('Recall: %.3f' % recall_score(y_true=y_test, y_pred=y_pred))
print('F1: %.3f' % f1_score(y_true=y_test, y_pred=y_pred))
```

```
Accuracy: 0.717
Precision: 0.700
Recall: 0.840
F1: 0.764
```

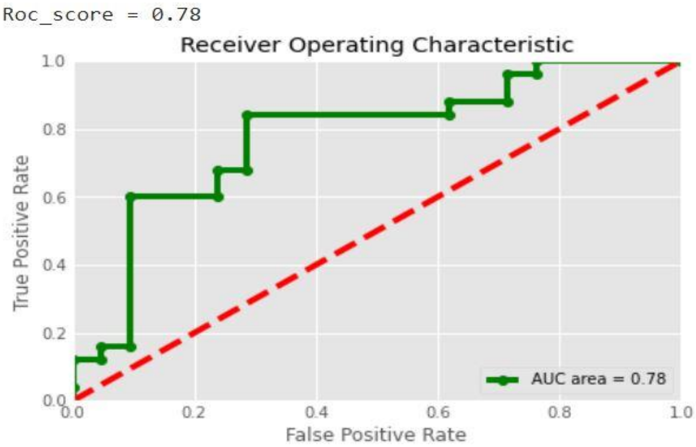
The accuracy score for the model is 71.7%.



```
112] from sklearn import metrics

     # Model Accuracy: how often is the classifier correct?

     logistic_reg_accuracy=metrics.accuracy_score(y_test, y_pred)



     print("Accuracy:",logistic_reg_accuracy)
```

```
Accuracy: 0.717391304347826
```

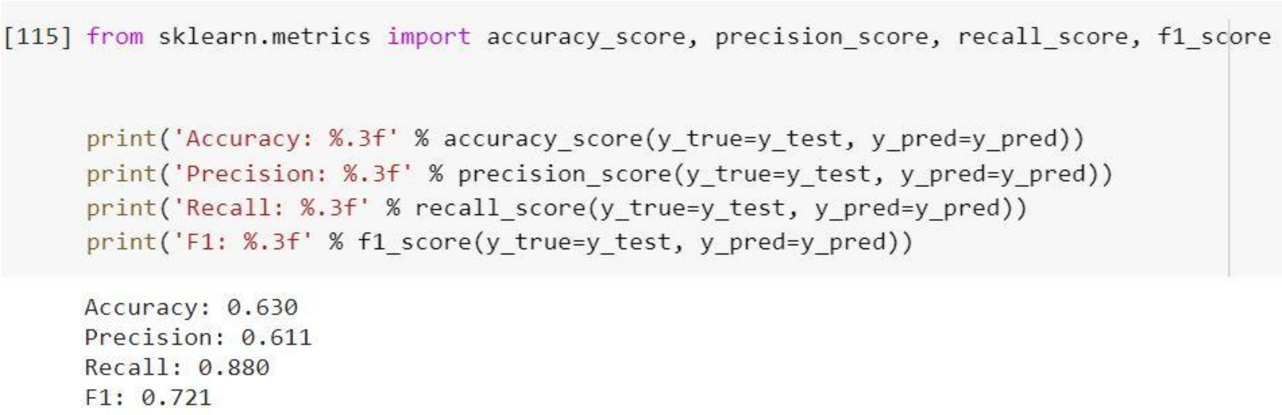
The Roc_auc score for the model is 78%.



```
Roc_score = 0.78
```

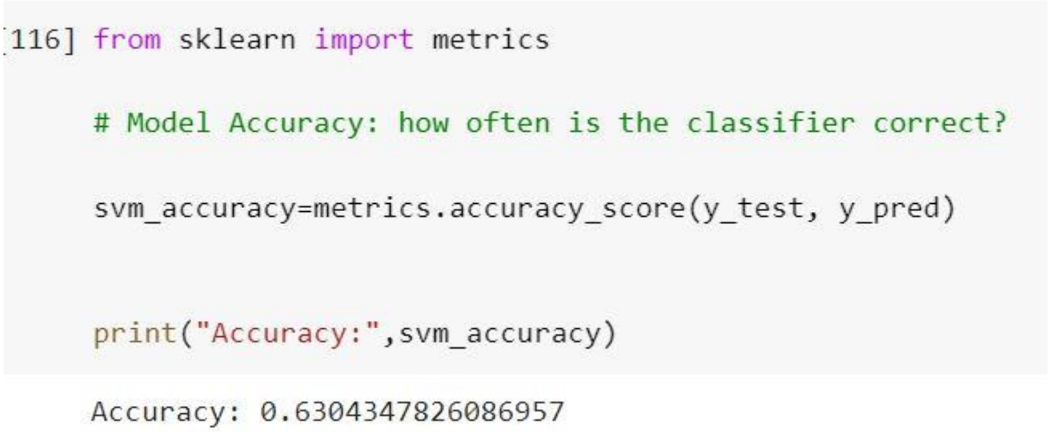
2. <u>Support Vector Machine: -</u>

The second classification model which we are using is Support Vector Machine. Using the metrics we get the accuracy, precision, Recall and F1-score .



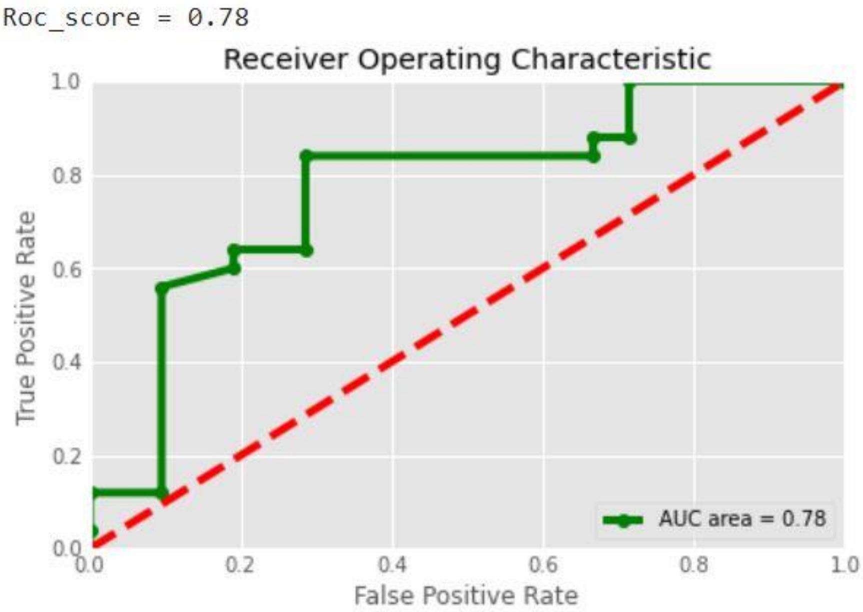
```
[115] from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score


      print('Accuracy: %.3f' % accuracy_score(y_true=y_test, y_pred=y_pred))
      print('Precision: %.3f' % precision_score(y_true=y_test, y_pred=y_pred))
      print('Recall: %.3f' % recall_score(y_true=y_test, y_pred=y_pred))
      print('F1: %.3f' % f1_score(y_true=y_test, y_pred=y_pred))

      Accuracy: 0.630
      Precision: 0.611
      Recall: 0.880
      F1: 0.721
```

The accuracy score for the model is 63%.



```
116] from sklearn import metrics

     # Model Accuracy: how often is the classifier correct?

     svm_accuracy=metrics.accuracy_score(y_test, y_pred)


     print("Accuracy:",svm_accuracy)

     Accuracy: 0.6304347826086957
```

The Roc_auc score for the model is 77%.

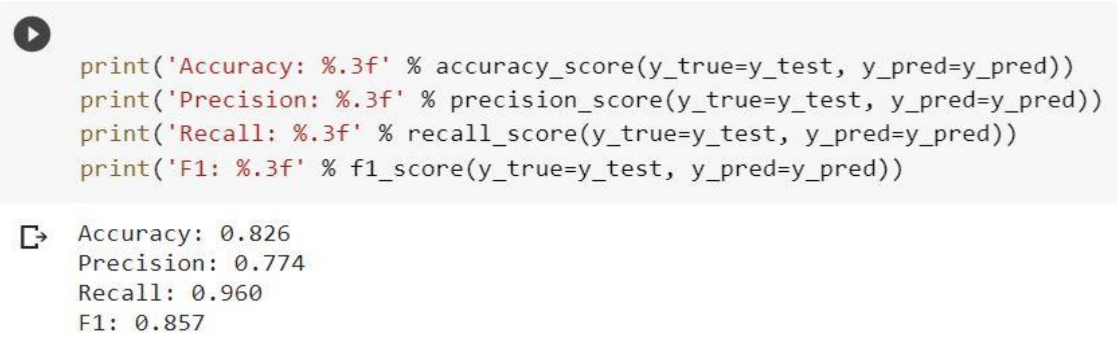


```
Roc_score = 0.78
```

3. Gaussian Naïve Bayes: - (Best Model)

The third model which we are using is Gaussian Naive Bayes. Using the metrics we get the accuracy, precision, Recall and F1-score .

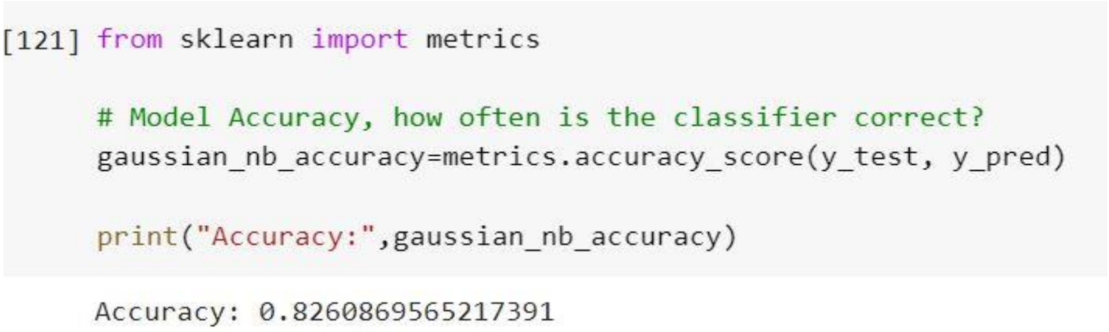
We use the gaussian naive bayes model to classify the class whether the patient will suffer from "Heart attack" or not. We use this model because every feature in this data is independent of each other and its relation to the target variable is also independent. The model is able to predict the results with 82.6% accuracy which is relatively decent.



```
print('Accuracy: %.3f' % accuracy_score(y_true=y_test, y_pred=y_pred))
print('Precision: %.3f' % precision_score(y_true=y_test, y_pred=y_pred))
print('Recall: %.3f' % recall_score(y_true=y_test, y_pred=y_pred))
print('F1: %.3f' % f1_score(y_true=y_test, y_pred=y_pred))
```

```
Accuracy: 0.826
Precision: 0.774
Recall: 0.960
F1: 0.857
```

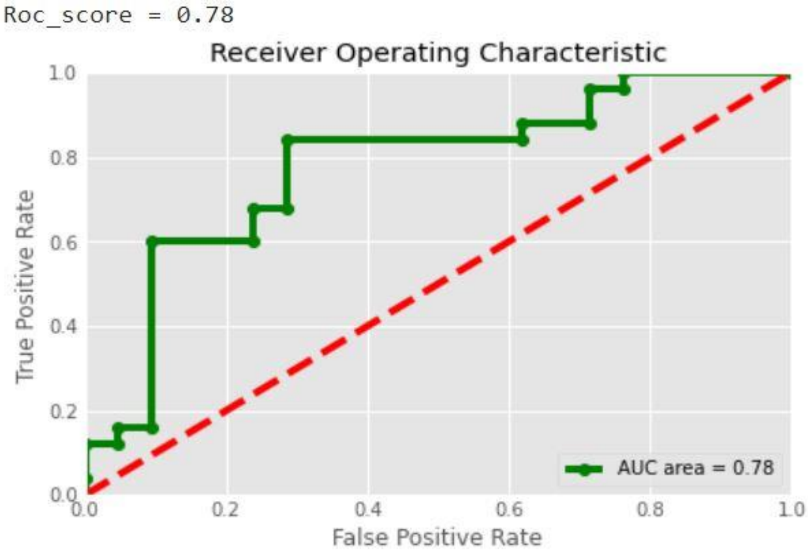
The accuracy score for the model is 82.6%.



```
[121] from sklearn import metrics

      # Model Accuracy, how often is the classifier correct?
      gaussian_nb_accuracy=metrics.accuracy_score(y_test, y_pred)

      print("Accuracy:",gaussian_nb_accuracy)
```

```
      Accuracy: 0.8260869565217391
```

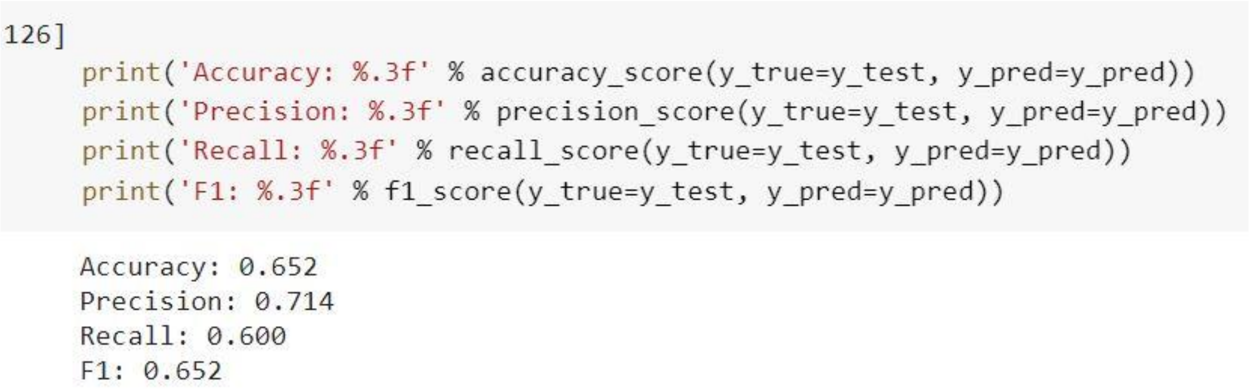
The Roc_auc score for the model is 89%.



```
Roc_score = 0.78
```

4. K-Nearest Neighbours: -

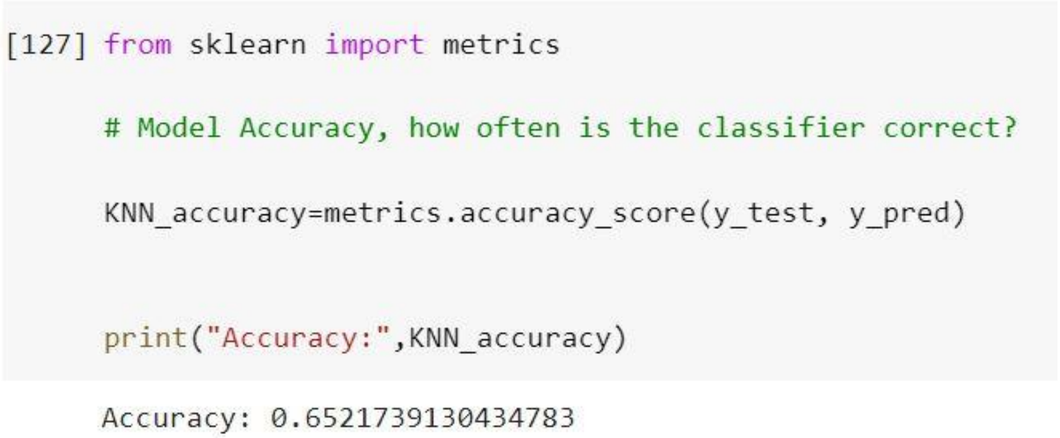
The fourth classification model we use is KNN. Using the metrics we get the accuracy, precision, Recall and F1-score .



```
126]
    print('Accuracy: %.3f' % accuracy_score(y_true=y_test, y_pred=y_pred))
    print('Precision: %.3f' % precision_score(y_true=y_test, y_pred=y_pred))
    print('Recall: %.3f' % recall_score(y_true=y_test, y_pred=y_pred))
    print('F1: %.3f' % f1_score(y_true=y_test, y_pred=y_pred))
```

```
    Accuracy: 0.652
    Precision: 0.714
    Recall: 0.600
    F1: 0.652
```

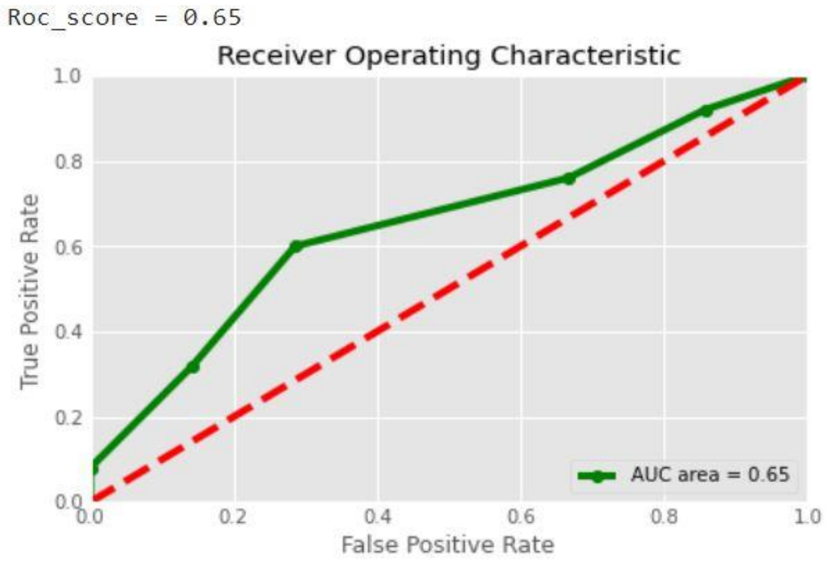
The accuracy score for the model is 65.2%.



```
[127] from sklearn import metrics

      # Model Accuracy, how often is the classifier correct?

      KNN_accuracy=metrics.accuracy_score(y_test, y_pred)


      print("Accuracy:",KNN_accuracy)
```

```
      Accuracy: 0.6521739130434783
```

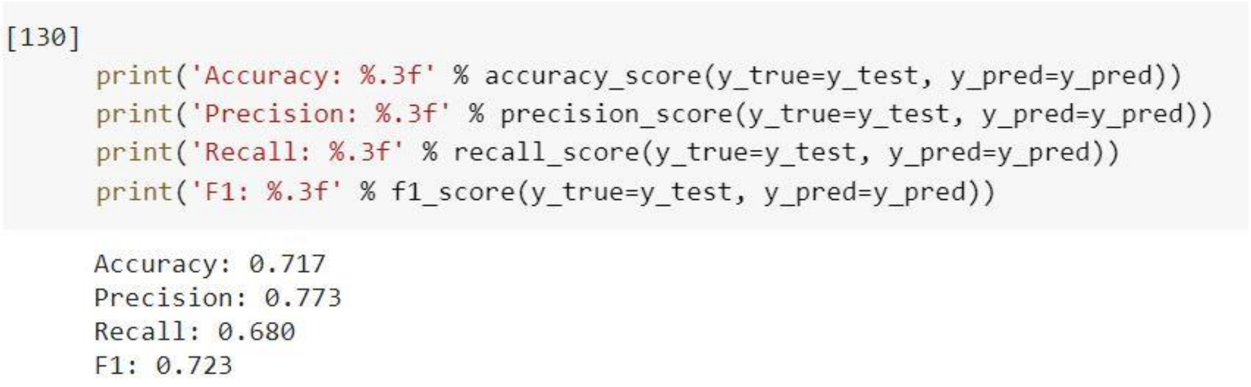
The Roc_auc score for the model is 65%.



```
Roc_score = 0.65
```

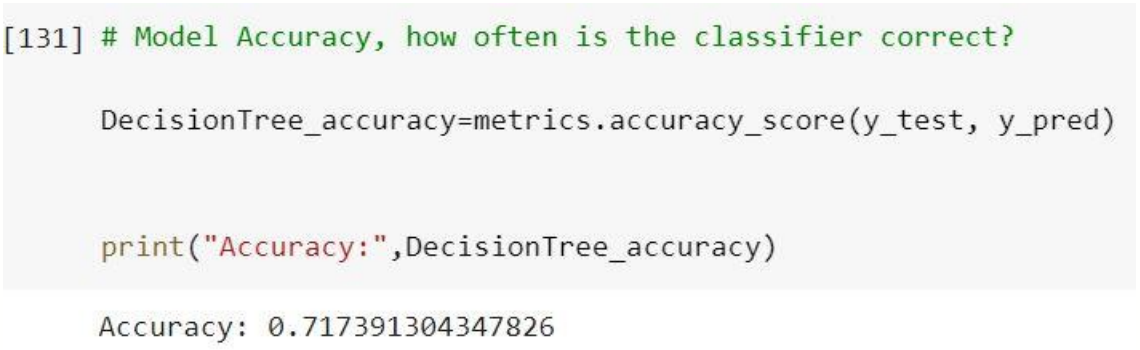
5. <u>DecisionTree Classifier: -</u>

The fifth classification model we are using is DecisionTree classifier. Using the metrics we get the accuracy, precision, Recall and F1-score .



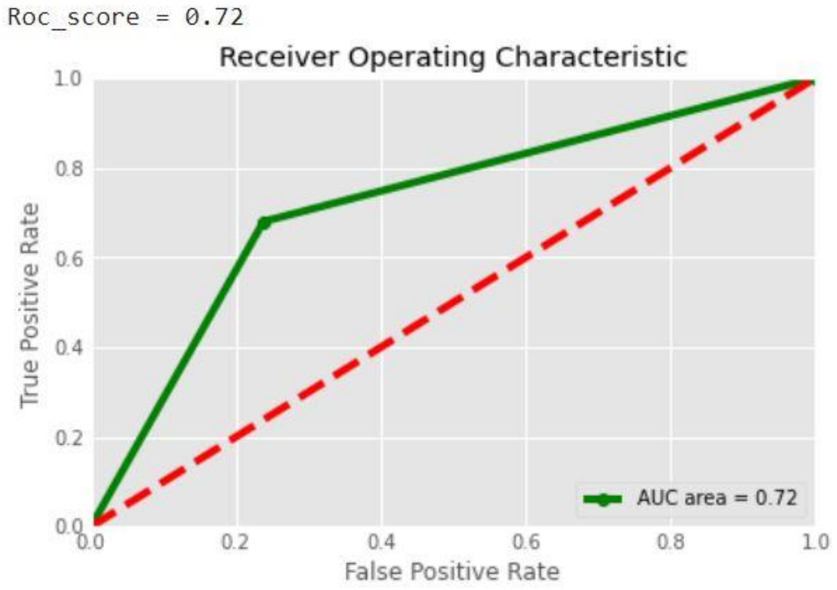
```
[130]
      print('Accuracy: %.3f' % accuracy_score(y_true=y_test, y_pred=y_pred))
      print('Precision: %.3f' % precision_score(y_true=y_test, y_pred=y_pred))
      print('Recall: %.3f' % recall_score(y_true=y_test, y_pred=y_pred))
      print('F1: %.3f' % f1_score(y_true=y_test, y_pred=y_pred))

      Accuracy: 0.717
      Precision: 0.773
      Recall: 0.680
      F1: 0.723
```

The accuracy score for the model is 71.7%.



```
[131] # Model Accuracy, how often is the classifier correct?

      DecisionTree_accuracy=metrics.accuracy_score(y_test, y_pred)


      print("Accuracy:",DecisionTree_accuracy)

      Accuracy: 0.717391304347826
```

The Roc_auc score for the model is 76%.



```
Roc_score = 0.72
```

‘

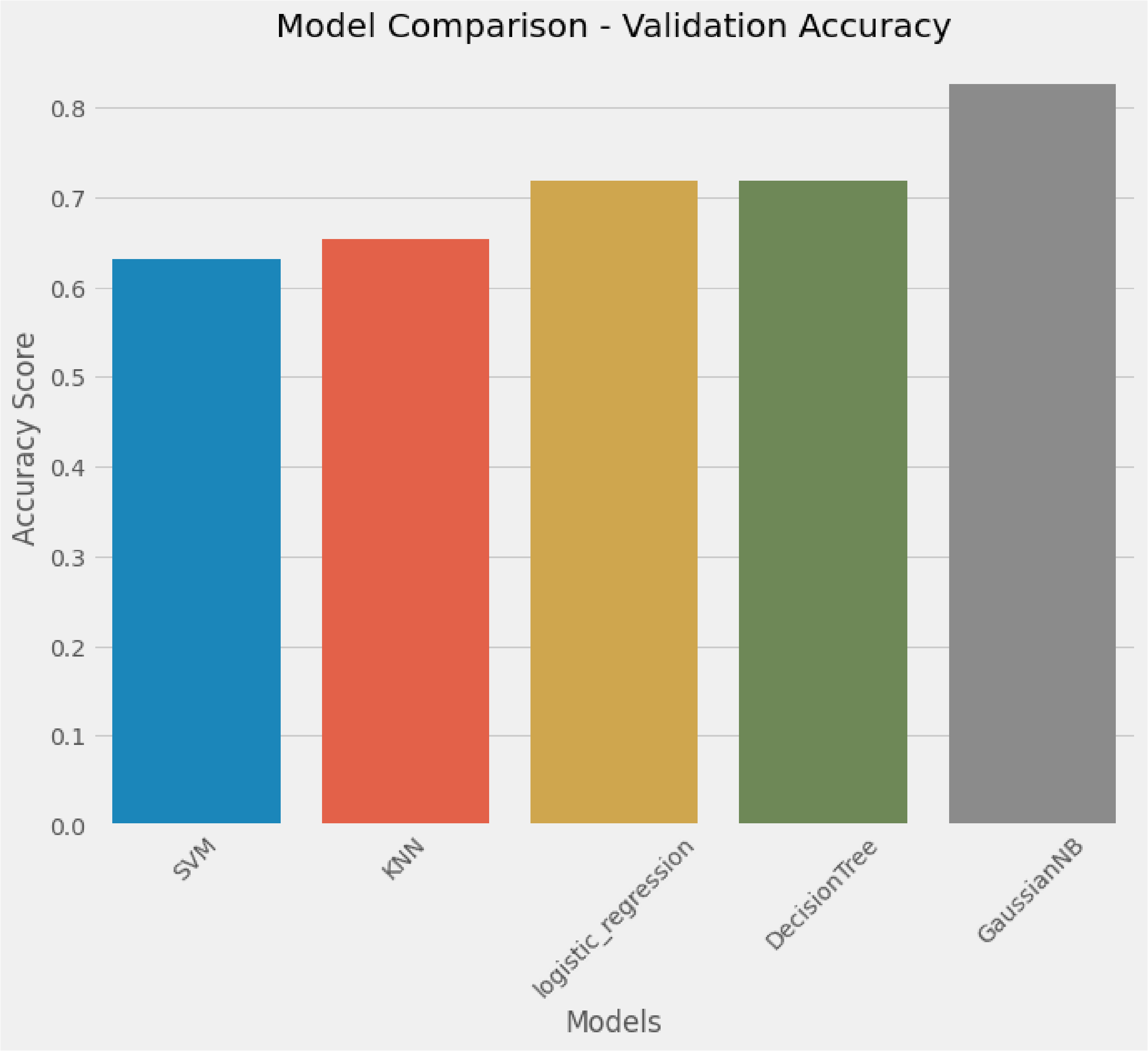
Comparing the models using accuracy score: -

Results: -

The barplot shows the Gaussian Naive bayes model performs the best amongst all the classificati on models that we used.

The accuracy of the models follows the order as listed: -

1.Gaussian naive bayes (Best model)
2.Decission Tree
3.Logistic Regression
4.K-Nearest-Neighbors
5.Support vector Machine



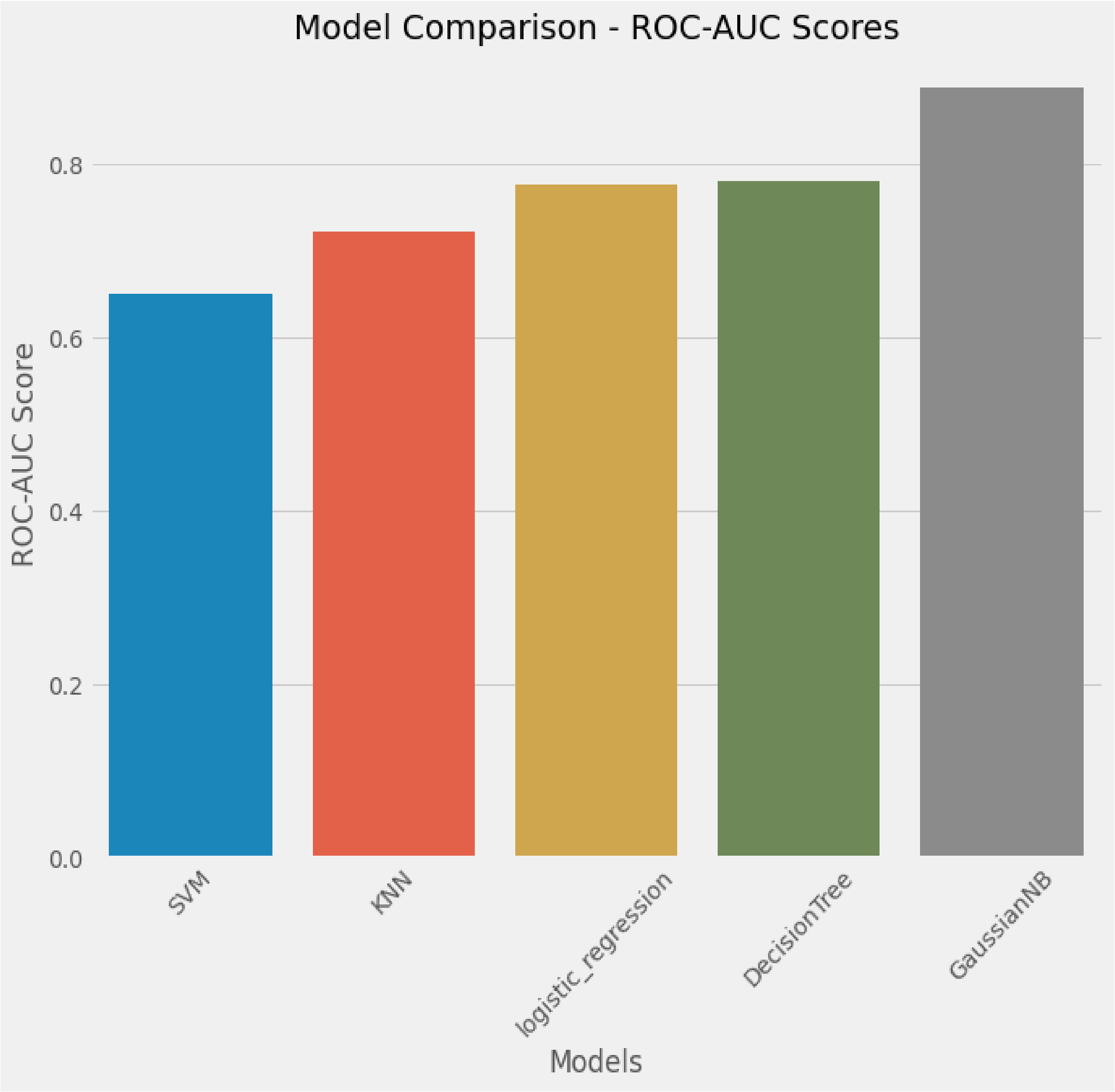
Comparing the models using Roc_auc score: -

Results: -

The barplot shows the Gaussian Naive bayes model performs the best amongst all the classification models that we used.

The Roc_auc of the models follows the order as listed: -

1.Gaussian naive bayes (Best model)
2.Decission Tree
3.Logistic Regression
4.K-Nearest-Neighbors
5.Support vector Machine


Model Comparison - ROC-AUC Scores
ROC-AUC Score
SVM
KNN
logistic_regression
DecisionTree
GaussianNB
Models