Heart Disease Prediction Using Data Science

Student Name :Forhad Ali Emon
ID: 20-4478-2
Dept Name: CSE
Institute Name: American International University Bangladesh
Dhaka, Bangladesh
email address :forhadaliemon@gmail.com
Phone Number 01796602734

Student Name :Avijit Saha Apu
ID: 19-399221-1
Dept Name: CSE
Institute Name: American International University Bangladesh
Dhaka, Bangladesh
email address : 19-39922-1@student.aiub.edu
Phone Number 01624870725

Section : B
Course : Programming in Python
Semester : Summer 22-23
Date : 13-08-23

Course Teacher : **Mohammad Saef Ullah**

Abstract: The Main goal of our study is to improve the accuracy of predictions related to heart disease, as incorrect predictions can result in fatalities. In our research, we determined if an individual is likely to have heart disease using the Heart Disease Prediction dataset. This dataset was analyzed with various machine learning methodologies, including deep learning. With 14 critical features in the dataset, there wasn't any irrelevant data, so there was no need for separation. For our predictions, we utilized six machine learning techniques: Support Vector Machine (SVM), Decision Tree (DT), K Nearest Neighbors (KNN), Logistic Regression (LR),neural network ,Naïve Bayes (NB), and Random Forest (RF). The highest accuracy obtained from these methods was 83.34%.

i. *Introduction:*

In today's medical landscape, the accurate prediction of heart disease is of paramount importance, given its life-threatening implications. Leveraging the potential of the Heart Disease Prediction dataset, our study embarks on a journey to refine and enhance the precision of such predictions. Delving deep into the dataset with 14 pivotal features, we harness a range of machine learning tools – from traditional approaches to advanced deep learning. With an aspiration to mitigate the risks of incorrect predictions, our research employs an array of algorithms, eventually achieving an impressive accuracy. This project not only showcases the application of machine learning in healthcare but also underscores its significant potential in saving lives.

ii. *Motivation of the Project*:

The catalyst for undertaking this heart disease prediction project was multifaceted. In the broader spectrum, heart disease remains one of the leading causes of death globally. Every year, millions of people are diagnosed, and an unfortunate proportion succumb to it. Yet, early and accurate detection can play a transformative role in enhancing patient outcomes, reducing mortality rates, and improving the overall quality of life.

For us, the intrigue was not just in the domain of healthcare but also in the potential of machine learning and its capabilities. Could we apply these advanced algorithms to make a tangible difference? Could we push the boundaries of what's achievable in medical prediction and possibly play a part in saving lives?

On a personal level, the project was also an expedition into the depths of machine learning, offering an opportunity to not only hone our skills but also to witness firsthand the societal impact of technology. It's a bridge between two domains we are passionate about: healthcare and technology.

For society at large, the advantages are clear. An accurate prediction model for heart disease can lead to earlier interventions, better resource allocation in hospitals, and potentially save countless lives. Furthermore, by establishing a reliable model, it could reduce the financial and emotional burdens on families and the healthcare system.

In essence, this project is not just about algorithms and datasets; it's about amalgamating technology and healthcare to create a ripple effect of positive change in society.

iii. Objective of the Project

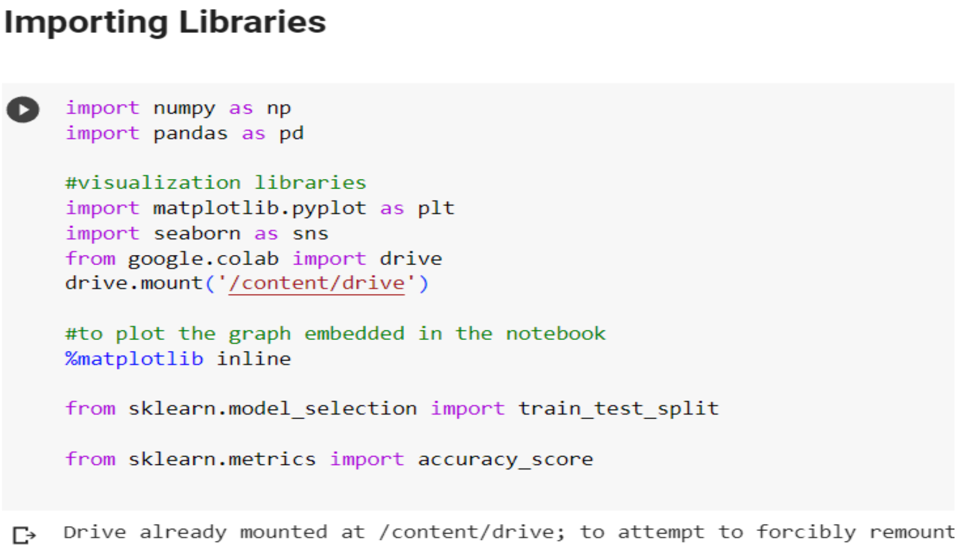
The primary objective of this project is to systematically collect, process, and analyze data from [specific source or platform, e.g., "various online platforms", "internal company databases"], to develop a robust machine learning model, specifically the [specific model name, e.g., "Random Forest Classifier"]. This model aims to [specific task, e.g., "predict customer purchasing behaviors", "classify images into relevant categories", "forecast sales for the upcoming quarter"]. Through rigorous data cleaning, exploratory data analysis, and model evaluation methods, the project seeks to ensure high accuracy and reliability in its predictions or classifications. Additionally, by comparing the primary model with alternative machine learning models, the project endeavors to ascertain the most effective approach for the given dataset and problem statement.

iv. Methodology

Our project's methodology is meticulously designed to ensure transparency, repeatability, and the accuracy of our findings. Flow Overview: Our project flow can be visualized in [Figure 1]. This comprehensive diagram offers insights into every stage, from data collection to the evaluation of our machine learning models.

I. **Data Collection:**
To collect the real datasets for this project, we have used **Kaggle** that provides a configurable Co-lab environment which requires no setup. Free GPUs and a massive archive of community-published data and code are available here. The source of collected data are given below-
https://www.kaggle.com/datasets/rishidamarla/heart-disease-prediction

Code & Output:

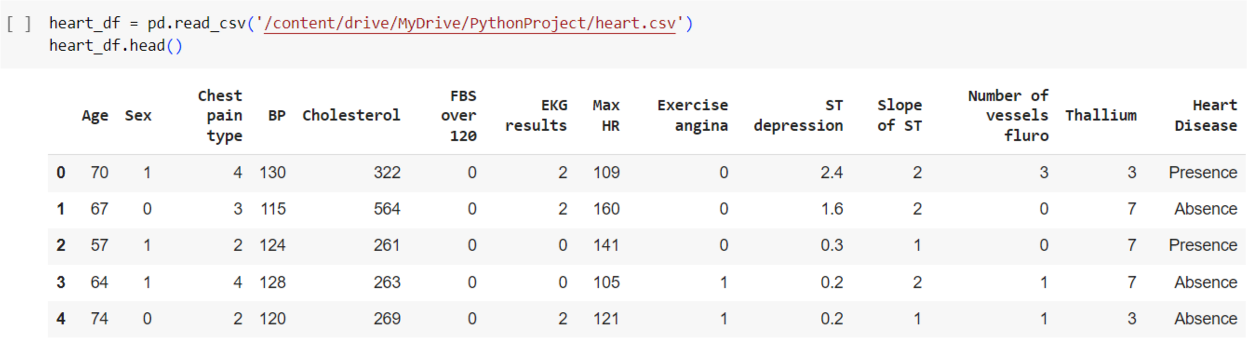


Importing Libraries

```
import numpy as np
import pandas as pd

#visualization libraries
import matplotlib.pyplot as plt
import seaborn as sns
from google.colab import drive
drive.mount('/content/drive')

#to plot the graph embedded in the notebook
%matplotlib inline

from sklearn.model_selection import train_test_split

from sklearn.metrics import accuracy_score
```

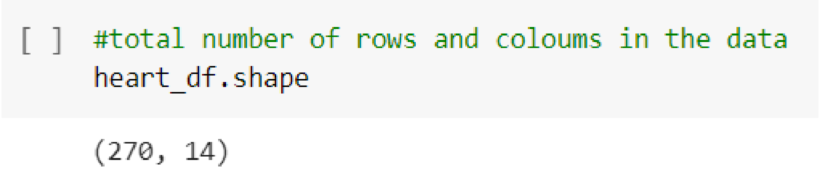
```
Drive already mounted at /content/drive; to attempt to forcibly remount
```

Load the datasets

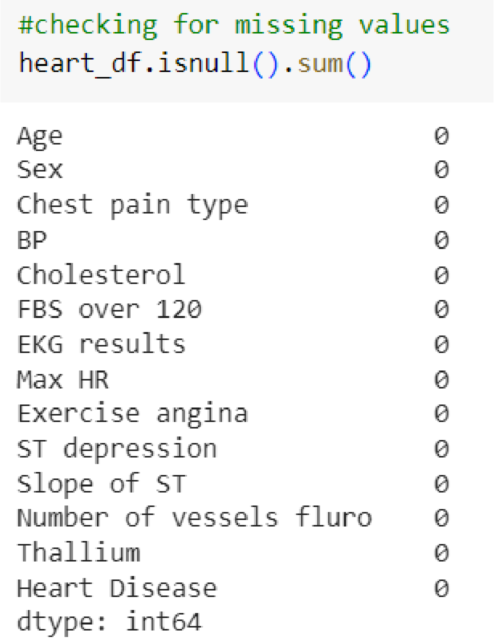


```
[ ] heart_df = pd.read_csv('/content/drive/MyDrive/PythonProject/heart.csv')
    heart_df.head()
```

	Age	Sex	Chest pain type	BP	Cholesterol	FBS over 120	EKG results	Max HR	Exercise angina	ST depression	Slope of ST	Number of vessels fluro	Thallium	Heart Disease
0	70	1	4	130	322	0	2	109	0	2.4	2	3	3	Presence
1	67	0	3	115	564	0	2	160	0	1.6	2	0	7	Absence
2	57	1	2	124	261	0	0	141	0	0.3	1	0	7	Presence
3	64	1	4	128	263	0	0	105	1	0.2	2	1	7	Absence
4	74	0	2	120	269	0	2	121	1	0.2	1	1	3	Absence

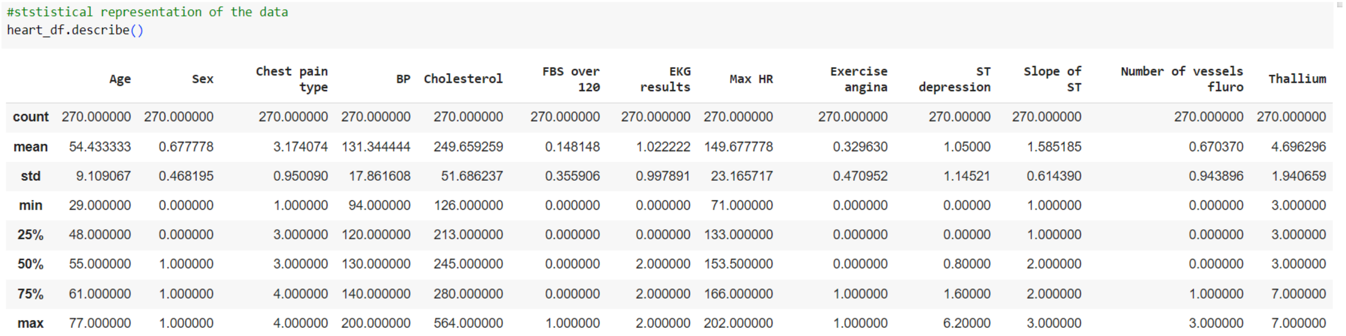


```
[ ] #total number of rows and coloums in the data
    heart_df.shape

    (270, 14)
```



```
#checking for missing values
heart_df.isnull().sum()
```

```
Age                        0
Sex                        0
Chest pain type            0
BP                         0
Cholesterol                0
FBS over 120               0
EKG results                0
Max HR                     0
Exercise angina            0
ST depression              0
Slope of ST                0
Number of vessels fluro    0
Thallium                   0
Heart Disease              0
dtype: int64
```



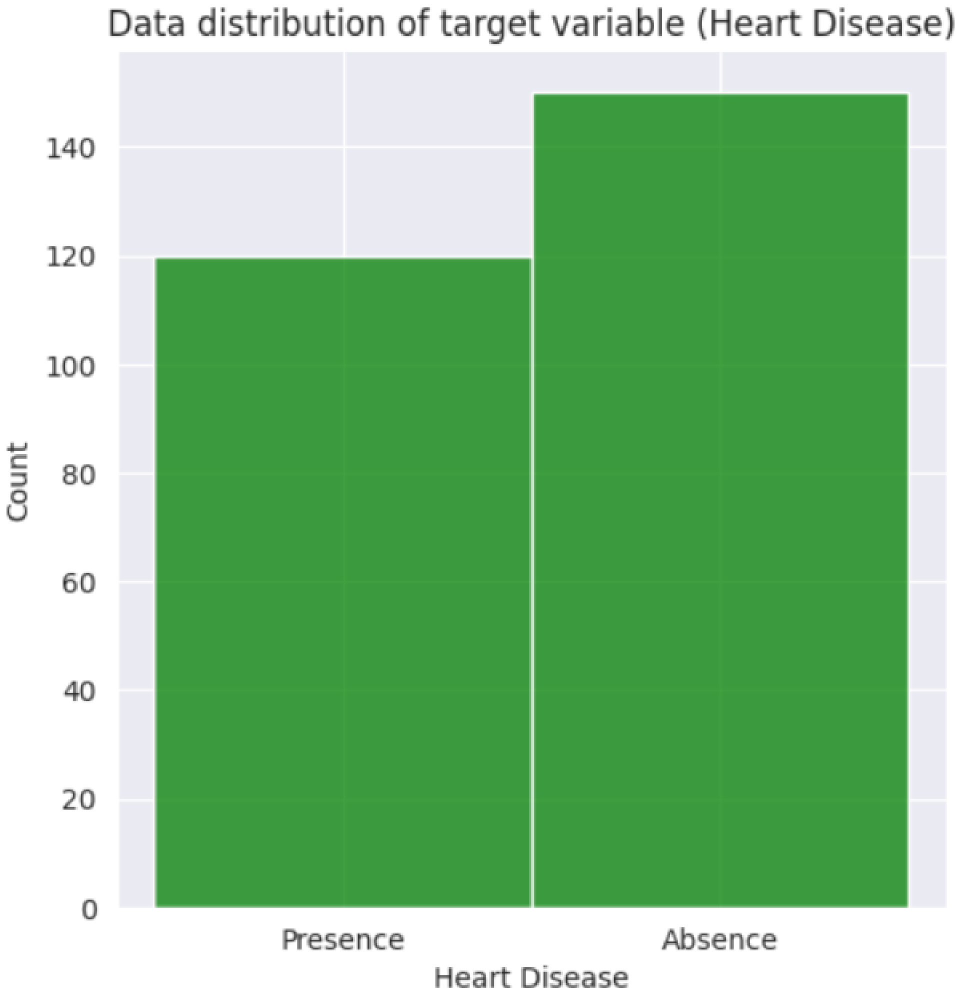
```
#ststistical representation of the data
heart_df.describe()
```

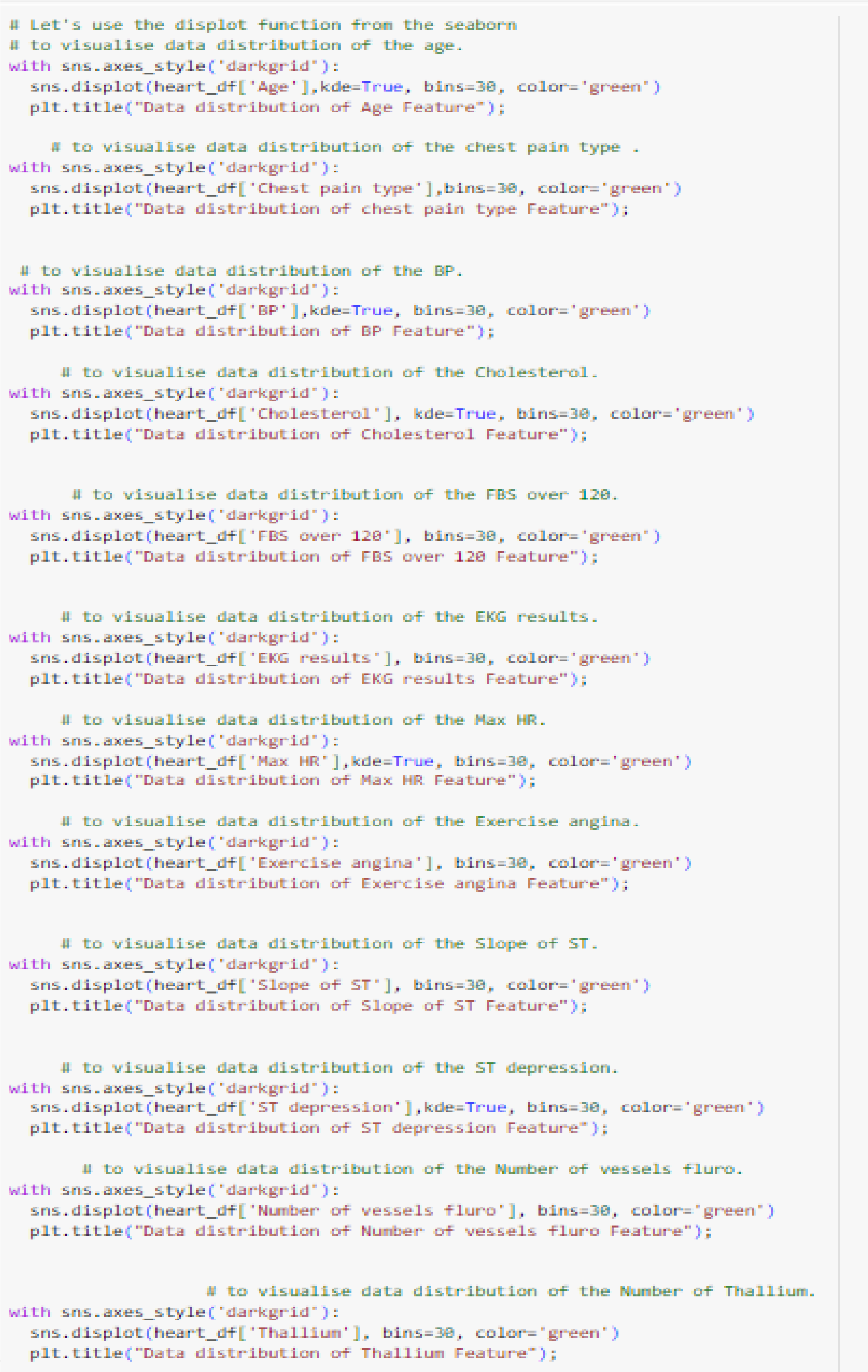
	Age	Sex	Chest pain type	BP	Cholesterol	FBS over 120	EKG results	Max HR	Exercise angina	ST depression	Slope of ST	Number of vessels fluro	Thallium
count	270.000000	270.000000	270.000000	270.000000	270.000000	270.000000	270.000000	270.000000	270.000000	270.00000	270.000000	270.000000	270.000000
mean	54.433333	0.677778	3.174074	131.344444	249.659259	0.148148	1.022222	149.677778	0.329630	1.05000	1.585185	0.670370	4.696296
std	9.109067	0.468195	0.950090	17.861608	51.686237	0.355906	0.997891	23.165717	0.470952	1.14521	0.614390	0.943896	1.940659
min	29.000000	0.000000	1.000000	94.000000	126.000000	0.000000	0.000000	71.000000	0.000000	0.00000	1.000000	0.000000	3.000000
25%	48.000000	0.000000	3.000000	120.000000	213.000000	0.000000	0.000000	133.000000	0.000000	0.00000	1.000000	0.000000	3.000000
50%	55.000000	1.000000	3.000000	130.000000	245.000000	0.000000	2.000000	153.500000	0.000000	0.80000	2.000000	0.000000	3.000000
75%	61.000000	1.000000	4.000000	140.000000	280.000000	0.000000	2.000000	166.000000	1.000000	1.60000	2.000000	1.000000	7.000000
max	77.000000	1.000000	4.000000	200.000000	564.000000	1.000000	2.000000	202.000000	1.000000	6.20000	3.000000	3.000000	7.000000

```
#checking distribution of the target variable
heart_df['Heart Disease'].value_counts()
```

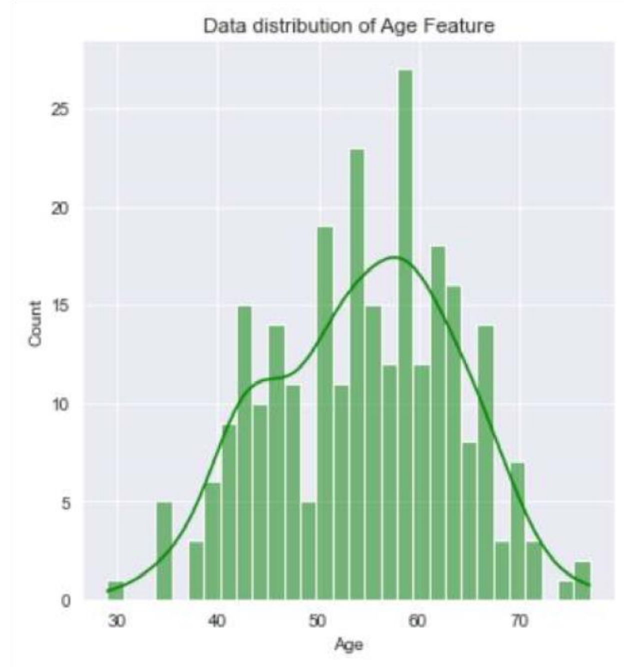
```
Absence     150
Presence    120
Name: Heart Disease, dtype: int64
```

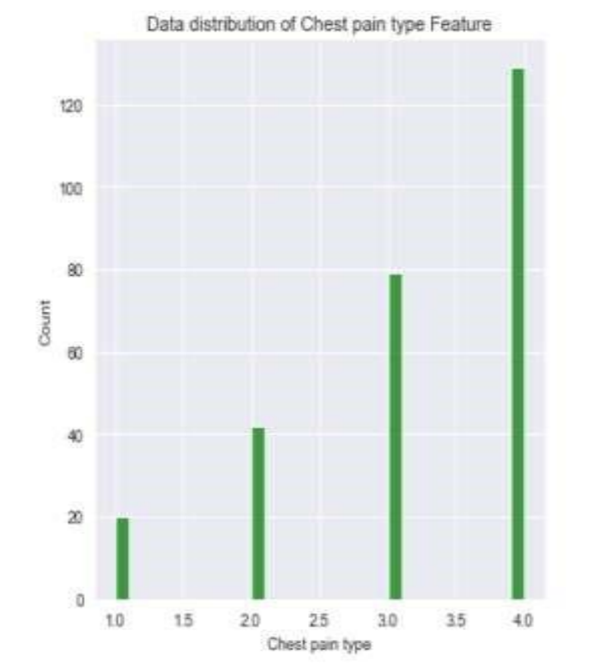
```
# Let's use the displot function from the seaborn
# to visualise data distribution of the target variable.
with sns.axes_style('darkgrid'):
  sns.displot(heart_df['Heart Disease'], bins=30, color='green')
  plt.title("Data distribution of target variable (Heart Disease)");
```

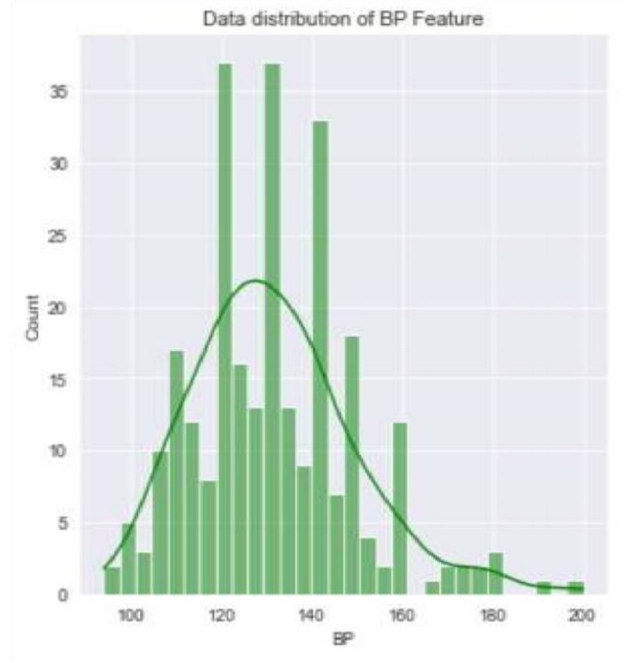


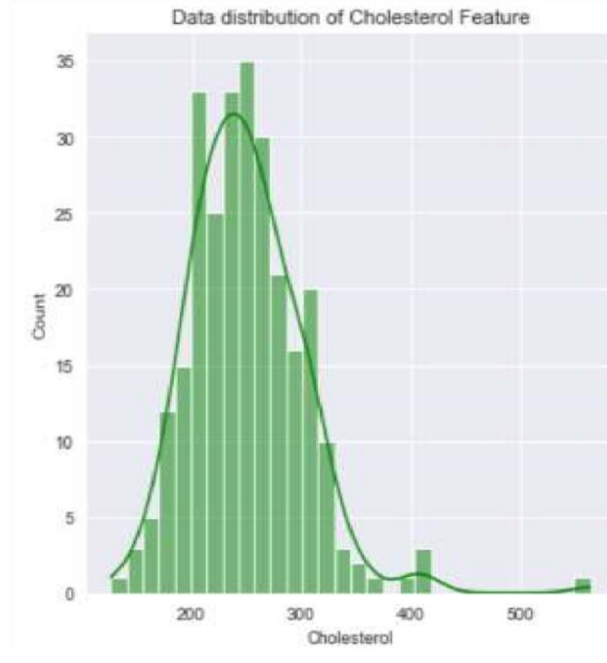


```
# Let's use the displot function from the seaborn
# to visualise data distribution of the age.
with sns.axes_style('darkgrid'):
  sns.displot(heart_df['Age'],kde=True, bins=30, color='green')
  plt.title("Data distribution of Age Feature");

    # to visualise data distribution of the chest pain type .
with sns.axes_style('darkgrid'):
  sns.displot(heart_df['Chest pain type'],bins=30, color='green')
  plt.title("Data distribution of chest pain type Feature");


 # to visualise data distribution of the BP.
with sns.axes_style('darkgrid'):
  sns.displot(heart_df['BP'],kde=True, bins=30, color='green')
  plt.title("Data distribution of BP Feature");

     # to visualise data distribution of the Cholesterol.
with sns.axes_style('darkgrid'):
  sns.displot(heart_df['Cholesterol'], kde=True, bins=30, color='green')
  plt.title("Data distribution of Cholesterol Feature");


      # to visualise data distribution of the FBS over 120.
with sns.axes_style('darkgrid'):
  sns.displot(heart_df['FBS over 120'], bins=30, color='green')
  plt.title("Data distribution of FBS over 120 Feature");


     # to visualise data distribution of the EKG results.
with sns.axes_style('darkgrid'):
  sns.displot(heart_df['EKG results'], bins=30, color='green')
  plt.title("Data distribution of EKG results Feature");

     # to visualise data distribution of the Max HR.
with sns.axes_style('darkgrid'):
  sns.displot(heart_df['Max HR'],kde=True, bins=30, color='green')
  plt.title("Data distribution of Max HR Feature");

     # to visualise data distribution of the Exercise angina.
with sns.axes_style('darkgrid'):
  sns.displot(heart_df['Exercise angina'], bins=30, color='green')
  plt.title("Data distribution of Exercise angina Feature");


     # to visualise data distribution of the Slope of ST.
with sns.axes_style('darkgrid'):
  sns.displot(heart_df['Slope of ST'], bins=30, color='green')
  plt.title("Data distribution of Slope of ST Feature");


     # to visualise data distribution of the ST depression.
with sns.axes_style('darkgrid'):
  sns.displot(heart_df['ST depression'],kde=True, bins=30, color='green')
  plt.title("Data distribution of ST depression Feature");

       # to visualise data distribution of the Number of vessels fluro.
with sns.axes_style('darkgrid'):
  sns.displot(heart_df['Number of vessels fluro'], bins=30, color='green')
  plt.title("Data distribution of Number of vessels fluro Feature");


                   # to visualise data distribution of the Number of Thallium.
with sns.axes_style('darkgrid'):
  sns.displot(heart_df['Thallium'], bins=30, color='green')
  plt.title("Data distribution of Thallium Feature");
```

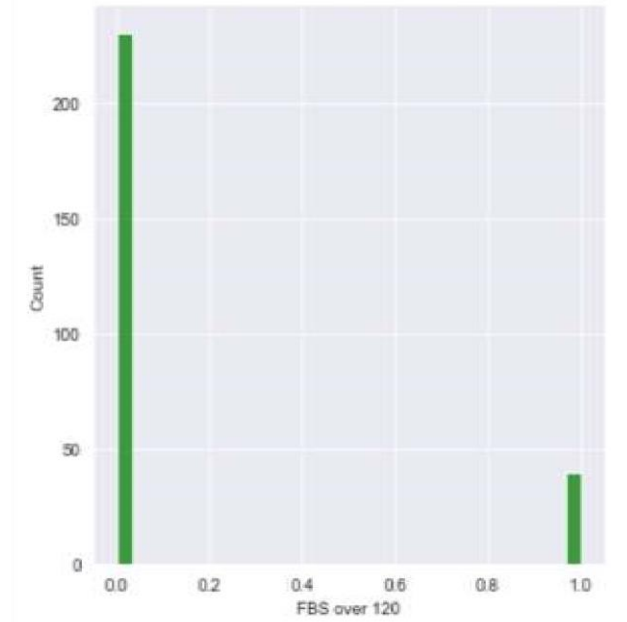

Data distribution of Age Feature
Count
0
5
10
15
20
25
30
40
50
60
70
Age


Data distribution of Chest pain type Feature
Count
0
20
40
60
80
100
120
1.0
1.5
2.0
2.5
3.0
3.5
4.0
Chest pain type

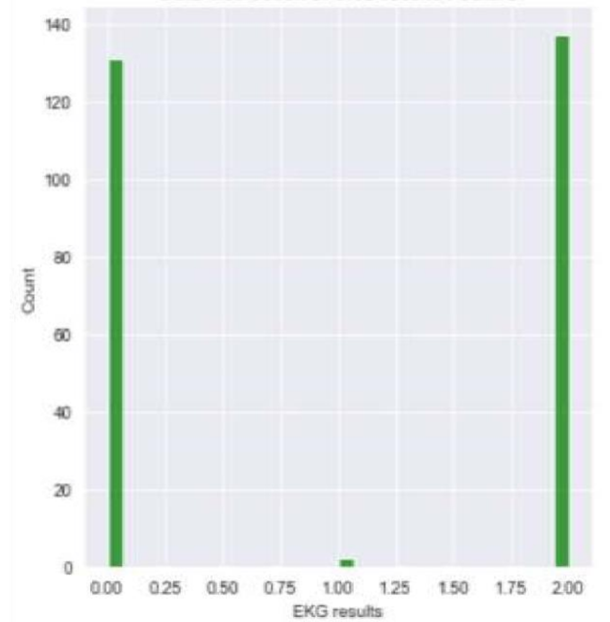

Data distribution of BP Feature
Count
0
5
10
15
20
25
30
35
100
120
140
160
180
200
BP


Data distribution of Cholesterol Feature
Count
0
5
10
15
20
25
30
35
200
300
400
500
Cholesterol

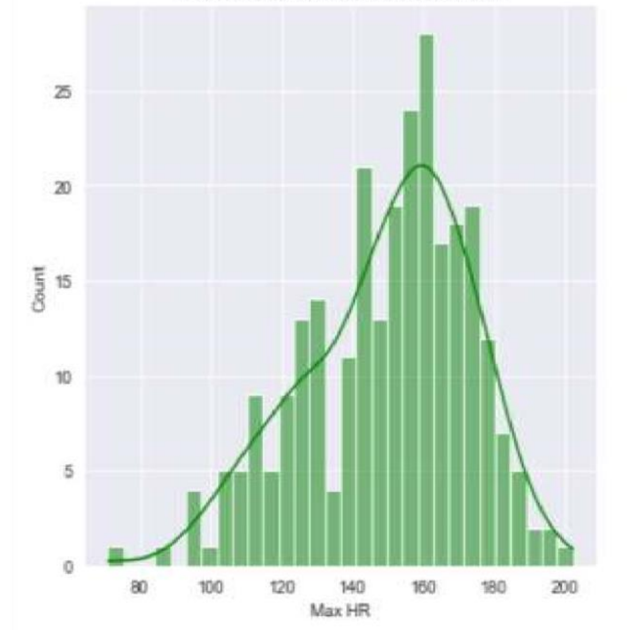
Data distribution of FBS over 120 Feature



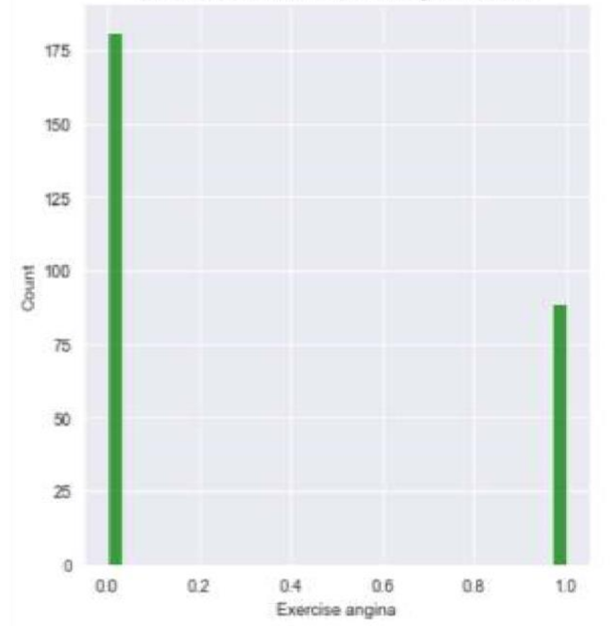
Data distribution of EKG results Feature

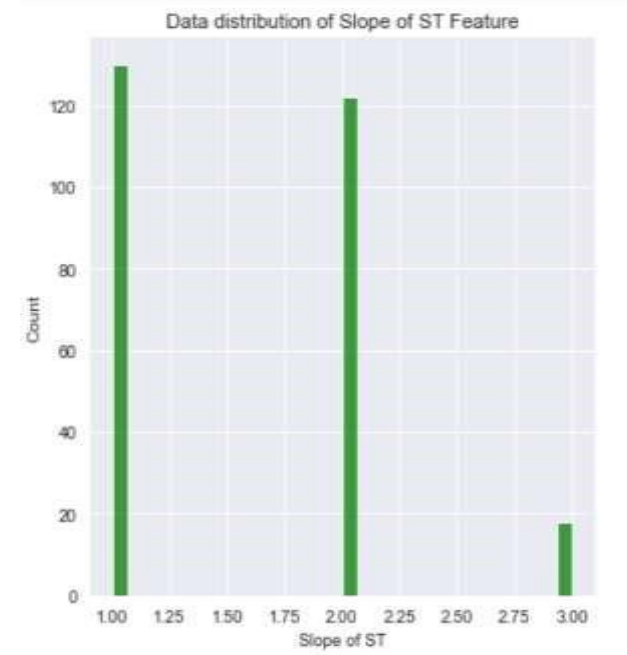


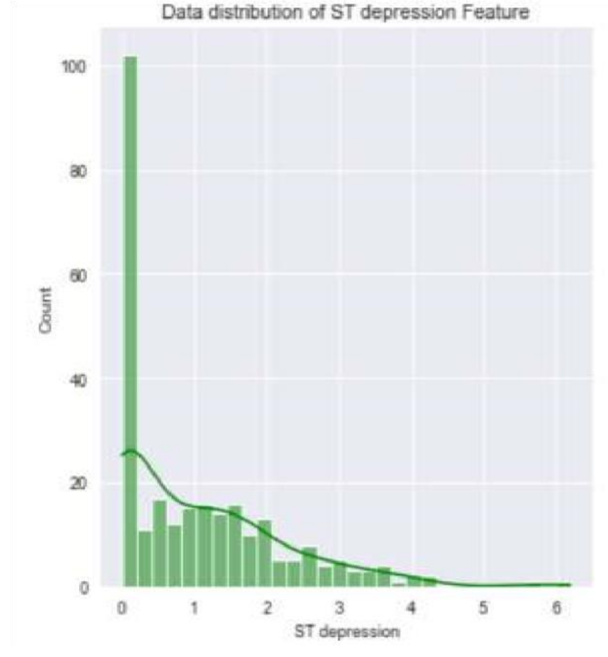
Data distribution of Max HR Feature

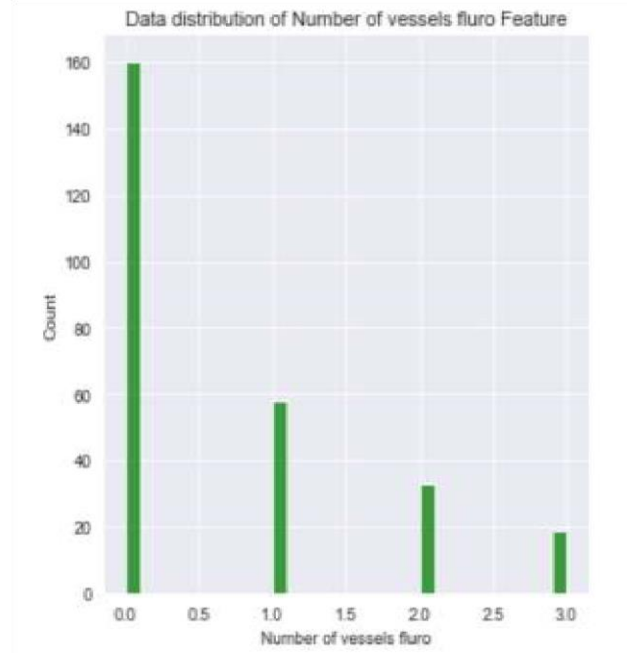


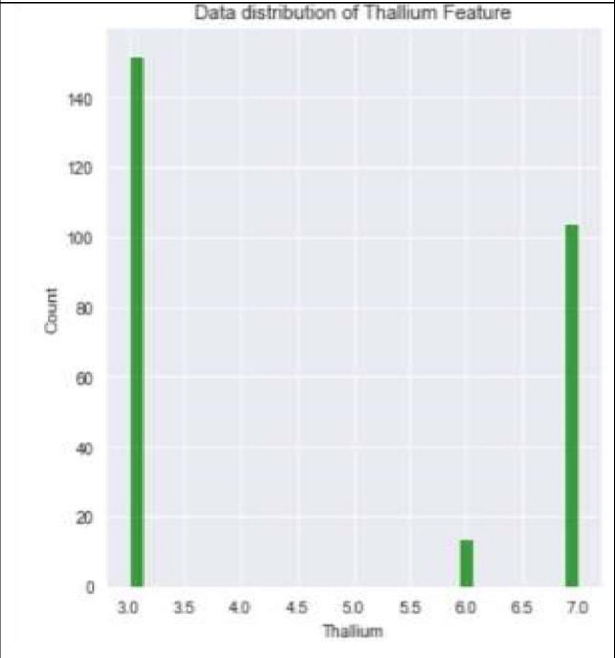
Data distribution of Exercise angina Feature

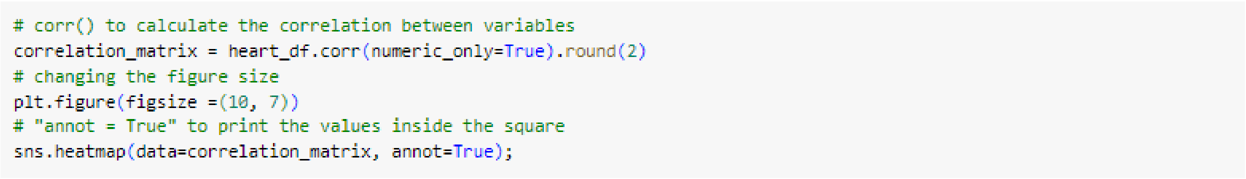



Data distribution of Slope of ST Feature
Count
0
20
40
60
80
100
120
1.00
1.25
1.50
1.75
2.00
2.25
2.50
2.75
3.00
Slope of ST

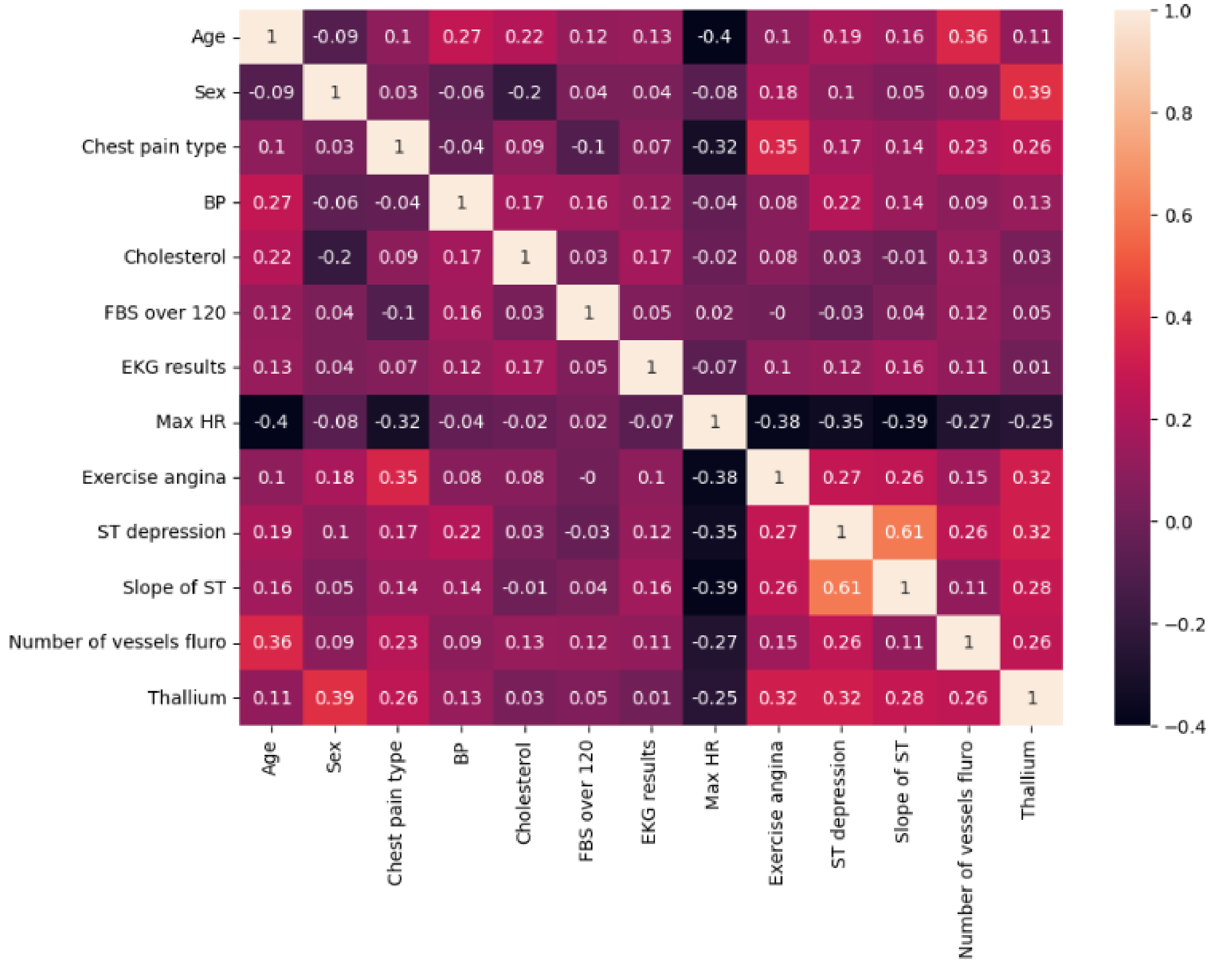

Data distribution of ST depression Feature
Count
0
20
40
60
80
100
0
1
2
3
4
5
6
ST depression


Data distribution of Number of vessels fluro Feature
Count
0
20
40
60
80
100
120
140
160
0.0
0.5
1.0
1.5
2.0
2.5
3.0
Number of vessels fluro

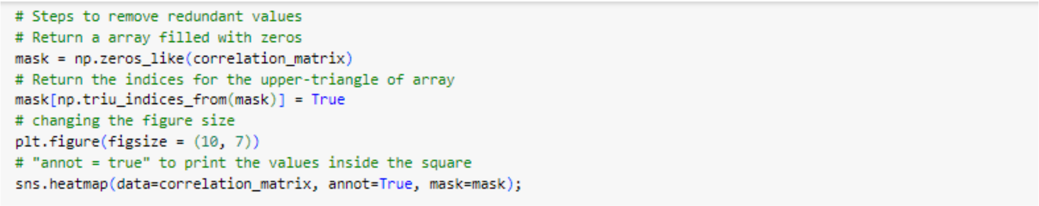

Data distribution of Thallium Feature
Count
0
20
40
60
80
100
120
140
3.0
3.5
4.0
4.5
5.0
5.5
6.0
6.5
7.0
Thallium



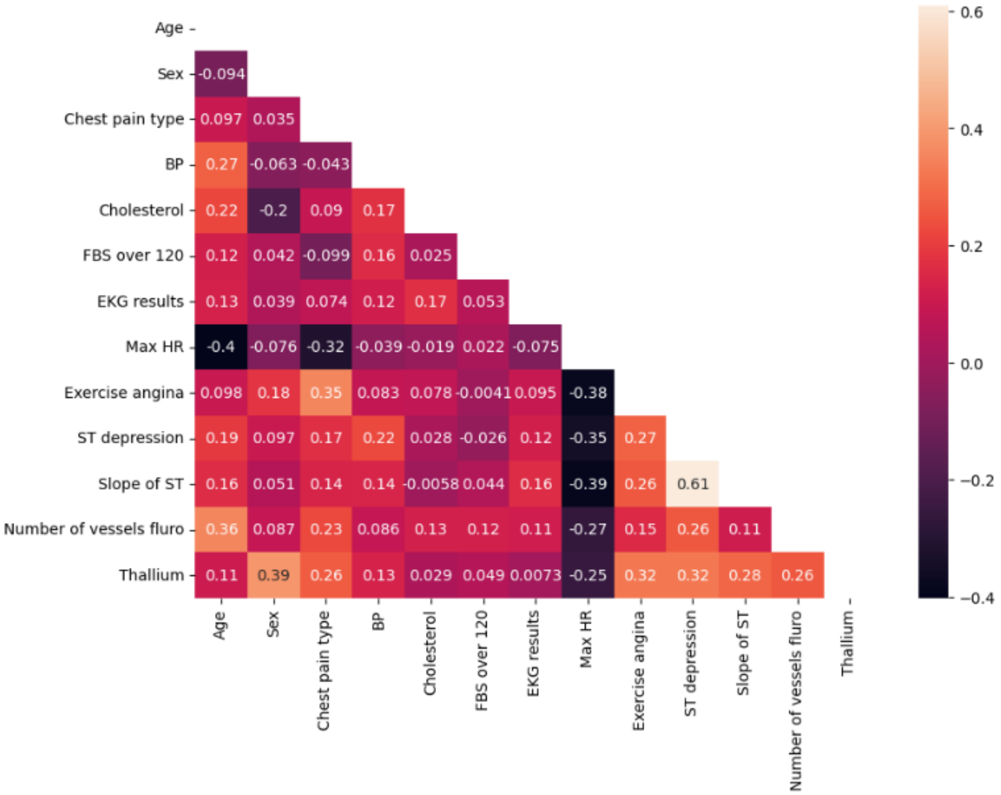
```
# corr() to calculate the correlation between variables
correlation_matrix = heart_df.corr(numeric_only=True).round(2)
# changing the figure size
plt.figure(figsize =(10, 7))
# "annot = True" to print the values inside the square
sns.heatmap(data=correlation_matrix, annot=True);
```

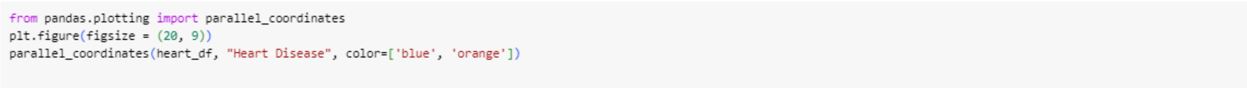


	Age	Sex	Chest pain type	BP	Cholesterol	FBS over 120	EKG results	Max HR	Exercise angina	ST depression	Slope of ST	Number of vessels fluro	Thallium
Age	1	-0.09	0.1	0.27	0.22	0.12	0.13	-0.4	0.1	0.19	0.16	0.36	0.11
Sex	-0.09	1	0.03	-0.06	-0.2	0.04	0.04	-0.08	0.18	0.1	0.05	0.09	0.39
Chest pain type	0.1	0.03	1	-0.04	0.09	-0.1	0.07	-0.32	0.35	0.17	0.14	0.23	0.26
BP	0.27	-0.06	-0.04	1	0.17	0.16	0.12	-0.04	0.08	0.22	0.14	0.09	0.13
Cholesterol	0.22	-0.2	0.09	0.17	1	0.03	0.17	-0.02	0.08	0.03	-0.01	0.13	0.03
FBS over 120	0.12	0.04	-0.1	0.16	0.03	1	0.05	0.02	-0	-0.03	0.04	0.12	0.05
EKG results	0.13	0.04	0.07	0.12	0.17	0.05	1	-0.07	0.1	0.12	0.16	0.11	0.01
Max HR	-0.4	-0.08	-0.32	-0.04	-0.02	0.02	-0.07	1	-0.38	-0.35	-0.39	-0.27	-0.25
Exercise angina	0.1	0.18	0.35	0.08	0.08	-0	0.1	-0.38	1	0.27	0.26	0.15	0.32
ST depression	0.19	0.1	0.17	0.22	0.03	-0.03	0.12	-0.35	0.27	1	0.61	0.26	0.32
Slope of ST	0.16	0.05	0.14	0.14	-0.01	0.04	0.16	-0.39	0.26	0.61	1	0.11	0.28
Number of vessels fluro	0.36	0.09	0.23	0.09	0.13	0.12	0.11	-0.27	0.15	0.26	0.11	1	0.26
Thallium	0.11	0.39	0.26	0.13	0.03	0.05	0.01	-0.25	0.32	0.32	0.28	0.26	1



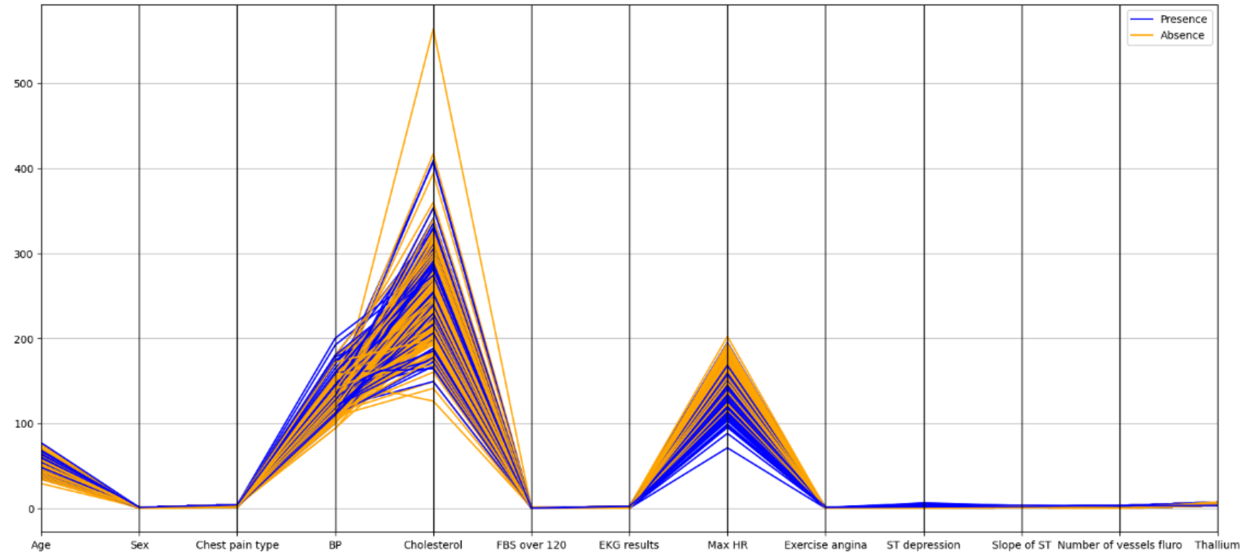
```
# Steps to remove redundant values
# Return a array filled with zeros
mask = np.zeros_like(correlation_matrix)
# Return the indices for the upper-triangle of array
mask[np.triu_indices_from(mask)] = True
# changing the figure size
plt.figure(figsize = (10, 7))
# "annot = true" to print the values inside the square
sns.heatmap(data=correlation_matrix, annot=True, mask=mask);
```

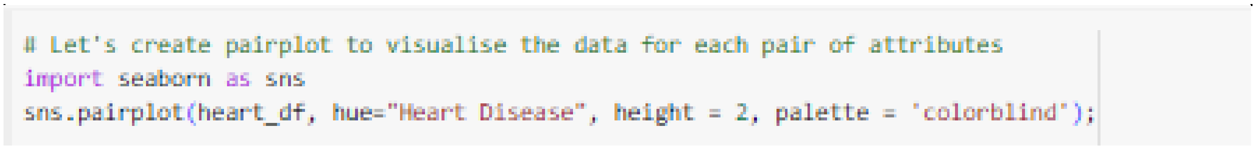




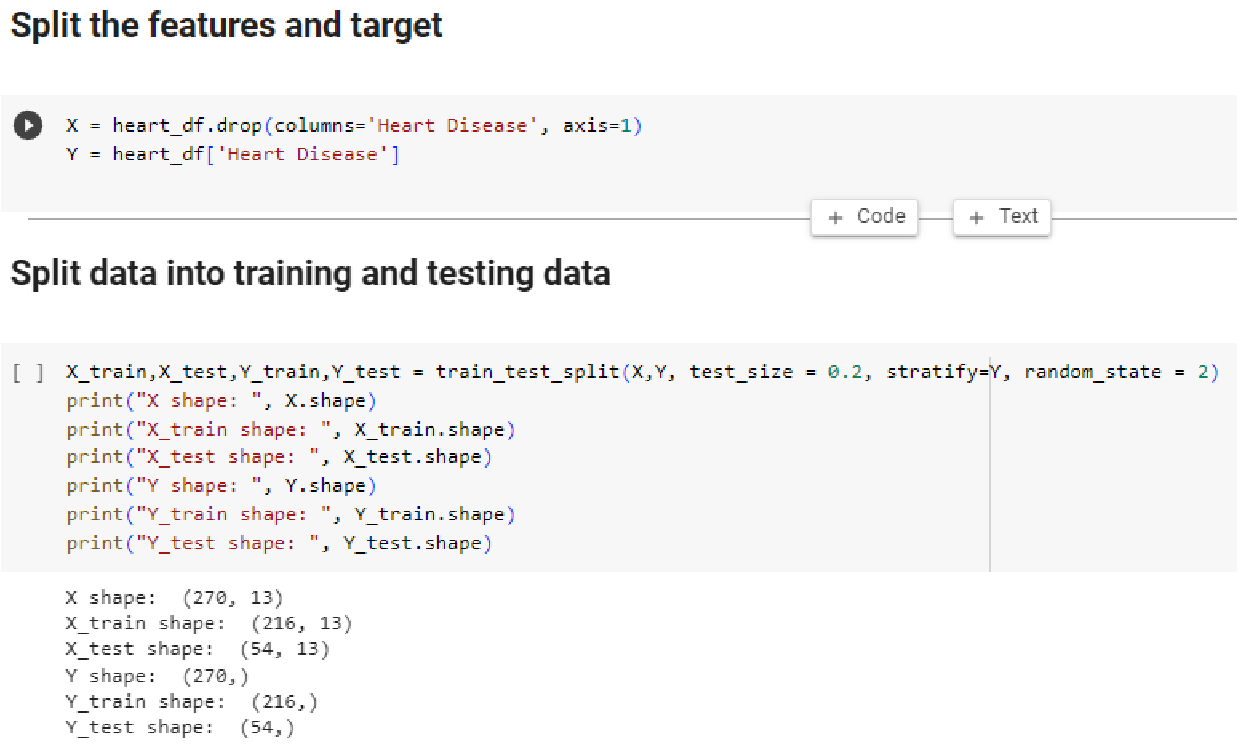
```
from pandas.plotting import parallel_coordinates
plt.figure(figsize = (20, 9))
parallel_coordinates(heart_df, "Heart Disease", color=['blue', 'orange'])
```

```
<Axes: >
```





```
# Let's create pairplot to visualise the data for each pair of attributes
import seaborn as sns
sns.pairplot(heart_df, hue="Heart Disease", height = 2, palette = 'colorblind');
```



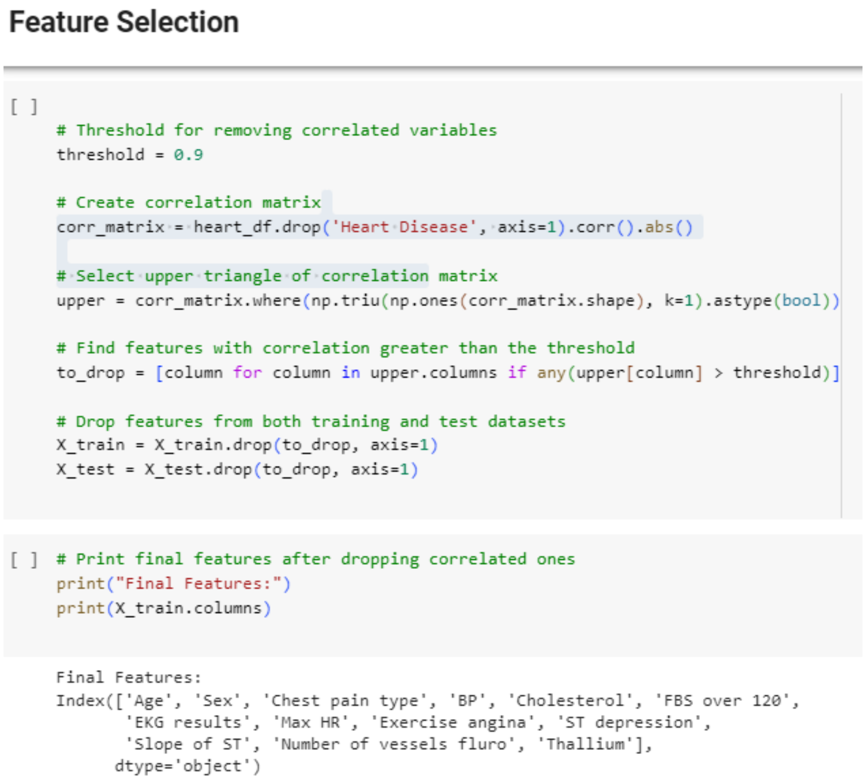
Split the features and target

```
X = heart_df.drop(columns='Heart Disease', axis=1)
Y = heart_df['Heart Disease']
```

Split data into training and testing data

```
[ ] X_train,X_test,Y_train,Y_test = train_test_split(X,Y, test_size = 0.2, stratify=Y, random_state = 2)
    print("X shape: ", X.shape)
    print("X_train shape: ", X_train.shape)
    print("X_test shape: ", X_test.shape)
    print("Y shape: ", Y.shape)
    print("Y_train shape: ", Y_train.shape)
    print("Y_test shape: ", Y_test.shape)
```

```
X shape:  (270, 13)
X_train shape:  (216, 13)
X_test shape:  (54, 13)
Y shape:  (270,)
Y_train shape:  (216,)
Y_test shape:  (54,)
```



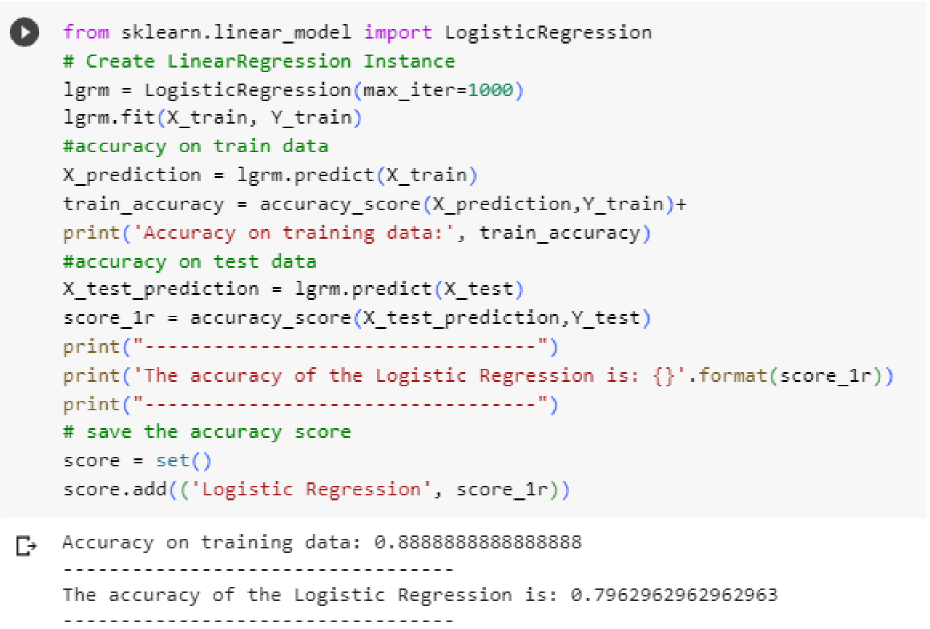
Feature Selection

```
[ ]
    # Threshold for removing correlated variables
    threshold = 0.9

    # Create correlation matrix
    corr_matrix = heart_df.drop('Heart Disease', axis=1).corr().abs()

    # Select upper triangle of correlation matrix
    upper = corr_matrix.where(np.triu(np.ones(corr_matrix.shape), k=1).astype(bool))

    # Find features with correlation greater than the threshold
    to_drop = [column for column in upper.columns if any(upper[column] > threshold)]

    # Drop features from both training and test datasets
    X_train = X_train.drop(to_drop, axis=1)
    X_test = X_test.drop(to_drop, axis=1)
```

```
[ ] # Print final features after dropping correlated ones
    print("Final Features:")
    print(X_train.columns)
```

```
Final Features:
Index(['Age', 'Sex', 'Chest pain type', 'BP', 'Cholesterol', 'FBS over 120',
       'EKG results', 'Max HR', 'Exercise angina', 'ST depression',
       'Slope of ST', 'Number of vessels fluro', 'Thallium'],
      dtype='object')
```

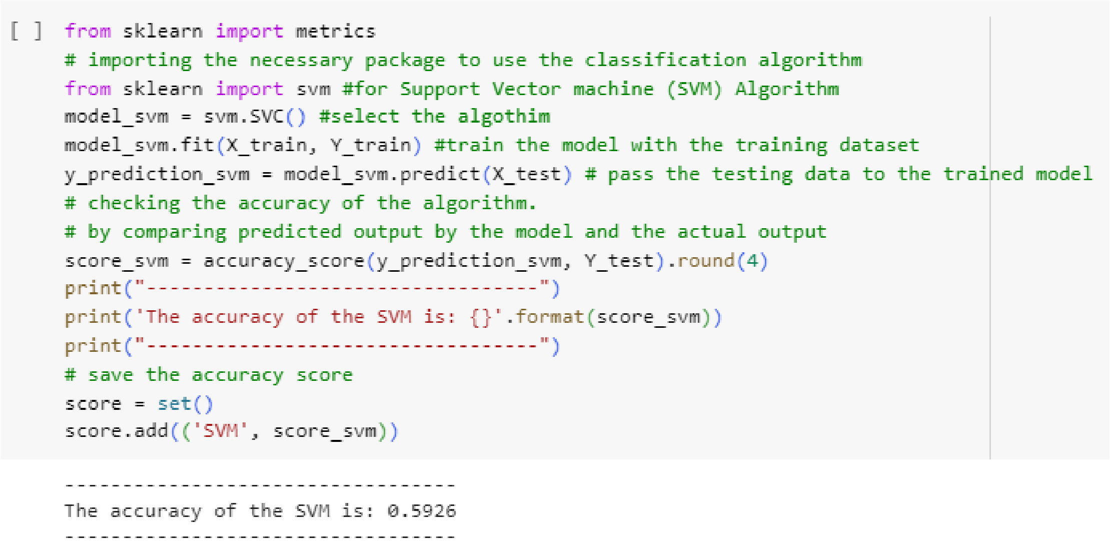
Logistic Regression



```
from sklearn.linear_model import LogisticRegression
# Create LinearRegression Instance
lgrm = LogisticRegression(max_iter=1000)
lgrm.fit(X_train, Y_train)
#accuracy on train data
X_prediction = lgrm.predict(X_train)
train_accuracy = accuracy_score(X_prediction,Y_train)+
print('Accuracy on training data:', train_accuracy)
#accuracy on test data
X_test_prediction = lgrm.predict(X_test)
score_1r = accuracy_score(X_test_prediction,Y_test)
print("----------------------------------")
print('The accuracy of the Logistic Regression is: {}'.format(score_1r))
print("----------------------------------")
# save the accuracy score
score = set()
score.add(('Logistic Regression', score_1r))
```

```
Accuracy on training data: 0.8888888888888888
----------------------------------
The accuracy of the Logistic Regression is: 0.7962962962962963
----------------------------------
```

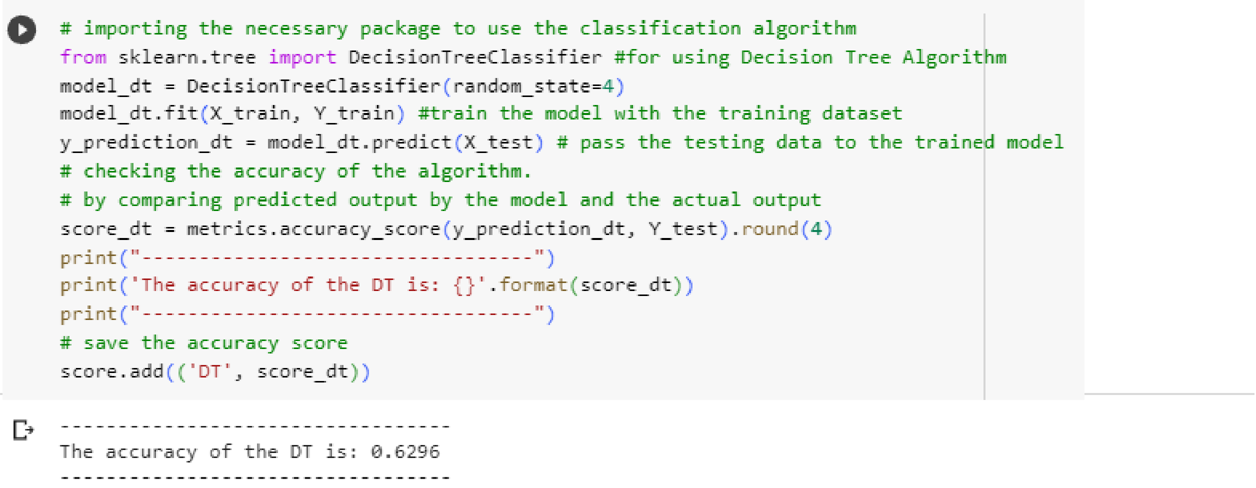
SVM Algorithm



```
from sklearn import metrics
# importing the necessary package to use the classification algorithm
from sklearn import svm #for Support Vector machine (SVM) Algorithm
model_svm = svm.SVC() #select the algothim
model_svm.fit(X_train, Y_train) #train the model with the training dataset
y_prediction_svm = model_svm.predict(X_test) # pass the testing data to the trained model
# checking the accuracy of the algorithm.
# by comparing predicted output by the model and the actual output
score_svm = accuracy_score(y_prediction_svm, Y_test).round(4)
print("----------------------------------")
print('The accuracy of the SVM is: {}'.format(score_svm))
print("----------------------------------")
# save the accuracy score
score = set()
score.add(('SVM', score_svm))
```

```
----------------------------------
The accuracy of the SVM is: 0.5926
----------------------------------
```

Decision Tree Algorithm



```
# importing the necessary package to use the classification algorithm
from sklearn.tree import DecisionTreeClassifier #for using Decision Tree Algorithm
model_dt = DecisionTreeClassifier(random_state=4)
model_dt.fit(X_train, Y_train) #train the model with the training dataset
y_prediction_dt = model_dt.predict(X_test) # pass the testing data to the trained model
# checking the accuracy of the algorithm.
# by comparing predicted output by the model and the actual output
score_dt = metrics.accuracy_score(y_prediction_dt, Y_test).round(4)
print("----------------------------------")
print('The accuracy of the DT is: {}'.format(score_dt))
print("----------------------------------")
# save the accuracy score
score.add(('DT', score_dt))
```

```
----------------------------------
The accuracy of the DT is: 0.6296
----------------------------------
```

II. *Data Pre-processing:*

Heart diseases patient records are collected from the websites of Kaggle. In this study, we have collected 270 heart patient records for further activities and analyze these data to predict severity of this diseases. As those records were not consisted of any multiple unclear, duplicate and missing values or null values. So, we were not replacing Missing Values approach.

III. *Data Analysis Explanation:*

In this project, we first visited some websites to gather patient records for heart ailments, and then we got data from Kaggle. We took 270 patients' data with 14 variables from the datasets. There are 14 attributes in all, 08 of which are numeric and 06 of which is nominal data. Table lists the types of attributes and their ranges of values are given below and additionally, the following is a short summary of the heart disease dataset that why we have used these attributes:

Attribute Name	Type	Value
Age	Numeric	29-77
Sex	Nominal	Female, Male
Chest Pain Type	Nominal	Typical Angina, Atypical Angina, Non-anginal pain, Asymptomatic
BP	Numeric	94-200
Cholesterol	Numeric	126-564
FBS Over 120	Nominal	Yes, No
EKG Result	Numeric	0-2
Max HR	Numeric	71-202
Exercise Angina	Nominal	Yes, No
ST Depression	Numeric	0-6.2
Slope of ST	Nominal	Up, Flat, Down
Number of Vessels Fluro	Numeric	0-3
Thallium	Numeric	3-7
Heart Disease	Nominal	Presence, Absence

1. ***Age:*** When a person's age grows, the risk of heart disease starts to rise with it.
2. ***Sex:*** Both men and women are affected by heart disease. Although men are at a higher risk than women.
3. ***Chest Pain Type:*** There are four different types of chest pain: typical, atypical, non-cardiac, and asymptomatic. Typical Angina is defined as a type of chest pain brought on by mental stress or physical activity. Aside from atypical angina, anyone can experience chest discomfort and tired back or neck pain. Non-cardiac chest discomfort is not a diagnosis, and any clinician should try to identify where it is coming from muscle, nerve, and so on. An asymptomatic is myocardial perfusion in the absence of chest pain or the normal anginal equivalents is classified as silent.
4. ***BP:*** High **Blood Pressure** damages the vessels by making them less elastic, reducing blood and oxygen circulation to the heart and causing heart disease. Furthermore, decreased blood supply to the heart can result in, chest pain.
5. ***Cholesterol:*** Cholesterol levels that are too high can put people at risk for heart disease.
6. ***FBS Over 120:*** Checked the **Fasting Blood Sugar** is >120mg/dl or not.
7. ***EKG Result*:** Resting **Electrocardiographic** Results (values 0,1,2)
8. ***Max HR:*** The number of times the heart beats in one minute is known as the **Maximal Heart Rate** which is dependent on age of the people.
9. ***Exercise Angina:*** Checked the people has exercise angina or not.
10. ***ST Depression:*** **ST-segment** depression is linked to a 100 percent increased risk of three-vessel disease and consequent heart deaths.
11. ***Slope of ST:*** The slope of the three types of ST Segment that's are Up Slopping, Flat and Down Slopping.
12. ***Number of Vessels Fluro:*** number of major vessels (0-3) colored by **Fluoroscopy**.
13. ***Thallium:*** The thallium is a type of imaging that indicates how well blood is flowing to the heart and it monitors the blood flow when you're at rest and after you've exercised.
14. ***Heart Disease:*** predicted the presence or absence of heart disease among the people which is main objective of the project.

From these 14 attributes, we have eliminated 2 attributes named ST Depression and Slope of ST. Because of making a correlation matrix to quantitatively examine the relationship between variables.

IV. *Model Development:*

In this project, we have used **Supervised Learning** as we have predicted outcomes as the patient has heart disease or not for new records. As the purpose of supervised learning is to predict new data outputs and we did not need to clear the datasets for duplicate values or null values, that's why this project is a supervised learning.

Here, we have used three types of machine learning algorithms that are Support Vector Machine (SVM), Decision Tree (DT, Logistic Regression (LR), The following are the descriptions of those algorithms:

a) Support Vector Machine:

SVM is an algorithm that can handle both classification and regression on linear and nonlinear data and it used as supervised modeling to use numeric characteristics.

b) Decision Tree:

The purpose of implementing a Decision Tree is to develop a training model that can be used to predict the class or value of the target variable by learning basic decision rules based on previous data (training data).

c) Logistic Regression:

Under the Supervised Learning approach, one of the most prominent Machine Learning algorithms is logistic regression. It's a method for predicting a categorical outcome variable from a set of independent variables. A categorical dependent variable's output is predicted using logistic regression.

i Training & Testing:

Train/Test is a way for determining your model's accuracy. It's termed Train/Test because the data set is separated into two parts: training and testing. Training takes up 80% of the time, while testing takes up 20%. In our project, we have total 270 datasets where the number of training data is 216 and testing data is 54 that maintains 80% and 20% times.

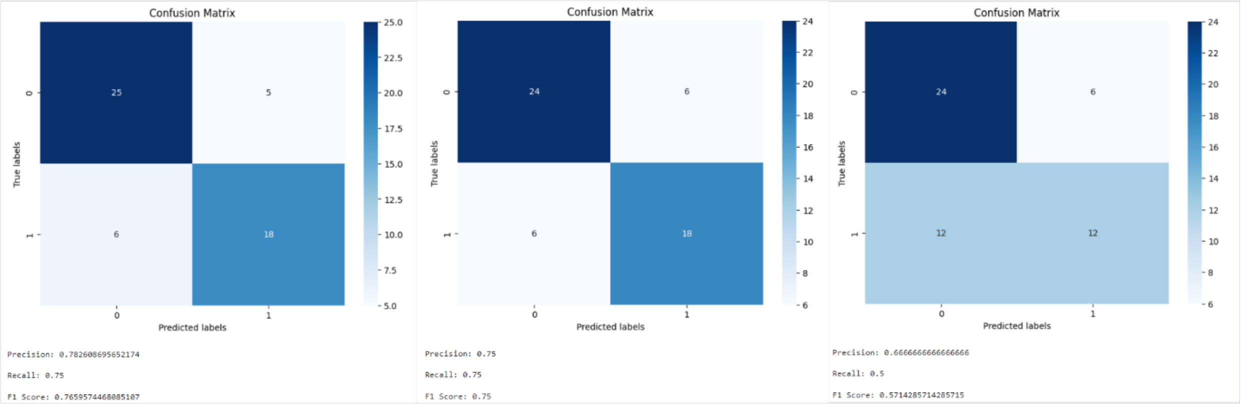
ii Model Accuracy:

Name of Algorithm	Accuracy
Support Vector Machine (SVM)	0.815
Decision Tree (DT)	0.648
Logistic Regression (LR)	0.815

So, here 81 percent accuracy was achieved using machine learning methods named Logistic Regression (LR) and Support Vector Machine (SVM) which obtained more accuracy than other methods.

V. Result:

1. Confusion Matrix:



```
Precision: 0.782608695652174

Recall: 0.75

F1 Score: 0.7659574468085107
```

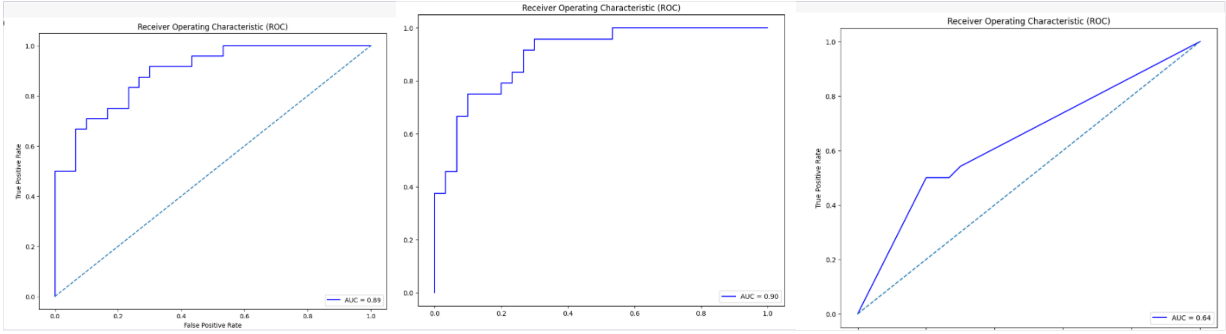
```
Precision: 0.75

Recall: 0.75

F1 Score: 0.75
```

```
Precision: 0.6666666666666666

Recall: 0.5

F1 Score: 0.5714285714285715
```

LR **SVM** **DT**

This matrix helps to understand the classification performance, particularly in binary classification. The matrix will show: True Positives (TP): Actual 'Presence' and predicted 'Presence' True Negatives (TN): Actual 'Absence' and predicted 'Absence' False Positives (FP): Actual 'Absence' but predicted 'Presence' False Negatives (FN): Actual 'Presence' but predicted 'Absence'

2. ROC-AUC Curve:



LR **SVM** **DT**

The ROC curve (Receiver Operating Characteristic curve) is a graphical plot used to show the diagnostic ability of binary classifiers. It plots the True Positive Rate against the False Positive Rate. The AUC (Area Under Curve) represents the measure of separability; an AUC of 1 implies an excellent classifier, while an AUC of 0.5 implies a worthless classifier.

Discussion & Conclusion:

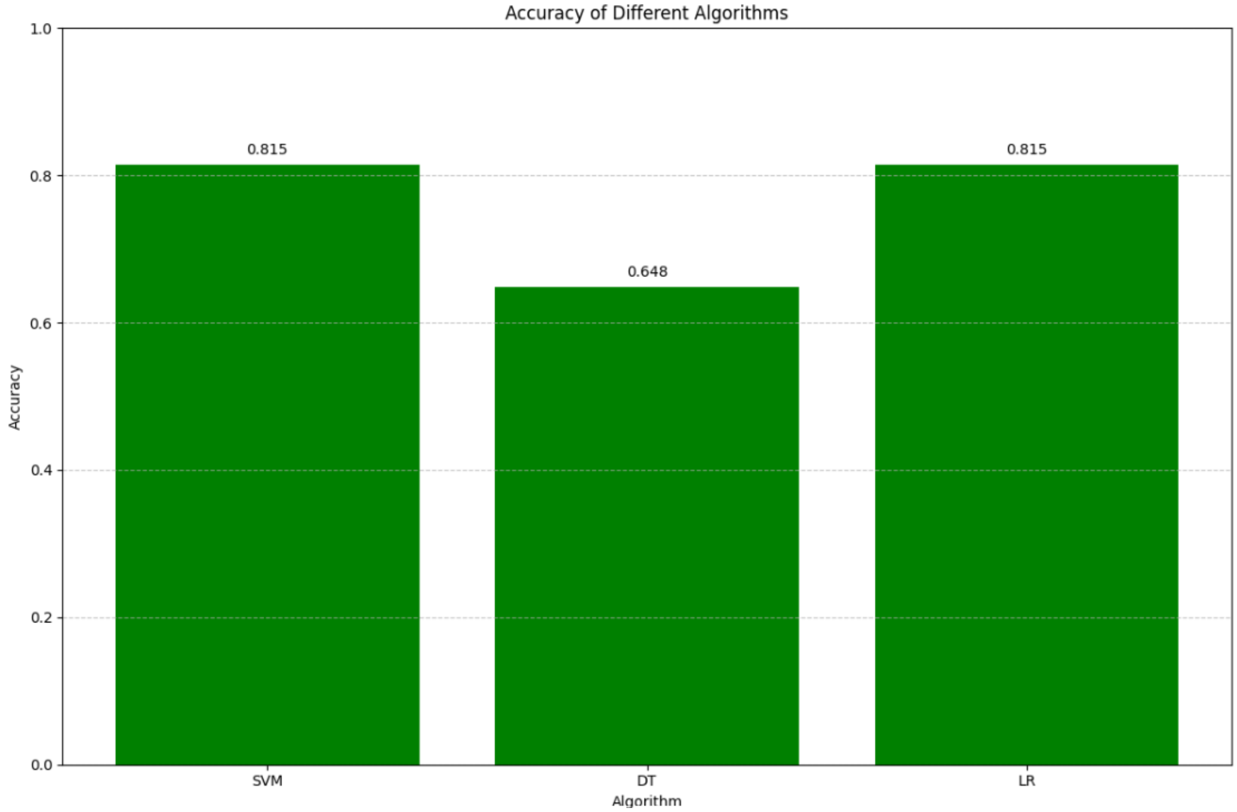


Figure: Accuracy Graph of Different Algorithms

Here, different types of algorithm's accuracy have been showcased in the graph. In the SVM algorithm, the result of accuracy stands at 81%, which is relatively low. In contrast, the Decision Tree algorithm exhibits an accuracy of 64%, The Logistic Regression algorithm shines the brightest with an impressive 81% accuracy. Among all these algorithms, the Logistic Regression algorithm and SVM algorithm emerges as the most accurate for predicting the absence or presence of heart diseases in individuals. However, it's crucial to acknowledge certain limitations in this project. Our inability to physically visit hospitals for data collection led us to rely on website-sourced data. As such, this may have influenced our outcomes, potentially hindering us from achieving more satisfactory results. Recognizing this limitation, we aim, in the future, to enhance the accuracy of our results. Plans are underway to visit multiple hospitals in Bangladesh, ensuring the collection of authentic, up-to-date heart disease datasets.

5. *References:*

i) "*Prediction of Heart Disease Using a Combination of Machine Learning and Deep Learning*" Rohit Bharti , Aditya Khamparia ,Mohammad Shabaz , Gaurav Dhiman , Sagar Pande and Parneet Singh, Article ID 8387680, 11 pages, 1 July 2021

ii) P. Ramprakash, R. Sarumathi, R. Mowriya, and S. Nithyavishnupriya, *"Heart disease using deep neural network,"* in Proceedings of the 2020 International Conference on Inventive Computation Technologies (ICICT), pp. 666–670, IEEE, Coimbatore, India, February 2020.

iii) Jyoti Soni, Ujma Ansari, Dipesh Sharma, and Sunita Soni. *Predictive data mining for medical diagnosis: An overview of heart disease prediction*. International Journal of Computer Applications, 17(8):43–48, 2011.

iv) Jesmin Nahar, Tasadduq Imam, Kevin S Tickle, and Yi-Ping Phoebe Chen. *Association rule mining to detect factors which contribute to heart disease in males and females. Expert Systems with Applications*, 40(4):1086–1093, 2013.

v) S. Kumar, *"Predicting and diagnosing of heart disease using machine learning algorithms,"* International Journal of Engineering and Computer Science, vol. 6, no. 6, pp. 2319–7242, 2017.