HEART DISEASE PREDICTION USING MACHINE LEARNING

DISSERTATION SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE AWARD OF THE DIPLOMA OF

IT ENABLE SERVICES & MANAGEMENT

VI^th^ Semester

Jan – May, 2021

Project Guide :-

Mrs. Sunita Kr. Chaurasia

Submitted by :-

Rahul kumar

1813111034

RAJOKARI INSTITUTE OF TECHNOLOGY -(GOVT),
RAJOKRI VILLAGE, RAJOKRI NEW DELHI, DELHI
110038,INDIA

Rajokari Institute of Technology

Department of Training and Technical Education, Delhi

DECLARATION

I hereby declare that the project titled **Heart Disease Prediction Using Machine Learning** Submitted by me for Diploma in *Information Technology Enable Service Management System VI*th semester to *Rajokari Institute of Technology*, *Department of Training and Technical Education, Delhi,* comprises my own work and due acknowledgement has been made in text to all other material used.

Signature of Student……………………….

Name: **Rahul kumar**

Date: …………………………

Rajokari Institute of Technology

Department of Training and Technical Education, Delhi

CERTIFICATE FROM GUIDE

It is to certify that the project entitled "**____Heart Disease Prediction Using Machine Learning submitted _____**", submitted by **Mr. Rahul kumar** to the *Rajokari Institute of Technology, Department of Training and Technical Education, Delhi,* has been completed under my supervision and the work is carried out and presented in a manner required for its acceptance to Diploma in *Information Technology Enable Service Management System VIth semester.*

Project Guide

Signature: ………………………..

Name: **Mrs. Sunita Kr. Chaurasia**

Date: ………………………..

Rajokari Institute of Technology

Department of Training and Technical Education, Delhi

EXAMINATION APPROVAL CERTIFICATE

It is to certify that we have examined the project entitled "**Heart Disease Prediction Using Machine Learning submitted**", submitted by **Mr. Rahul kumar** to the *Rajokari Institute of Technology, Department of Training and Technical Education, Delhi,* and hereby accord our approval of it as a study carried out and presented in a manner required for its acceptance *Diploma in Information Technology Enable Service Management VI*th *semester.*

Internal Examiner **External Examiner**

Signature: Signature:

Name: Name:

Date: Date:

ACKNOWLEDGEMENT

I have taken My efforts in this project. However, it would not have been possible without the kind support and help of many individuals. We would like to extend our sincere thanks to all of them.

We are highly indebted to Principal Sir. Mr. Porusu Ramanaiah And Mrs. Sunita kr. Chaurasia for her guidance and constant supervision as well as for providing necessary information regarding the project & also for her support in completing the project.

We would like to express our gratitude towards our families for their kind cooperation and encouragement which help me in completion of this project.

Our thanks and appreciations also go to people who have willingly helped us out with their abilities.

Table of Contents

Abstract

This report represents the mini-project assigned to seventh semester students for the partialfulfillment of COMP 484, Machine Learning, given by the department of computer science and engineering, KU. Cardiovascular diseases are the most common cause of death worldwide over the last few decades in the developed as well as underdeveloped and developing countries. Early detection of cardiac diseases and continuous supervision of clinicians can reduce the mortality rate. However, it is not possible to monitor patients every day in all cases accurately and consultation of a patient for 24 hours by a doctor is not available since it requires more sapience, time and expertise. In this project, we have developed and researched about models for heart disease prediction through the various heart attributes of patient and detect impending heart disease using Machine learning techniques like backward elimination algorithm, Random Forest Classifier Andric on the dataset available publicly in Kaggle Website, further evaluating the results using confusion matrix and cross validation. The early prognosis of cardiovascular diseases can aid in making decisions on lifestyle changes in high risk patients and in turn reduce the complications, which can be a great milestone in the field of medicine.

Keywords: Machine Learning, Random Forest Classifier, Cross-Validation, Backward Elimination, REFCV, Cardiovascular Diseases.

LIST OF FIGURES

LIST OF ABBREVIATIONS

1. IDE: Integrated Development Environment
2. REFCV: Recursive Feature Elimination using Cross-Validation
3. CV: Cross Validation
4. RFE: Recursive Feature Eliminatio

CHAPTER 1 : Introduction

According to the World Health Organization, every year 12 million deaths occur worldwide due to Heart Disease. The load of cardiovascular disease is rapidly increasing all over the world from the past few years. Many researches have been conducted in attempt to pinpoint the most influential factors of heart disease as well as accurately predict the overall risk. Heart Disease is even highlighted as a silent killer which leads to the death of the person without obvious symptoms. The early diagnosis of heart disease plays a vital role in making decisions on lifestyle changes in high-risk patients and in turn reduce the complications. This project aims to predict future Heart Disease by analysing data of patients which classifies whether they have heart disease or not using machine-learning algorithms.

1.1 Problem definition

The major challenge in heart disease is its detection. There are instruments available which can predict heart disease but either they are expensive or are not efficient to calculate chance of heart disease in human. Early detection of cardiac diseases can decrease the mortality rate and overall complications. However, it is not possible to monitor patients every day in all cases accurately and consultation of a patient for 24 hours by a doctor is not available since it requires more sapience, time and expertise. Since we have a good amount of data in today’s world, we can use various machine learning algorithms to analyse the data for hidden patterns. The hidden patterns can be used for health diagnosis in medicinal data.

1.2 Motivation

Machine learning techniques have been around us and has been compared and used for analysis for many kinds of data science applications. The major motivation behind this research-based project was to explore the feature selection methods, data preparation and processing behind the training models in the machine learning. With first hand models and libraries, the challenge we face today is data where beside their abundance, and our cooked models, the accuracy we see during training, testing and actual validation has a higher variance. Hence this project is carried out with the motivation to explore behind the models, and further implement Random Forest Classifier model to train the obtained data. Furthermore, as the whole machine learning is motivated to develop an appropriate computer-based system and decision support that can aid to early detection of heart disease, in this project we have developed a model which classifies if patient will have heart disease in ten years or not based on various features (i.e., potential risk factors that can cause heart disease) using Random Forest Classifier. Hence, the early prognosis of cardiovascular diseases can aid in making decisions on lifestyle changes in high-risk patients and in turn reduce the complications, which ca n be a great milestone in the field of medicine.

1.3 Aim and Objectives

The main objective of developing this project are :

1. To develop machine learning model to predict future possibility of heart disease by implementing RandomForestClassifier.

2. To determine significant risk factors based on medical dataset which may lead to heart disease.

3. To analyse feature selection methods and understand their working principle.

4. The main objective of this research is to develop a heart prediction system. The system can discover and extract hidden knowledge associated with diseases from a historical heart data set

5. Heart disease prediction system aims to exploit data mining techniques on medical data set to assist in the prediction of the heart diseases.

6.Provides new approach to concealed patterns in the data.

7.Helps avoid human biasness.

8.Reduce the cost of medical tests.

1.4 Scope and Limitation.

1.4.1 Scope.

Here the scope of the project is that integration of clinical decision support with computer-based patient records could reduce medical errors, enhance patient safety, decrease unwanted practice variation, and improve patient outcome. This suggestion is promising as data modeling and analysis tools, e.g., data mining, have the potential to generate a knowledge-rich environment which can help to significantly improve the quality of clinical decisions

1.4.2 Limitations.

Medical diagnosis is considered as a significant yet intricate task that needs to be carried out precisely and efficiently. The automation of the same would be highly beneficial. Clinical decisions are often made based on doctor’s intuition and experience rather than on the knowledge rich data hidden in the database. This practice leads to unwanted biases, errors and excessive medical costs which affects the quality of service provided to patients. Data mining have the potential to generate a knowledge-rich environment which can help to significantly improve the quality of clinical decisions.

1.5 Goal:

- Predict whether a patient should be diagnosed with Heart Disease. This is a binary outcome.
- Positive (+) = 1, patient diagnosed with Heart Disease
- Negative (-) = 0, patient not diagnosed with Heart Disease

- Experiment with various Classification Models & see which yields greatest accuracy.
- Examine trends & correlations within our data
- Determine which features are most important to Positive/Negative Heart Disease diagnosis

1.6 Features & Predictor

Our Predictor (Y, Positive or Negative diagnosis of Heart Disease) is determined by 13 features (X):

1. age (#)

2. sex : 1= Male, 0= Female (Binary)

3. (cp)chest pain type (4 values -Ordinal):Value 1: typical angina ,Value 2: atypical angina, Value 3: non-anginal pain , Value 4: asymptomatic

4. (trestbps) resting blood pressure (#)

5. (chol) serum cholesterol in mg/dl (#)

6. (fbs)fasting blood sugar > 120 mg/dl(Binary)(1 = true; 0 = false)

7. (restecg) resting electrocardiography results(values 0,1,2)

8. (thalach) maximum heart rate achieved (#)

9. (exang) exercise induced angina (binary) (1 = yes; 0 = no)

10. (oldpeak) = ST depression induced by exercise relative to rest (#)

11. (slope) of the peak exercise ST segment (Ordinal) (Value 1: up sloping , Value 2: flat , Value 3: down sloping)

12. (ca) number of major vessels (0–3, Ordinal) colored by fluoroscopy

13. (thal) maximum heart rate achieved — (Ordinal): 3 = normal; 6 = fixed defect; 7 = reversible defect

CHAPTER 2: RELATED WORKS

Now we've got our data split into training and test sets, it's time to build a machine learning model.

We'll train it (find the patterns) on the training set.

And we'll test it (use the patterns) on the test set.

We're going to try 3 different machine learning models:

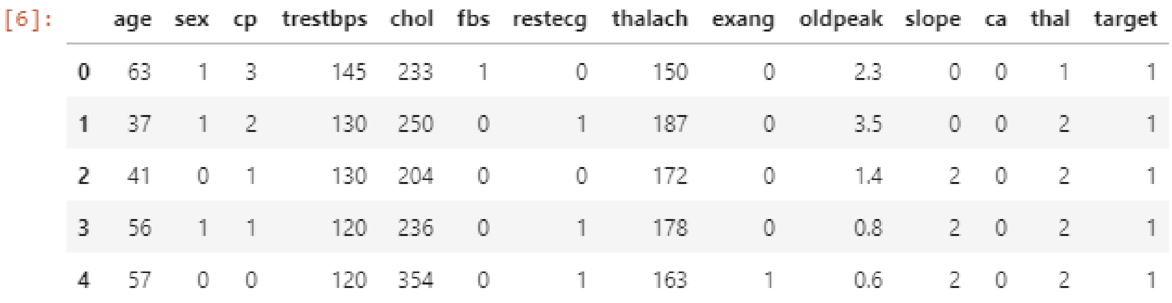
1. Logistic Regression
2. K-Nearest Neighbours Classifier
3. Support Vector machine
4. Decision Tree Classifier
5. Random Forest Classifier

S.NO	Model	Training Accuracy %	Testing Accuracy %
1	Logistic Regression	74	74
2	K-Nearest Neighbours Classifier	75	75
3	Support Vector machine	75	75
4	Decision Tree Classifier	69	69
5	Random Forest Classifier	86	86

CHAPTER 3

3.1 DATASETS

he dataset is publicly available on the Kaggle Website at [4] which is from an ongoing cardiovascular study on residents of the town of Framingham, Massachusetts. It provides patient information which includes over 4000 records and 14 attributes. The attributes include: age, sex, chest pain type, resting blood pressure, serum cholesterol, fasting, sugar blood, resting electrocardiographic results, maximum heart rate, exercise induced angina, ST depression induced by exercise, slope of the peak exercise, number of major vessels, and target ranging from 0 to 2, where 0 is absence of heart disease. The data set is in csv (Comma Separated Value) format which is further prepared to data frame as supported by pandas library in python.



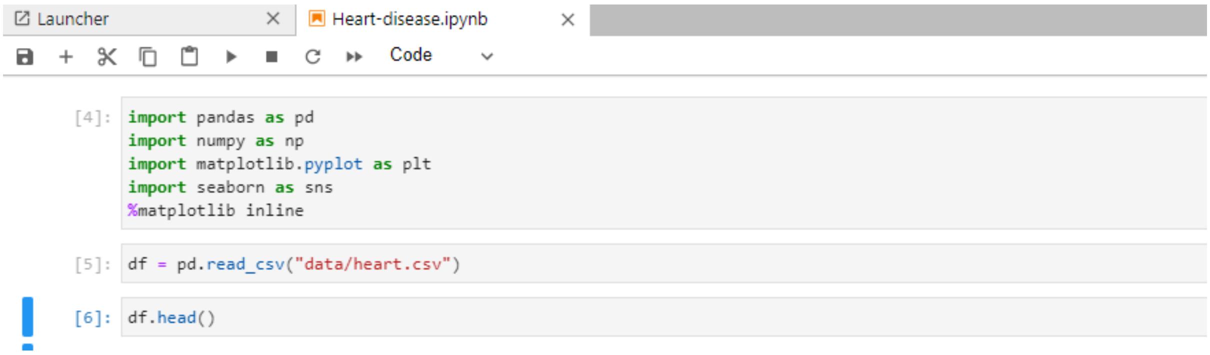
[6]:

	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

Figure 1 Original Dataset Snapshot

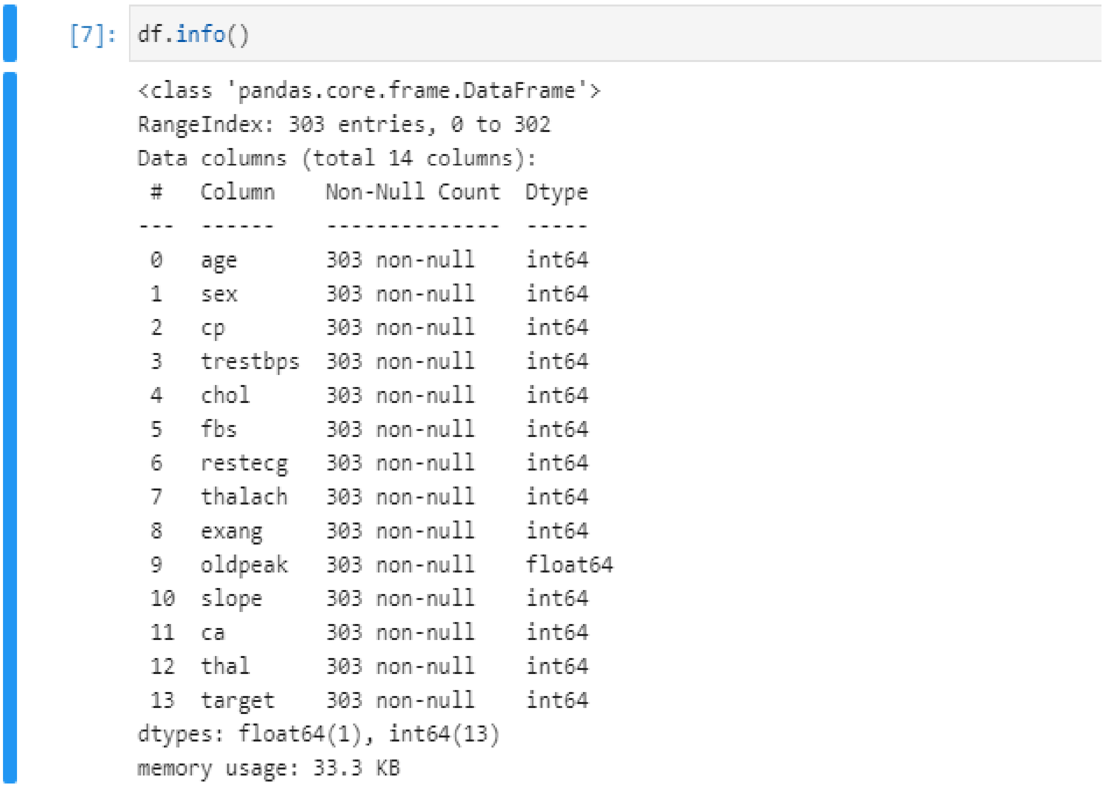
The education data is irrelevant to the heart disease of an individual, so it is dropped. Further with this dataset pre-processing and experiments are then carried out

3.2 DATA WINDING



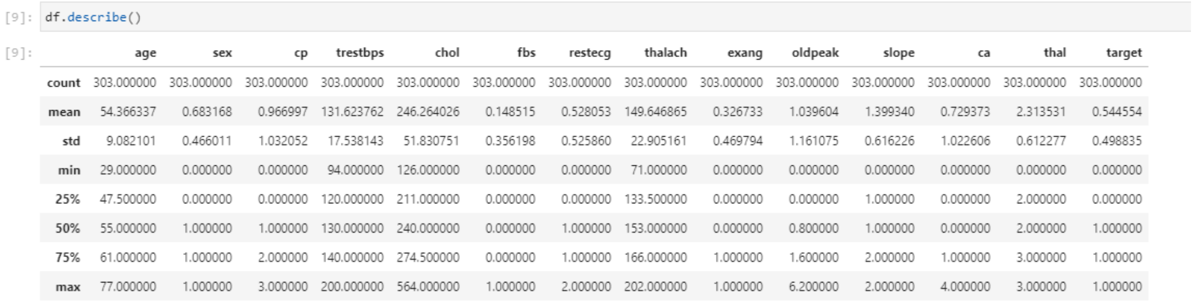
```
[4]: import pandas as pd
     import numpy as np
     import matplotlib.pyplot as plt
     import seaborn as sns
     %matplotlib inline

[5]: df = pd.read_csv("data/heart.csv")

[6]: df.head()
```

Figure 2 Import libaray & read csv file



```
[7]: df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 303 entries, 0 to 302
Data columns (total 14 columns):
 #   Column    Non-Null Count  Dtype  
---  ------    --------------  -----  
 0   age       303 non-null    int64  
 1   sex       303 non-null    int64  
 2   cp        303 non-null    int64  
 3   trestbps  303 non-null    int64  
 4   chol      303 non-null    int64  
 5   fbs       303 non-null    int64  
 6   restecg   303 non-null    int64  
 7   thalach   303 non-null    int64  
 8   exang     303 non-null    int64  
 9   oldpeak   303 non-null    float64
 10  slope     303 non-null    int64  
 11  ca        303 non-null    int64  
 12  thal      303 non-null    int64  
 13  target    303 non-null    int64  
dtypes: float64(1), int64(13)
memory usage: 33.3 KB
```

Figure 3 Describe data & dtypes

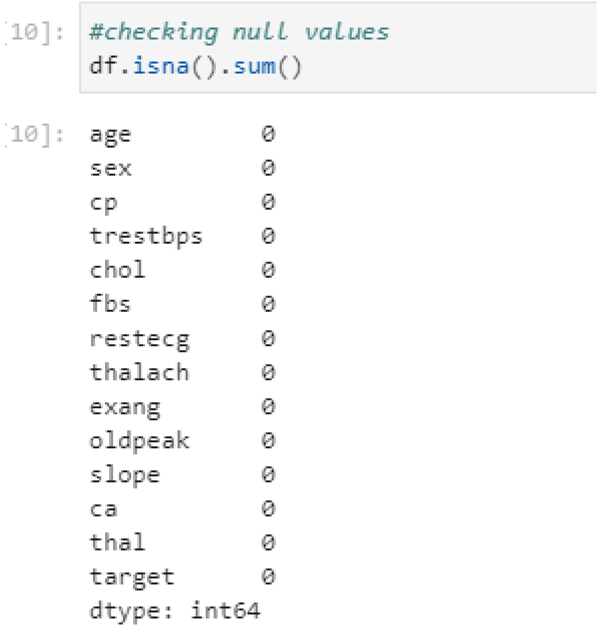


```
[9]: df.describe()
```

[9]:

	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

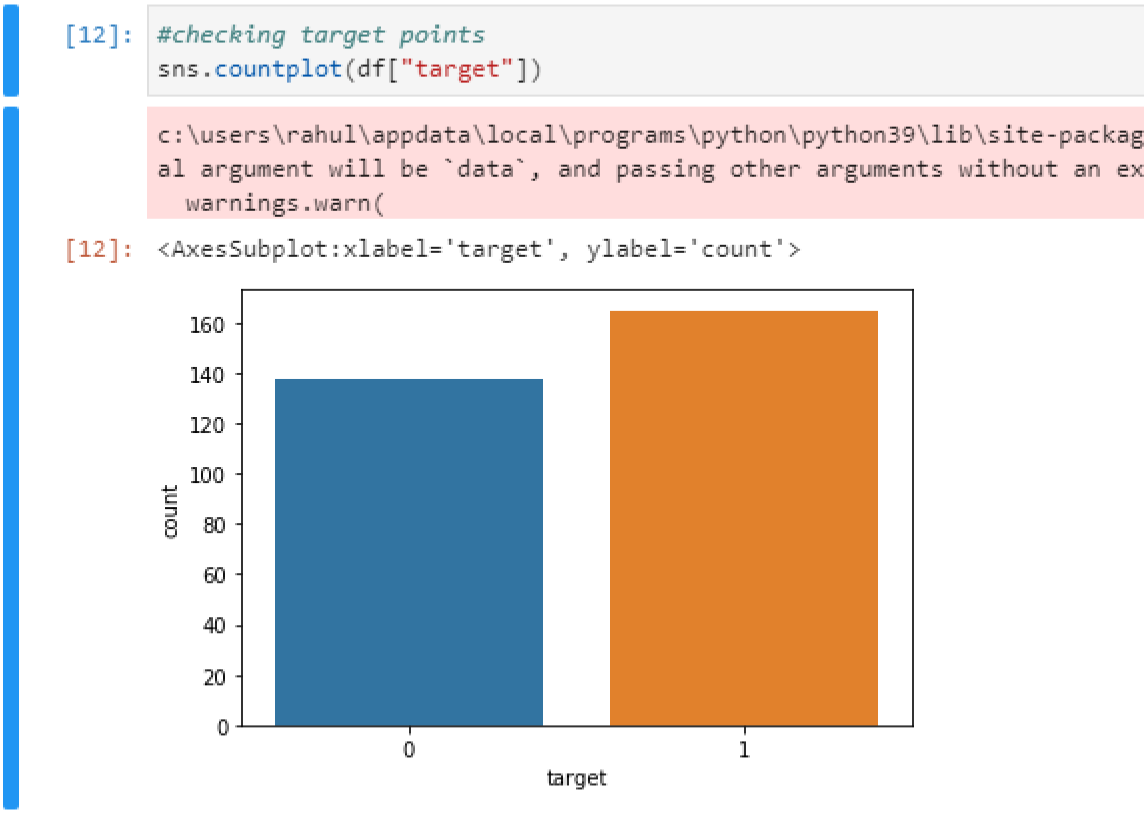
Figure 4 summarizes the count, mean, standard deviation, min, and max for numeric variables.



```
10]: #checking null values
     df.isna().sum()

10]: 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
```

Figure 5 Checking null values

Lets see if theirs a *good proportion* between our positive & negative **binary predictor.**



```
[12]: #checking target points
      sns.countplot(df["target"])

      c:\users\rahul\appdata\local\programs\python\python39\lib\site-packag
      al argument will be `data`, and passing other arguments without an ex
        warnings.warn(

[12]: <AxesSubplot:xlabel='target', ylabel='count'>
```

Figure 6 It appears we have a good balance between the two binary outputs.

CHAPATER 4: METHODS AND ALGORITHMS USED TOOLS

4.1 Random Forest Classifier using Scikit-learn

Random Forest Classifier is ensemble algorithm. In next one or two posts we shall explore such algorithms. Ensembled algorithms are those which combines more than one algorithms of same or different kind for classifying objects. For example, running prediction over Naive Bayes, SVM and Decision Tree and then taking vote for final consideration of class for test object.

4.1.1 Random Forest Classifier

Random forest classifier creates a set of decision trees from randomly selected subset of training set. It then aggregates the votes from different decision trees to decide the final class of the test object.
In Laymen’s term,

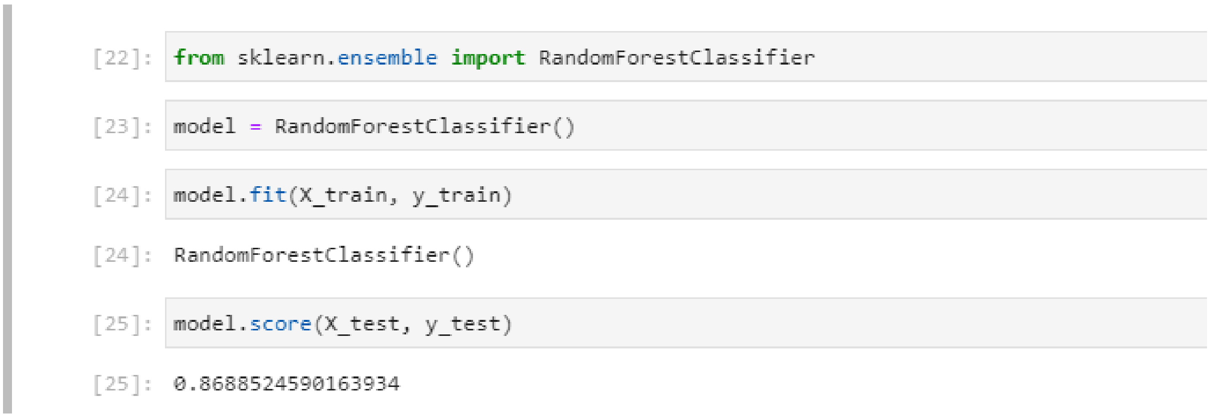
Suppose training set is given as : [X1, X2, X3, X4] with corresponding labels as [L1, L2, L3, L4], random forest may create three decision trees taking input of subset for example,

1. [X1, X2, X3]
2. [X1, X2, X4]
3. [X2, X3, X4]

Using Random Forest Classifier

The code for using Random Forest Classifier is similar to previous classifiers.

1. Import library
2. Create model
3. Train
4. Predict



```
[22]: from sklearn.ensemble import RandomForestClassifier

[23]: model = RandomForestClassifier()

[24]: model.fit(X_train, y_train)

[24]: RandomForestClassifier()

[25]: model.score(X_test, y_test)

[25]: 0.8688524590163934
```

Figure 7 Model import

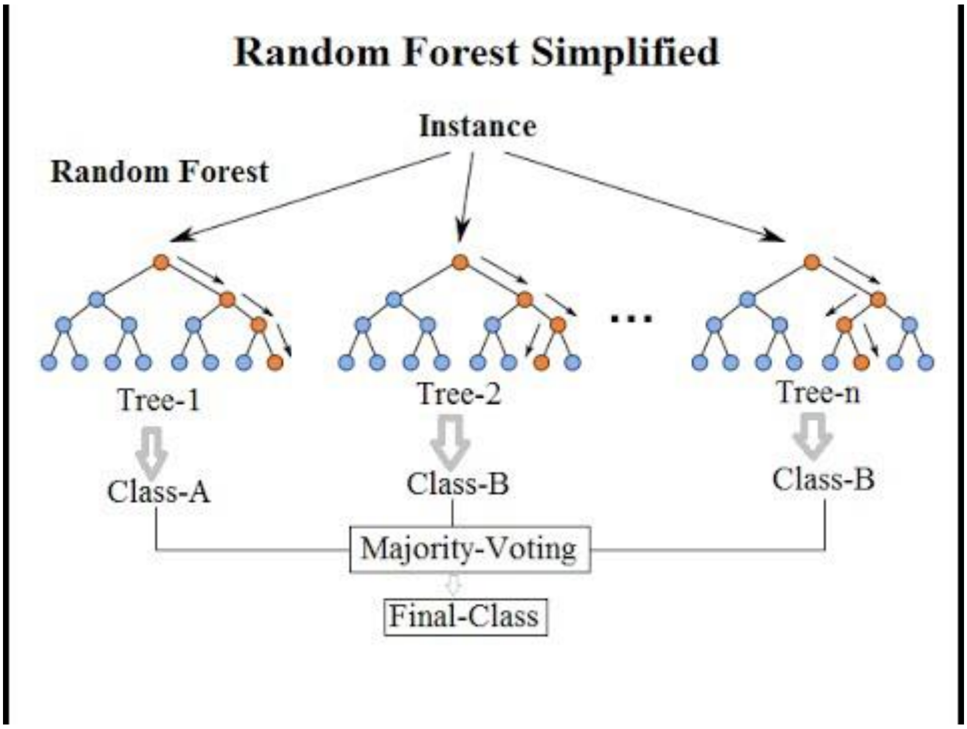


Figure 8 Random Forest Classification

4.2 TOOLS

For application development, the following Software Requirements are:

Operating System: Windows 7 or any Linux Debian Distro.

Language: R and Shiny Tools: RStudio IDE, Microsoft Excel (Optional).

Technologies used: R, Unix, Shiny

4.3 SOFTWARE REQUIREMENTS

Operating System Any OS with clients to access the
internet Network Wi-Fi Internet or cellular
Network

Create and design Data Flow and Context Diagram

Versioning Control

Medium to find reference to do system testing,

HARDWARE REQUIREMENTS

For application development, the following Software Requirements are:

Processor: Intel or high

RAM: 1024 MB

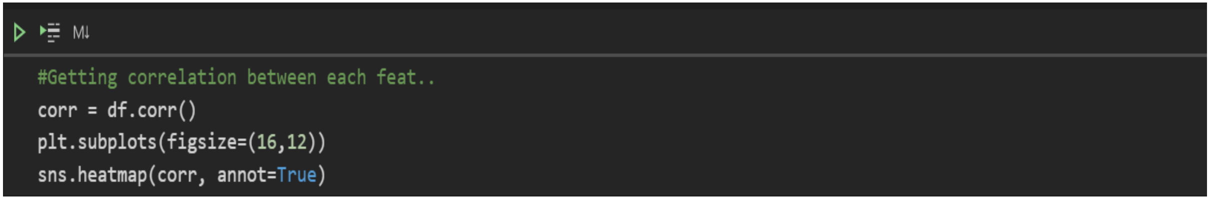
Space on disk: minimum 100mb For running the application:

Device: Any device that can access the internet Minimum space to execute: 20 MB

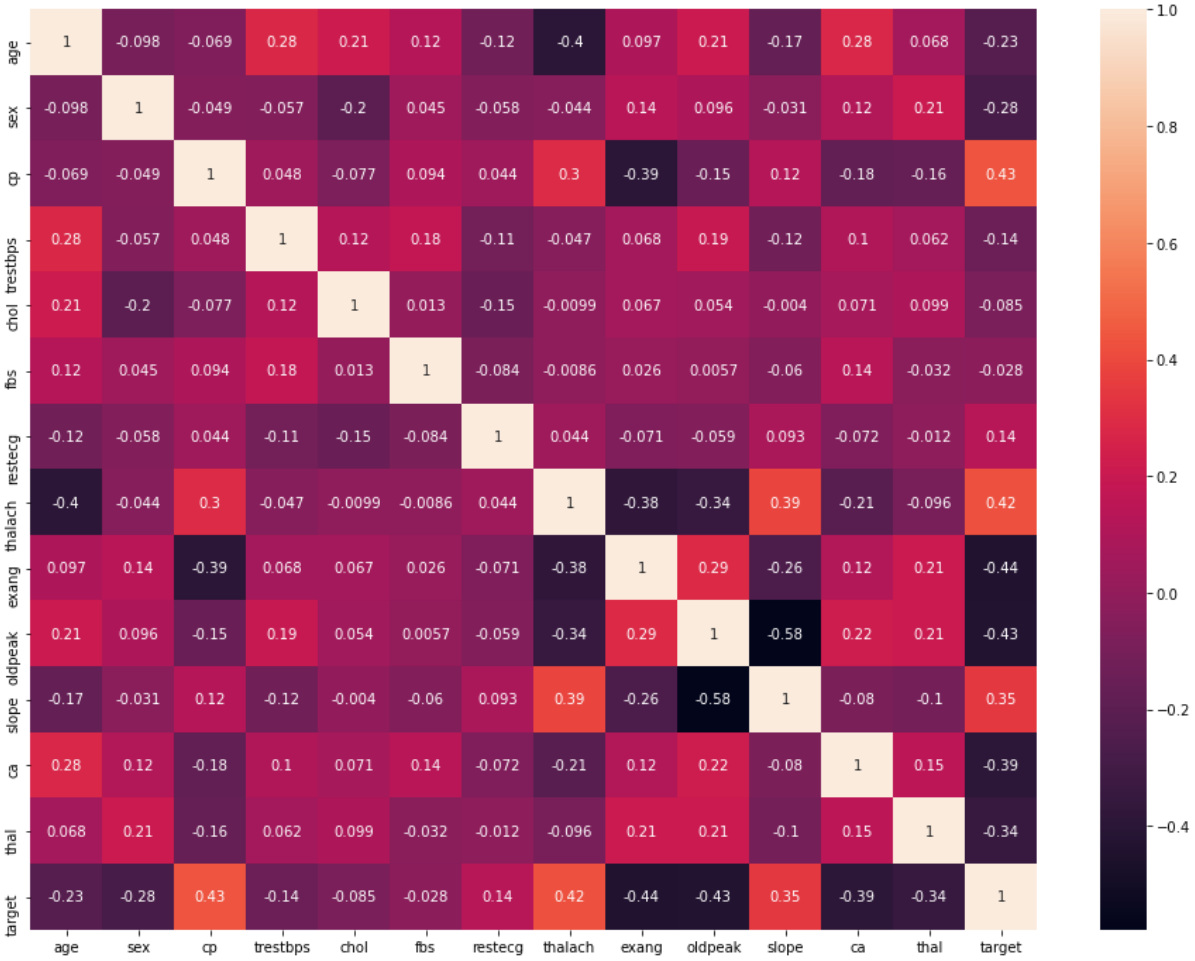
CHAPTER 5 : EXPERIMENTS

5.1 Exploratory Data Analysis

Correlation Matrix visualization Before Feature Selection shows It shows that there is no single feature that has a very high correlation with our target value. Also, some of the features have a negative correlation with the target value and some have positive. The data was also visualized through plots and bar graphs



```
#Getting correlation between each feat..
corr = df.corr()
plt.subplots(figsize=(16,12))
sns.heatmap(corr, annot=True)
```



	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
age	1	-0.098	-0.069	0.28	0.21	0.12	-0.12	-0.4	0.097	0.21	-0.17	0.28	0.068	-0.23
sex	-0.098	1	-0.049	-0.057	-0.2	0.045	-0.058	-0.044	0.14	0.096	-0.031	0.12	0.21	-0.28
cp	-0.069	-0.049	1	0.048	-0.077	0.094	0.044	0.3	-0.39	-0.15	0.12	-0.18	-0.16	0.43
trestbps	0.28	-0.057	0.048	1	0.12	0.18	-0.11	-0.047	0.068	0.19	-0.12	0.1	0.062	-0.14
chol	0.21	-0.2	-0.077	0.12	1	0.013	-0.15	-0.0099	0.067	0.054	-0.004	0.071	0.099	-0.085
fbs	0.12	0.045	0.094	0.18	0.013	1	-0.084	-0.0086	0.026	0.0057	-0.06	0.14	-0.032	-0.028
restecg	-0.12	-0.058	0.044	-0.11	-0.15	-0.084	1	0.044	-0.071	-0.059	0.093	-0.072	-0.012	0.14
thalach	-0.4	-0.044	0.3	-0.047	-0.0099	-0.0086	0.044	1	-0.38	-0.34	0.39	-0.21	-0.096	0.42
exang	0.097	0.14	-0.39	0.068	0.067	0.026	-0.071	-0.38	1	0.29	-0.26	0.12	0.21	-0.44
oldpeak	0.21	0.096	-0.15	0.19	0.054	0.0057	-0.059	-0.34	0.29	1	-0.58	0.22	0.21	-0.43
slope	-0.17	-0.031	0.12	-0.12	-0.004	-0.06	0.093	0.39	-0.26	-0.58	1	-0.08	-0.1	0.35
ca	0.28	0.12	-0.18	0.1	0.071	0.14	-0.072	-0.21	0.12	0.22	-0.08	1	0.15	-0.39
thal	0.068	0.21	-0.16	0.062	0.099	-0.032	-0.012	-0.096	0.21	0.21	-0.1	0.15	1	-0.34
target	-0.23	-0.28	0.43	-0.14	-0.085	-0.028	0.14	0.42	-0.44	-0.43	0.35	-0.39	-0.34	1

Figure 9 Within seconds, you can see whether something is positively or negatively correlated with our predictor (target)

Correlation Matrix- let’s you see correlations between all variables.

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:

Value 1: typical angina ,

Value 2: atypical angina,

Value 3: non-anginal pain ,

Value 4: asymptomatic.

In addition, we see a **negative correlation** between exercise induced angina (exang) & our predictor. This makes sense because when you excercise, your *heart requires more blood, but narrowed arteries slow down blood flow*.

Pairplots are also a great way to immediately see the correlations between all variables. But you will see me make it with only continuous columns from our data, because with so many features, it can be difficult to see each one. So instead I will make a pairplot with only our continuous features.

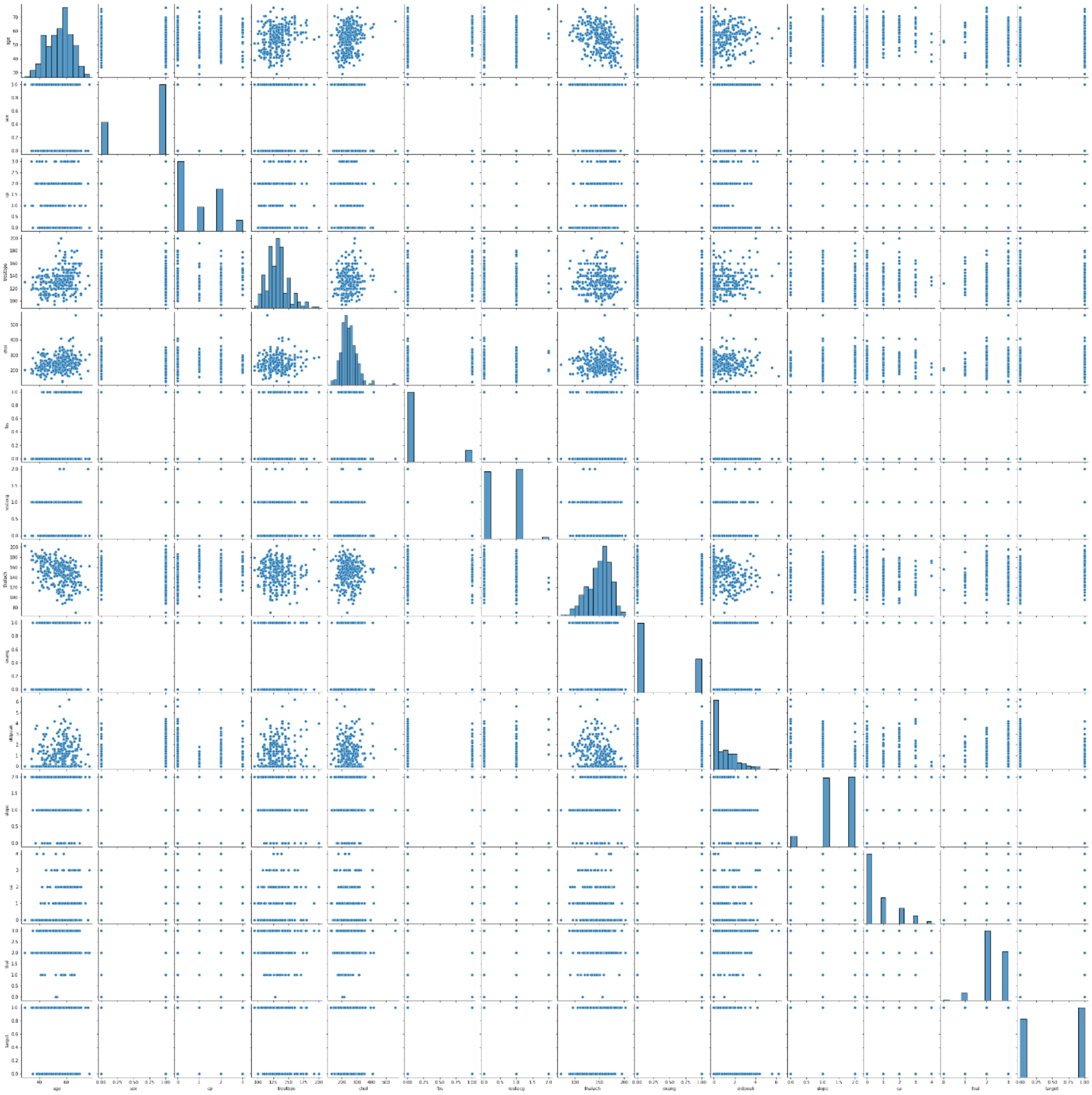
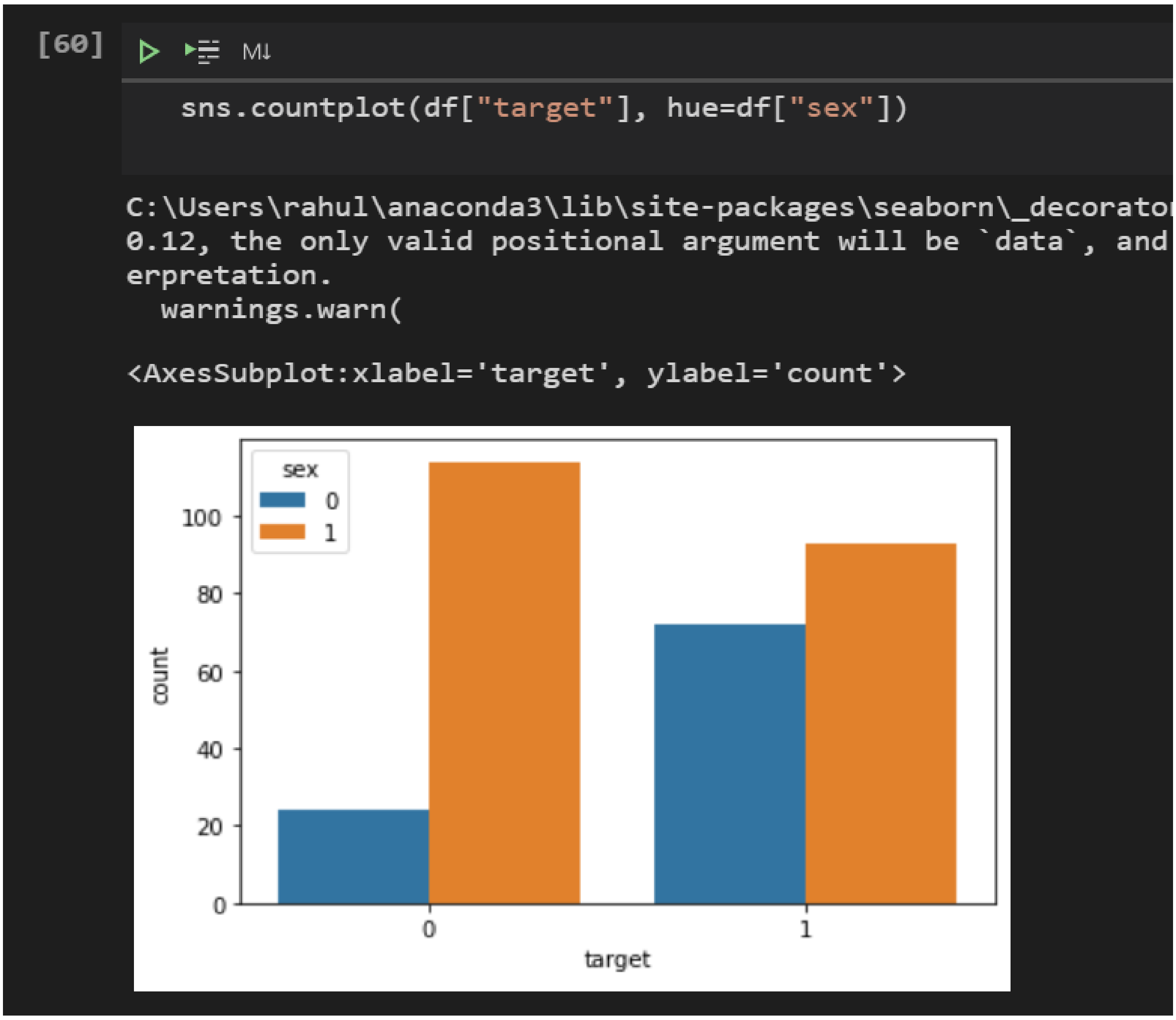


Figure 10 Chose to make a smaller pairplot with only the continuous variables, to dive deeper into the relationships. Also a great way to see if theirs a positive or negative correlation!



```
[60]
sns.countplot(df["target"], hue=df["sex"])
```

```
C:\Users\rahul\anaconda3\lib\site-packages\seaborn\_decorato
0.12, the only valid positional argument will be `data`, and
erpretation.
  warnings.warn(

<AxesSubplot:xlabel='target', ylabel='count'>
```

ST segment depression occurs because when the ventricle is at rest and therefore repolarized. If the trace in the ST segment is abnormally low below the baseline, this *can lead to this Heart Disease*. This is **supports** the plot above because low ST Depression yields people at greater risk for heart disease. While a high ST depression is considered normal & healthy. The “*slope*” hue, refers to the peak exercise ST segment, with values: 0: upsloping , 1: flat , 2: downsloping). Both positive & negative heart disease patients exhibit **equal distributions** of the 3 slope categories.

Chapter 6 : Machine Learning + Predictive Analytics

6.1 Prepare Data for Modeling

To prepare data for modeling, just remember ASN (Assign,Split, Normalize).

Assign the 13 features to X, & the last column to our classification predictor, y

```
X = data.iloc[:, :-1].values
y = data.iloc[:, -1].values
```

Split: the data set into the Training set and Test set

```
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(X,y,test_size = 0.2, random_state = 1)
```

Normalize: Standardizing the data will transform the data so that its distribution will have a mean of 0 and a standard deviation of 1.

```
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
x_train = sc.fit_transform(x_train)
x_test = sc.transform(x_test)
```

6.2 Modeling /Training

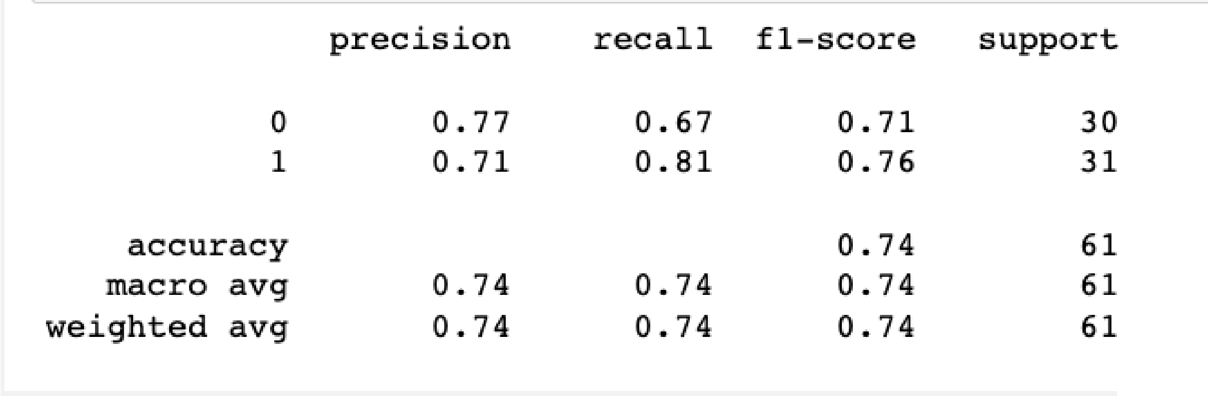
Now we'll Train various Classification Models on the Training set & see which yields the highest accuracy. We will compare the accuracy of Logistic Regression, K-NN (k-Nearest Neighbours), SVM (Support Vector Machine), Decision Trees, Random Forest.

Note: these are all supervised learning models.

Model 1: Logistic Regression

```
from sklearn.metrics import classification_report
from sklearn.linear_model import LogisticRegression

model1 = LogisticRegression(random_state=1) # get instance
of model
model1.fit(x_train, y_train) # Train/Fit model

y_pred1 = model1.predict(x_test) # get y predictions
```

```
print(classification_report(y_test, y_pred1)) # output
accuracy
```

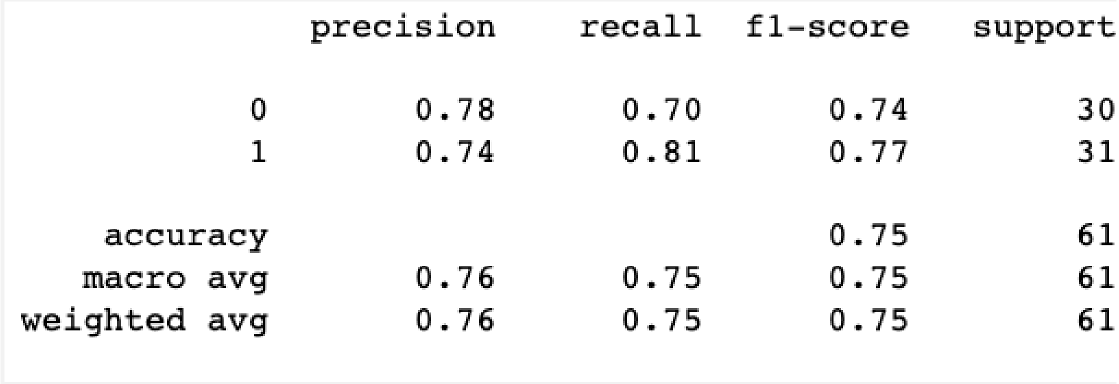


```
              precision    recall  f1-score   support

           0       0.77      0.67      0.71        30
           1       0.71      0.81      0.76        31

    accuracy                           0.74        61
   macro avg       0.74      0.74      0.74        61
weighted avg       0.74      0.74      0.74        61
```

Accuracy 74%

Model 2: K-NN (K-Nearest Neighbors)

```
from sklearn.metrics import classification_report
from sklearn.neighbors import KNeighborsClassifier

model2 = KneighborsClassifier() # get instance of model
model2.fit(x_train, y_train) # Train/Fit model

y_pred2 = model2.predict(x_test) # get y predictions
print(classification_report(y_test, y_pred2)) # output
accuracy
```

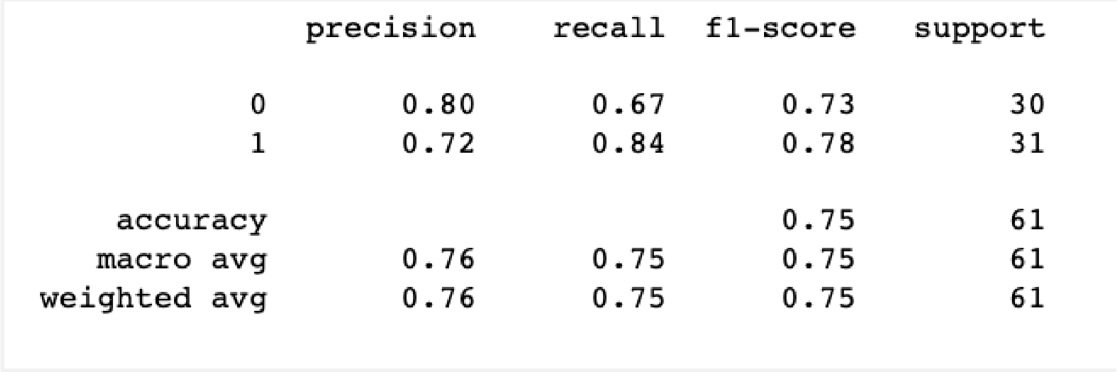


```
              precision    recall  f1-score   support

           0       0.78      0.70      0.74        30
           1       0.74      0.81      0.77        31

    accuracy                           0.75        61
   macro avg       0.76      0.75      0.75        61
weighted avg       0.76      0.75      0.75        61
```

Accuracy 75%

Model 3: SVM (Support Vector Machine)

```
from sklearn.metrics import classification_report
from sklearn.svm import SVC

model3 = SVC(random_state=1) # get instance of model
model3.fit(x_train, y_train) # Train/Fit model

y_pred3 = model3.predict(x_test) # get y predictions
print(classification_report(y_test, y_pred3)) # output
accuracy
```

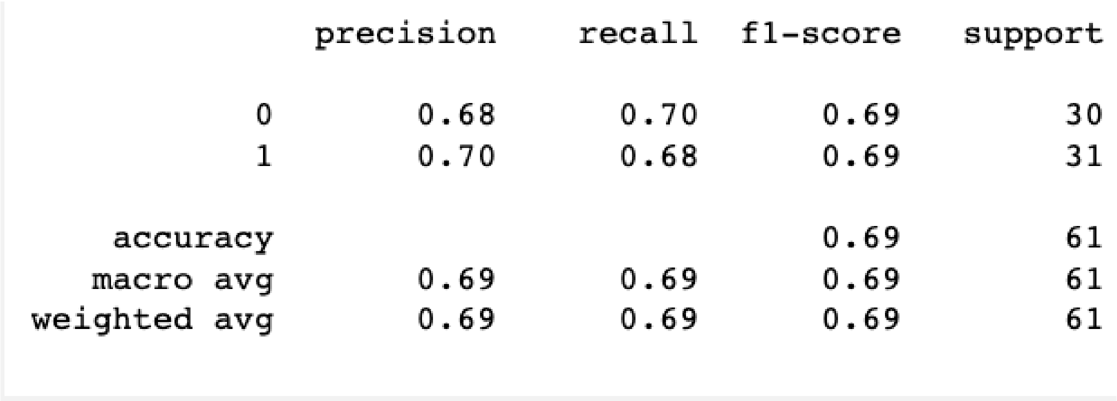


```
              precision    recall  f1-score   support

           0       0.80      0.67      0.73        30
           1       0.72      0.84      0.78        31

    accuracy                           0.75        61
   macro avg       0.76      0.75      0.75        61
weighted avg       0.76      0.75      0.75        61
```

Accuracy 75%

Model 4: Decision Trees

```
from sklearn.metrics import classification_report
from sklearn.tree import DecisionTreeClassifier

model5 = DecisionTreeClassifier(random_state=1) # get
instance of model
model5.fit(x_train, y_train) # Train/Fit model

y_pred5 = model5.predict(x_test) # get y predictions
print(classification_report(y_test, y_pred5)) # output
accuracy
```

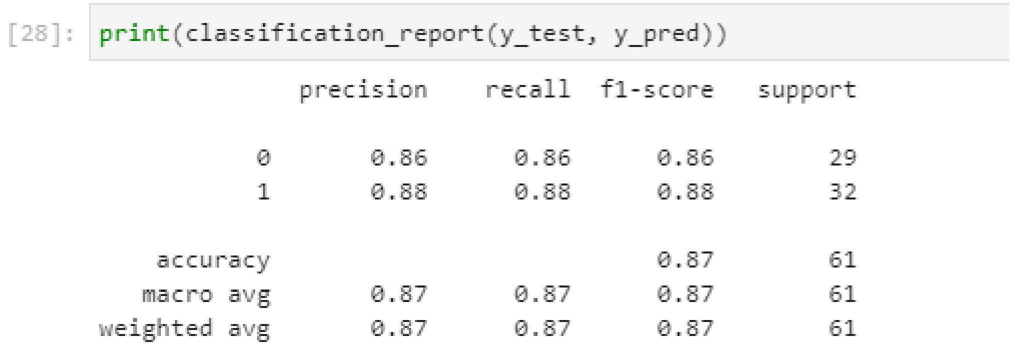


```
              precision    recall  f1-score   support

           0       0.68      0.70      0.69        30
           1       0.70      0.68      0.69        31

    accuracy                           0.69        61
   macro avg       0.69      0.69      0.69        61
weighted avg       0.69      0.69      0.69        61
```

Accuracy 69%

Model 5: Random Forest 🏆

```
from sklearn.metrics import classification_report
from sklearn.ensemble import RandomForestClassifier

model6 = RandomForestClassifier(random_state=1)# get
instance of model
model6.fit(x_train, y_train) # Train/Fit model

y_pred6 = model6.predict(x_test) # get y predictions
print(classification_report(y_test, y_pred6)) # output
accuracy
```



```
[28]: print(classification_report(y_test, y_pred))

              precision    recall  f1-score   support

           0       0.86      0.86      0.86        29
           1       0.88      0.88      0.88        32

    accuracy                           0.87        61
   macro avg       0.87      0.87      0.87        61
weighted avg       0.87      0.87      0.87        61
```

Figure 11 accuracy 86

Precision, Recall, F1-score and Support:

Precision : be “how many are correctly classified among that class”

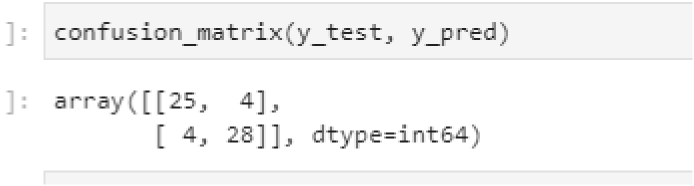
Recall : “how many of this class you find over the whole number of element of this class”

F1-score : harmonic mean of precision and recall values.
F1 score reaches its best value at 1 and worst value at 0.
F1 Score = 2 x ((precision x recall) / (precision + recall))

Support: # of samples of the true response that lie in that class.

6.3 Making the Confusion Matrix

```
from sklearn.metrics import confusion_matrix, accuracy_score
cm = confusion_matrix(y_test, y_pred6)
print(cm)
accuracy_score(y_test, y_pred6)
```



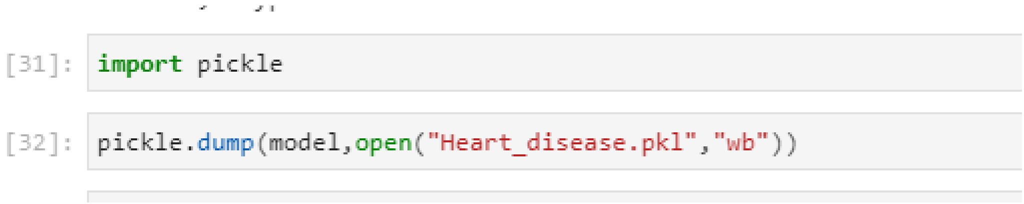
```
]: confusion_matrix(y_test, y_pred)

]: array([[25,  4],
          [ 4, 28]], dtype=int64)
```

6.4 MODEL SAVE

Approach 1 : Pickle approach

Following lines of code, the LR_Model which we created in the previous step is saved to file, and then loaded as a new object called Pickled_RL_Model.

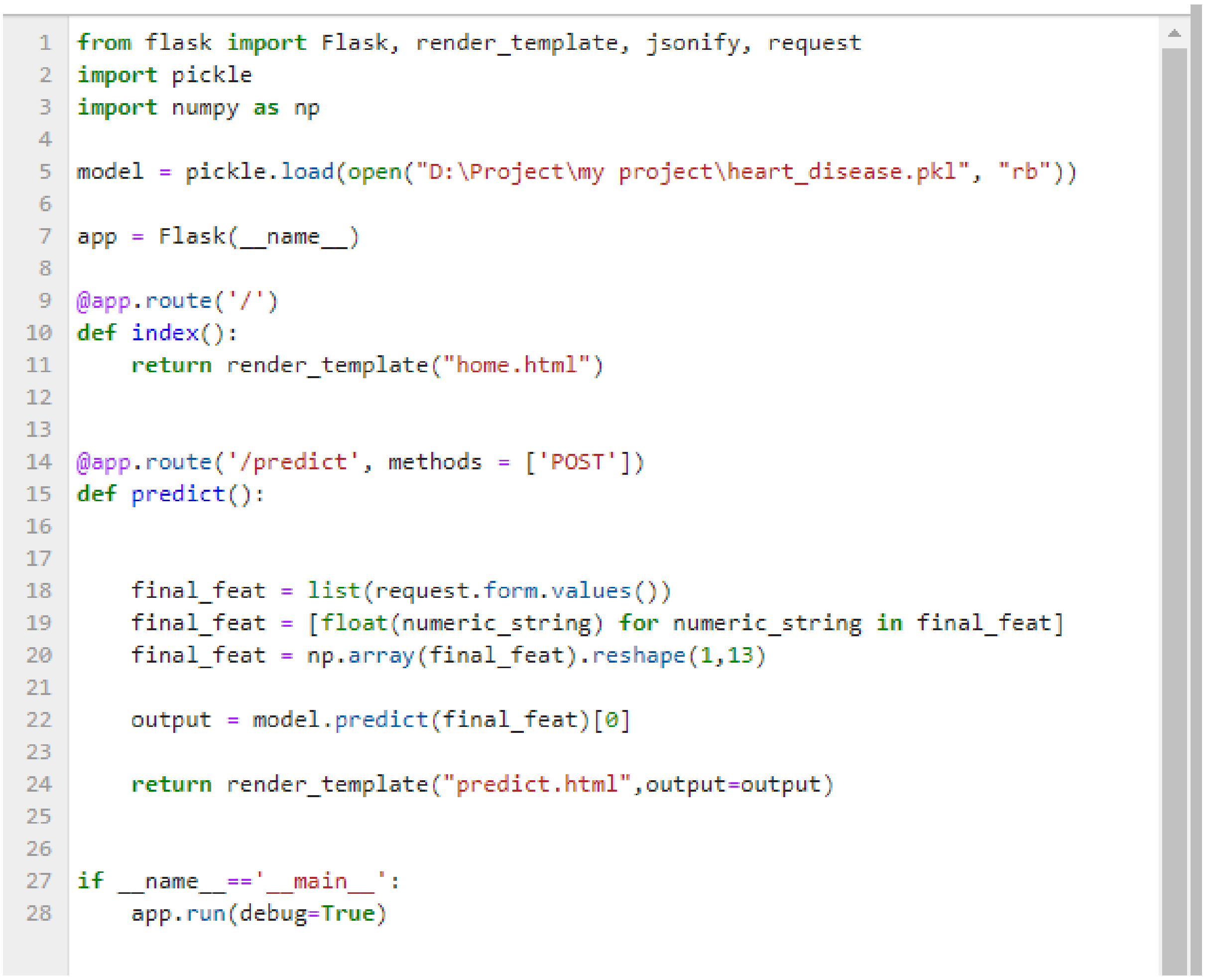
The loaded model is then used to calculate the accuracy score and predict outcomes on new unseen (test) data.



```
[31]: import pickle

[32]: pickle.dump(model,open("Heart_disease.pkl","wb"))
```

Figure 12 Saving model

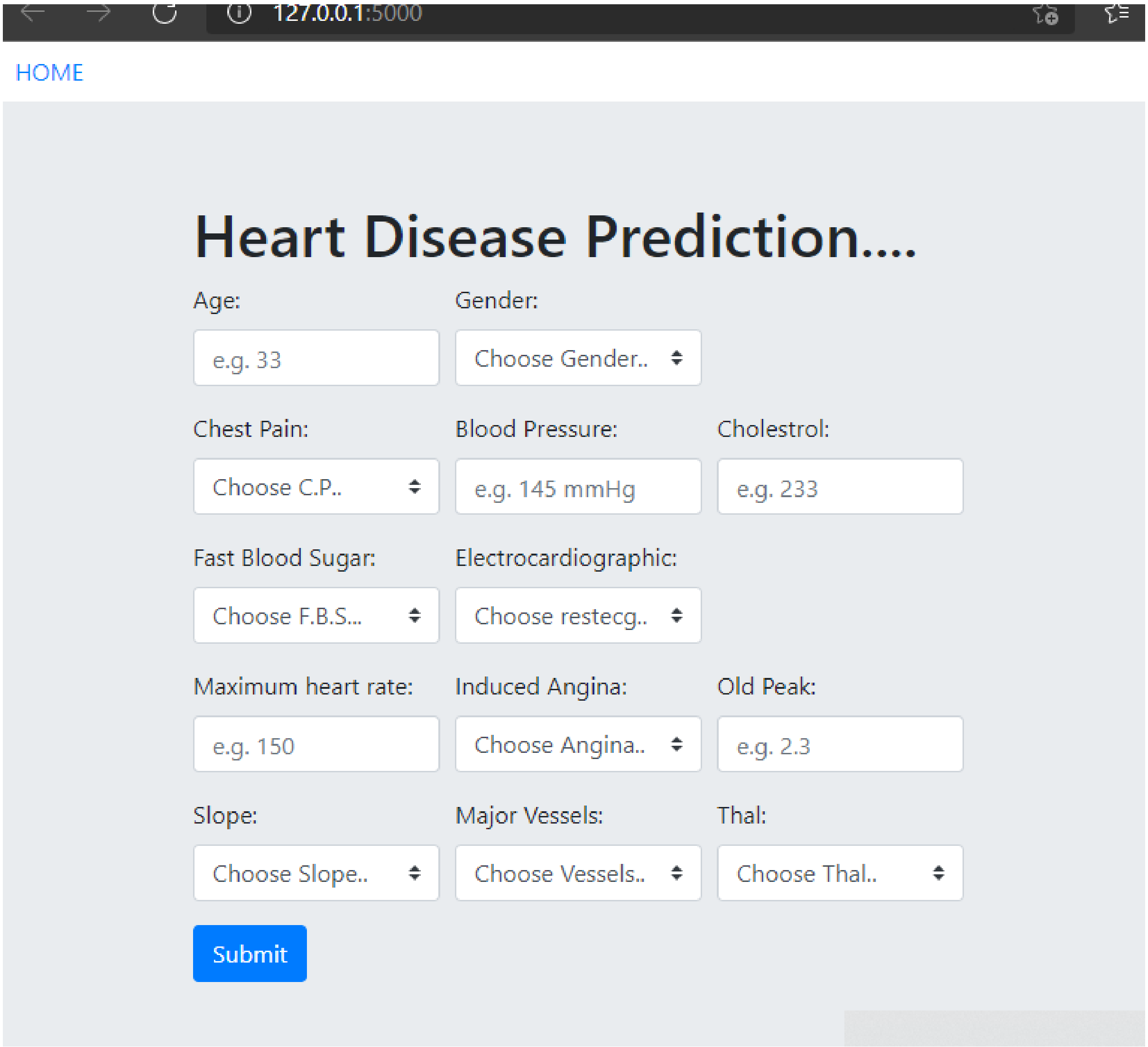
CHAPTER 7 : CONNECT TO WEB PAGE USING A PYTHON



```python
from flask import Flask, render_template, jsonify, request
import pickle
import numpy as np

model = pickle.load(open("D:\Project\my project\heart_disease.pkl", "rb"))

app = Flask(__name__)

@app.route('/')
def index():
    return render_template("home.html")


@app.route('/predict', methods = ['POST'])
def predict():


    final_feat = list(request.form.values())
    final_feat = [float(numeric_string) for numeric_string in final_feat]
    final_feat = np.array(final_feat).reshape(1,13)

    output = model.predict(final_feat)[0]

    return render_template("predict.html",output=output)


if __name__=='__main__':
    app.run(debug=True)
```

Figure 13 Connecting to web page

CHAPTER 8 : RESULT



127.0.0.1:5000

HOME

Heart Disease Prediction....

Age: e.g. 33

Gender: Choose Gender..

Chest Pain: Choose C.P..

Blood Pressure: e.g. 145 mmHg

Cholestrol: e.g. 233

Fast Blood Sugar: Choose F.B.S...

Electrocardiographic: Choose restecg..

Maximum heart rate: e.g. 150

Induced Angina: Choose Angina..

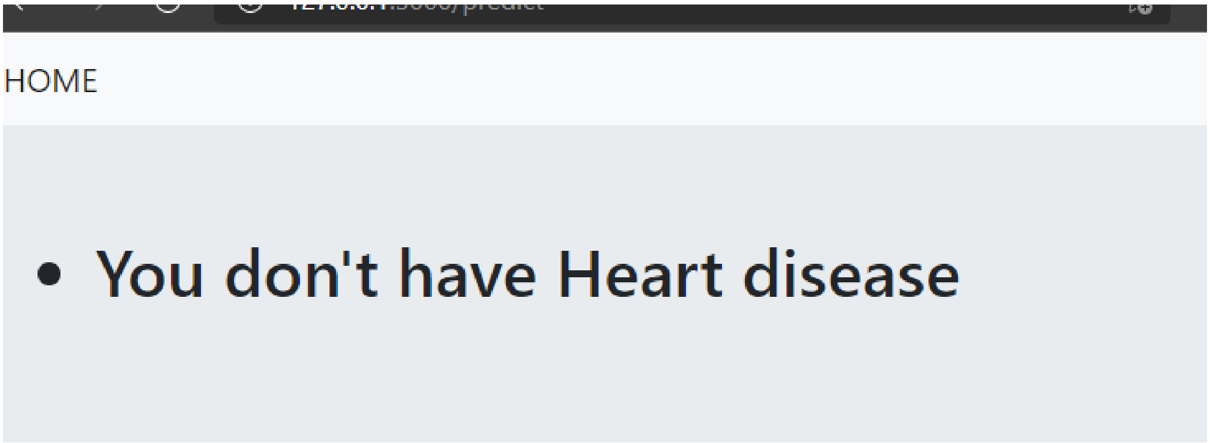
Old Peak: e.g. 2.3

Slope: Choose Slope..

Major Vessels: Choose Vessels..

Thal: Choose Thal..

Figure 14 ENTER YOUR DITAILS



HOME

- You don't have Heart disease

Figure 15 It is showing predition report

CHAPTER 9 : CODE

The coding portion were carried out to prepare the data, visualize it, pre-process it, building the model and then evaluating it. The code has been written in Python programming language using Jupyter Notebook as IDE. The experiments and all the models building are done based on python libraries. The code is available in the Git repository given in following link:

https://github.com/Rahul1-hy/Heart_disease_predition

9.1 Libraries used:

1. NumPy

2. Pandas

3. Matplotlib (pyplot, rcparams, matshow)

4. Pickle

5. flask

6.seabron

CHAPTER 10 : CONCLUSIONS

1. Out of the 13 features we examined, the top 4 significant features that helped us classify between a positive & negative Diagnosis were chest pain type (cp), maximum heart rate achieved (thalach), number of major vessels (ca), and ST depression induced by exercise relative to rest (oldpeak).

2. Our machine learning algorithm can now classify patients with Heart Disease. Now we can properly diagnose patients, & get them the help they needs to recover. By diagnosing detecting these features early, we may prevent worse symptoms from arising later.

3. Our Random Forest algorithm yields the highest accuracy, 86%. Any accuracy above 70% is considered good, but be careful because if your accuracy is extremely high, it may be too good to be true (an example of Over fitting). Thus, 80% is the ideal accuracy!

REFERENCES

[1] A. H. M. S. U. Marjia Sultana, "Analysis of Data Mining Techniques for Heart Disease Prediction," 2018.

[2] M. I. K. ,. A. I. ,. S. Musfiq Ali, "Heart Disease Prediction Using Machine Learning Algorithms".

[3] K. Bhanot, "towarddatascience.com," 13 Feb 2019. [Online]. Available: https://towardsdatascience.com/predicting-presence-of-heart-diseases-using-machinelearning-36f00f3edb2c.

[4] [Online]. Available: https://www.kaggle.com/ronitf/heart-disease-uci#heart.csv.

[5] M. A. K. S. H. K. M. a. V. P. M Marimuthu, "A Review on Heart Disease Prediction using Machine Learning and Data Analytics Approach".