A Project Report

on

“PREDICTING HEART DISEASE USING MACHING LEARNING”

Submitted in partial fulfillment of requirement for the award of the degree of

BACHELOR OF TECHNOLOGY

in

INFORMATION TECHNOLOGY

By:

Silky (100180572)
Sujal Sharma (100180554)

UNDER THE GUIDANCE OF

DR. LAXMI SHANKER SINGH
(Assistant Professor, Information Technology)

DEPARTMENT OF INFORMATION TECHNOLOGY

SIR CHHOTU RAM INSTITUTE OF ENGINEERING AND TECHNOLOGY
CHAUDHARY CHARAN SINGH UNIVERSITY, MEERUT

2018 - 2022

SIR CHHOTU RAM INSTITUTE OF ENGINEERING & TECHNOLOGY

CHAUDHARY CHARAN SINGH UNIVERSITY, MEERUT

Approved by A.I.C.T.E., New Delhi

Student Declaration/ Certificate-I

This is to certify that the project entitled PREDICTING HEART DISEASE USING MACHINE LEARNINIG in Meerut is submitted in partial fulfillment of the requirement of the degree of Bachelor of Technology Information Technology of **Sir Chhotu Ram Institute of Engineering and Technology, Chaudhary Charan Singh University Campus, Meerut (U.P.)** under the supervision.

(Project Supervisor) **(Project Co-Ordinator)**

SIR CHHOTU RAM INSTITUTE OF ENGINEERING & TECHNOLOGY

CHAUDHARY CHARAN SINGH UNIVERSITY, MEERUT

Approved by A.I.C.T.E., New Delhi

Student Declaration/ Certificate-II

The Project report entitled **PREDICTING HEART DISEASE USING MACHINE LEARNING** in Meerut is submitted by **SUJAL SHARMA (100180554), SILKY (100180572).** It warrants it's Acceptance as a prerequisite for the degree of Bachelor of Technology Information Technology of **Sir Chhotu Ram Institute of Engineering and Technology, Chaudhary Charan Singh University, Meerut (U.P.)** under the supervision**.**

(Internal Examiner) **(External Examiner)**

DR. MANAV BANSAL

(Co-Ordinator)

Dept. Of Information Technology

PROF. NIRAJ SINGHAL

(Director, SCRIET)

ACKNOWLEGEMENT

We extend our thanks to **PROF. NIRAJ SINGHAL** Director, Project Co-Ordinator **Er. ARPIT CHHABRA**, Project Supervisor **DR. LAXMI SHANKER SINGH** and the Head Of Department **DR. MANAV BANSAL** for giving us the motivation and guidance to complete this project.

We are deeply indebted to our project guide **PREDICTING HEART DISEASE USING MACHINE LEARNING** for the initial idea of the project and for all the guidance and encouragement he gave in the subsequent months. His supervision and co-operation at every stage helped us in successfully completing the project. Whatever intellectual effort may be reflected from this project is the direct result of the informative and stimulating discussions and suggestions he has given during the course of the project.

Lastly, we would like to thank the entire staff of the Information Technology Department and our college for their support and co-operation in the completion of this project.

(Students Signature)

ABSTRACT

Machine Learning is used across many ranges around the world. The healthcare industry is no exclusion. Machine Learning can play an essential role in predicting presence/absence of locomotors disorders, Heart diseases and more. Such information, if predicted well in advance, can provide important intuitions to doctors who can then adapt their diagnosis and dealing per patient basis. We work on predicting possible Heart Diseases in people using Machine Learning algorithms. In this project we perform the comparative analysis of classifiers like decision tree, NaïveBayes, Logistic Regression, SVM and Random Forest and we propose an ensemble classifier which perform hybrid classification by taking strong and weak classifiers since it can have multiple number of samples for training and validating the data so weperform the analysis of existing classifier and proposed classifier like Ada-boost and XG-boost which can give the better accuracy and predictive analysis..

Keywords: SVM; Naive Bayes; Decision Tree; Random Forest; Logistic Regression; Adaboost; XG-boost; python programming; confusion matrix; correlationmatrix.

CONTENTS

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

1.1Introduction

According to the World Health Organization, every year 12 million deaths occur worldwide due to Heart Disease. Heart disease is one of the biggest causes of morbidity and mortality among the population of the world. Prediction of cardiovascular disease is regarded as one of the most important subjects in the section of data analysis. 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 thedeath 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 reduces the complications.

Machine learning proves to be effective in assisting in making decisions and predictions from the large quantity of data produced by the health care industry. This project aims to predict future Heart Disease by analyzing data of patients whichclassifies whether they have heart disease or not using machine-learning algorithm. Machine Learning techniques can be a boon in this regard. Even though heart disease can occur in different forms, there is a common set of core risk factors that influence whether someone will ultimately be at risk for heart disease or not. By collecting the data from various sources, classifying them under suitable headings & finally analyzing to extract the desired data we can say that this technique can be very well adapted to do the prediction of heart disease.

1.2 Problem Statement

The major challenge in heart disease is its detection. There are instruments available which can predict heart disease but either it 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 everyday 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 analyze the data for hidden patterns. The hidden patterns can be used for health diagnosis in medicinal data.

1.3 Motivation for the Work

The main motivation of doing this research is to present a heart disease prediction model for the prediction of occurrence of heart disease. Further, this research work is aimed towards identifying the best classification algorithm for identifying the possibility of heart disease in a patient. This work is justified by performing a comparative study and analysis using three classification algorithms namely Naïve Bayes, Decision Tree, and Random Forest are used at different levels ofevaluations. Although these are commonly used machine learning algorithms, theheart disease prediction is a vital task involving highest possible accuracy. Hence, the three algorithms are evaluated at numerous levels and types of evaluation strategies. This will provide researchers and medical practitioners to establish a better.

1.4 Objective

The main objective of this project is to develop a heart disease prediction system. The heart disease prediction model should be trained using Machine learning algorithms to get efficient results. We want to provide a user-friendly environment where they can use general medical information and get the report if they are suffering from heart disease or not.

Other Aims of the model:

- Helps reducing errors
- To provide faster results
- To ensure accuracy in the prediction
- To reduce the cost of medical tests

1.5 Scope of the Project

The Heart disease prediction model can be used in various hospitals, medical clinics and in laboratories. The model promises to give more accurate results, decrease human biasness, eliminate costly tests and improve the overall human heart disease prediction outcome. It canhelp significantly in taking the quality decisions about their heart problems, not only for doctors as well as for patients. Since the system is user friendly, no specific qualifications arerequired by the users.

1.6 Need of Work

Heart disease is a major concern that needs to be dealt with more efficiency and accuracy. According to a study published in health journal The Lancet in the September 2018, Deaths due to cardiovascular diseases in India increased from 1.3 million in 1990 to 2.8 million in 2016, and more than half the deaths caused by heart ailments in 2016 were in persons less than 70 years of age. These heartbreaking numbers needs to get reduced. However, due to certain constraints, it is difficult to come to come to a conclusion whether a person is suffering from a heart disease or not. Thankfully, machine learning has proved to be great approach in determining the disease using large quantity of data produced by healthcare industries. In this project, we will be using different machine learning algorithms to train our model. After comparison, on the basis of accuracy, our final model will be decided.

CHAPTER 2
LITERATURE SURVEY

With growing development in the field of medical science alongside machine learning various experiments and researches has been carried out in these recent years releasing the relevant significant papers.

1. Purushottam ,et ,al proposed a paper “Efficient Heart Disease Prediction System” using hill climbing and decision tree algorithms .They used Cleveland dataset and preprocessing of data is performed before using classification algorithms. The Knowledge Extraction is done based on Evolutionary Learning (KEEL), an open-source data mining tool that fills the missing values in the data set.A decision tree follows top-down order. For each actual node selected by hill-climbing algorithm a node is selected by a test at each level. The parameters and their values used are confidence. Its minimum confidence value is 0.25. The accuracy of the system is about 86.7%.

2. Santhana Krishnan. J ,et ,al proposed a paper “Prediction of Heart Disease Using Machine Learning Algorithms" using decision tree and Naive Bayes algorithm for prediction of heart disease. In decision tree algorithm the tree is built using certain conditions which gives True or False decisions. The algorithms like SVM, KNN are results based on vertical or horizontal split conditions depends on dependent variables. But decision tree for a tree like structure having root node, leaves and branches base on the decision made in each of tree Decision tree also help in the understating the importance of the attributes in the dataset. They have also used Cleveland data set. Dataset splits in 70% training and 30% testing by using some methods. This algorithm gives 91% accuracy. The second algorithm is Naive Bayes, which is used for classification. It can handle complicated, nonlinear, dependent data so it is found suitable for heart disease dataset as this dataset is also complicated, dependent and nonlinear in nature. This algorithm gives an 87% accuracy.

3. Sonam Nikhar et al proposed paper " Prediction of Heart Disease Using Machine Learning Algorithms" their research gives point to point explanation of Naïve Bayes and decision tree classifier that are used especially in the prediction of heart disease. Some analysis has been led to think about the execution of prescient data mining strategy on the same dataset, and the result decided that Decision Tree has highest accuracy than Bayesian classifier.

4. Aditi Gavhane et al proposed a paper "Prediction of Heart Disease Using Machine Learning", in which training and testing of dataset is performed by using neuralnetwork algorithm multi-layer perceptron. In this algorithm there will be one input layer and one output layer and one or more layers are hidden layers between these two input and output layers. Through hidden layers each input node is connected to output layer. This connection is assigned with some random weights. The other input is calledbias which is assigned with weight based on requirement the connection between the nodes can be feedforwarded or feedback.

5. Avinash Golande et al, proposed "Heart Disease Prediction Using Effective Machine Learning Techniques" in which few data mining techniques are used that support the doctors to differentiate the heart disease. Usually utilized methodologies are k-nearest neighbour, Decision tree and Naïve Bayes. Other unique characterization-based strategies utilized are packing calculation, Part thickness, consecutive negligible streamlining and neural systems, straight Kernel selfarranging guide and SVM (Bolster Vector Machine).

6. Lakshmana Rao et al,proposed "Machine Learning Techniques for Heart Disease Prediction" in which the contributing elements for heart disease are more. So, it is difficult to distinguish heart disease.To find the seriousness of the heart disease among people different neural systems and data mining techniques are used.

7. Abhay Kishore et alproposed "Heart Attack Prediction Using Deep Learning" in which heart attack prediction system by using Deep learning techniques and to predictthe probable aspects of heart related infections of the patient Recurrent NeuralSystem is used. This model uses deep learning and data mining to give the best precisemodel and least blunders. This paper acts as strong reference model for another

type ofheart attack prediction models

8. Senthil Kumar Mohan et al, proposed “Effective Heart Disease Prediction Using Hybrid Machine Learning Techniques” in which their main objective is to improve exactness in cardiovascular problems. The algorithms used are KNN, LR, SVM, NN to produce an improved exhibition level with a precision level of 88.7% through the prediction model for heart disease with hybrid random forest with linear model(HRFLM).

9. Anjan N. Repaka et al, proposed a model stated the performance of prediction for two classification models, which is analyzed and compared to previous work. The experimental results show that accuracy is improved in finding the percentage of risk prediction of our proposed method in comparison with other models.

10. Aakash Chauhan et al, proposed “Heart Disease Prediction using Evolutionary Rule Learning”. Data is directly retrieved from electronic records that reduce the manual tasks. The amount of services are decreased and shown major number of rules helps within the best prediction of heart disease. Frequent pattern growth association mining is performed on patient’s dataset to generate strong association.

CHAPTER 3
METHODOLOGY

3.1 Existing System

Heart disease is even being highlighted as a silent killer which leads to the death of a person without obvious symptoms. The nature of the disease is the cause of growing anxiety about the disease & its consequences. Hence continued efforts are being done to predict the possibility of this deadly disease in prior. So that various tools & techniques are regularly being experimented with to suit the present-day healthneeds. Machine Learning techniques can be a boon in this regard. Even though heart disease can occur in different forms, there is a common set of core risk factors that influence whether someone will ultimately be at risk for heart disease or not. By collecting the data from various sources, classifying them under suitable headings & finally analysing to extract the desired data we can conclude. This technique can be very well adapted to the do the prediction of heart disease. As the well-known quote says "Prevention is better than cure", early prediction & its control can be helpful to prevent & decrease the death rates due to heart disease.

3.2 Proposed System

The working of the system starts with the collection of data and selecting the important attributes. Then the required data is preprocessed into the required format. The data is then divided into two parts training and testing data. The algorithms are applied and the model is trained using the training data. The accuracy of the system isobtained by testing the system using the testing data. This system is implemented using the following modules.

1.) Collection of Dataset
2.) Selection of attributes
3.) Data Pre-Processing
4.) Balancing of Data
5.) Disease Prediction

3.2.1 Collection of dataset

Initially, we collect a dataset for our heart disease prediction system. After the collection of the dataset, we split the dataset into training data and testing data. The training dataset is used for prediction model learning and testing data is used for evaluating the prediction model. For this project, 70% of training data is used and 30%of data is used for testing. The dataset used for this project is Heart Disease UCI. The dataset consists of 76 attributes; out of which, 14 attributes are used for the system.

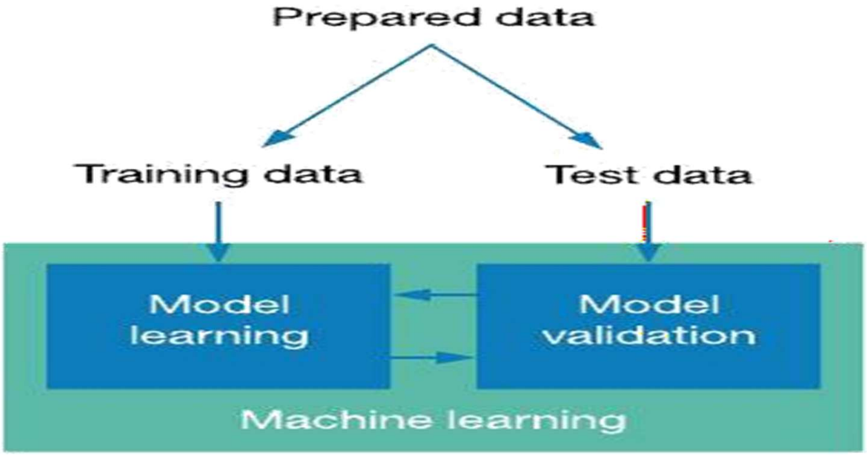
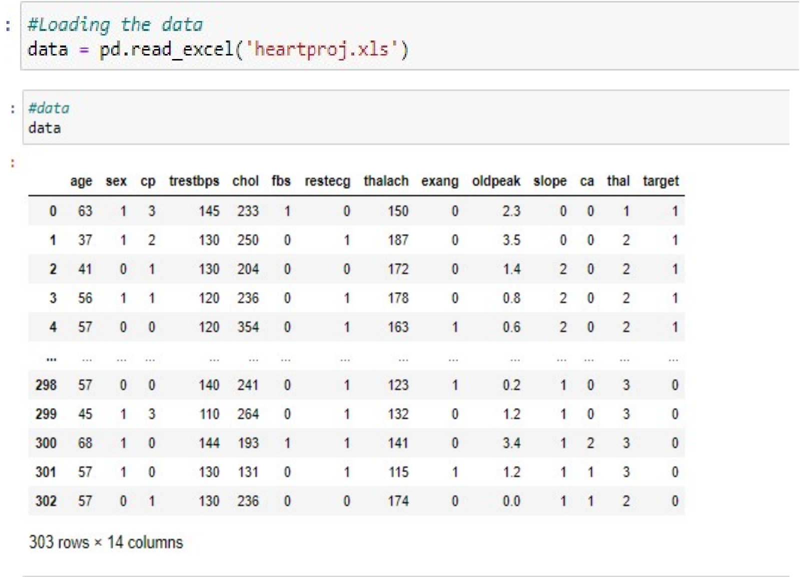


Figure: Collection of Data

3.2.1.1 Loading and viewing the dataset: We have used pandas and numpy libraries of Python for loading, viewing and for pre-processing work on the dataset. We loadedthe dataset into notebook and then viewed the attributes of the dataset. The dataset consists of 303 row and 14 columns.



```
#Loading the data
data = pd.read_excel('heartproj.xls')
```

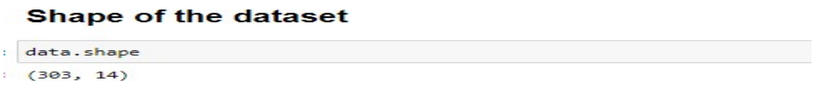
```
#data
data
```

	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

Fig- Loading and viewing dataset

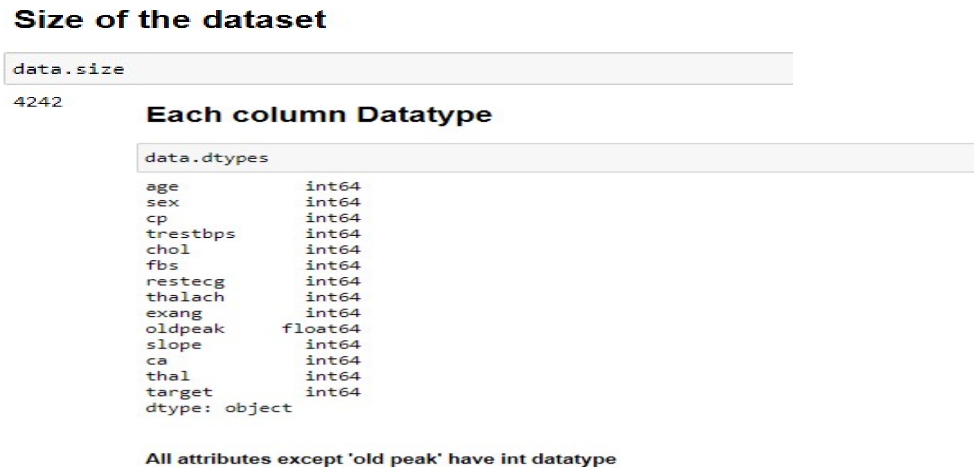
3.2.1.2 Data-type and data info- To view the data-type we used command dtypes. Also,to describe and get the information of the dataset we used df.info() and df.describe().



Shape of the dataset

```
data.shape
```

```
(303, 14)
```



Size of the dataset

```
data.size
```

```
4242
```

Each column Datatype

```
data.dtypes
```

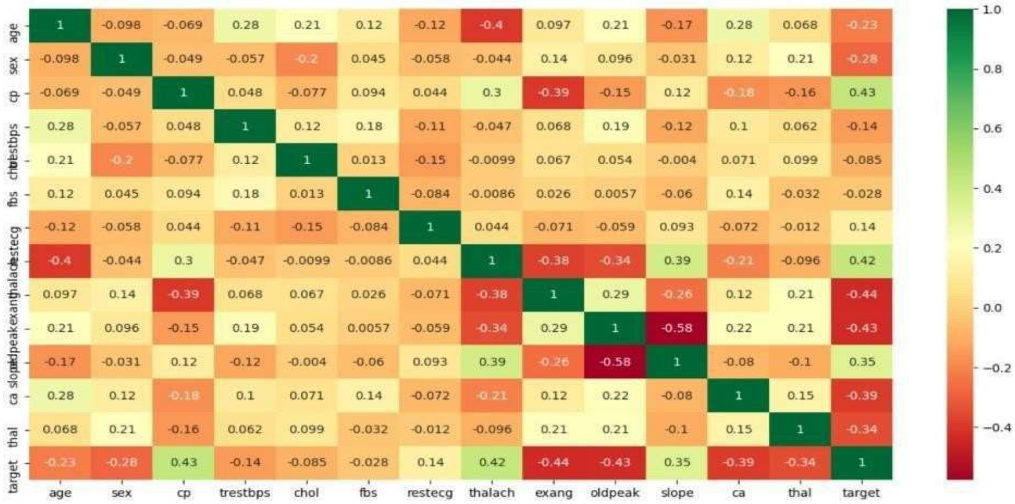
```
age            int64
sex            int64
cp             int64
trestbps       int64
chol           int64
fbs            int64
restecg        int64
thalach        int64
exang          int64
oldpeak      float64
slope          int64
ca             int64
thal           int64
target         int64
dtype: object
```

All attributes except 'old peak' have int datatype

Fig 3.3- Column datatype

3.2.2 Selection of Attributes

Attribute or Feature selection includes the selection of appropriate attributes forthe prediction system. This is used to increase the efficiency of the system. Various attributes of the patient like gender, chest pain type, fasting blood pressure, serum cholesterol, exang, etc are selected for the prediction. The Correlation matrix is used for attribute selection for this model.



	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: Correlation matrix

3.2.3 Pre-processing of Data

Data pre-processing is an important step for the creation of a machine learning model. Initially, data may not be clean or in the required format for the model which can cause misleading outcomes. In pre-processing of data, we transform data into our required format. It is used to deal with noises, duplicates, and missing values of the dataset. Data pre-processing has the activities like importing datasets, splitting datasets, attribute scaling, etc. Preprocessing of data is required for improving the accuracy of the model.

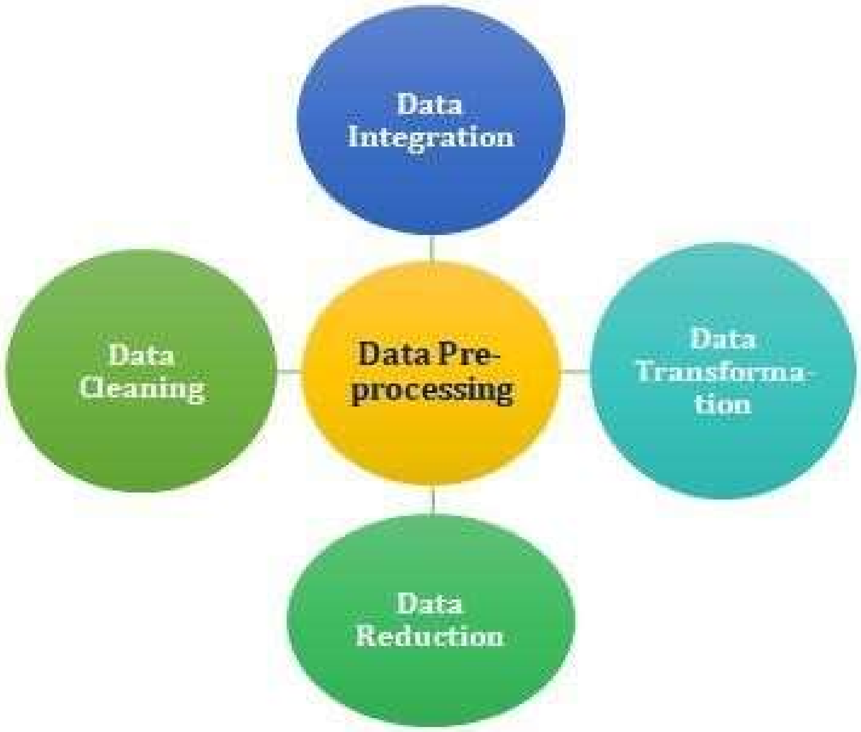
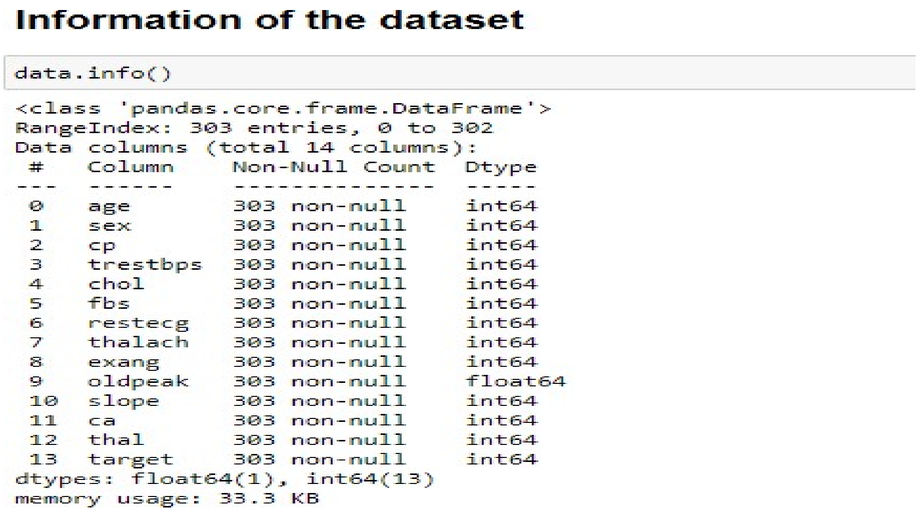


Figure: Data Pre-processing

3.2.3.1 Data pre-processing: After viewing the data-type and basic information of thedataset data pre-processing is required. Data is said to be unclean if it has missing attributes, attribute values, contains noise or outliers, duplicate values.

For data pre-processing we started with handling missing values. We found that there were no missing values in our dataset. Then we looked for the duplicate values and found one duplicate row. We dropped the duplicate row. Now our dataset contains 302rows and 14 columns. The categorical data was already numeric encoded and all the data types were compatible with each other. So data formatting was not required.



Information of the dataset

```
data.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
```

Fig Handling missing values

3.2.4 Balancing of Data

Imbalanced datasets can be balanced in two ways. They are Under Samplingand Over Sampling

(a) Under Sampling:

In Under Sampling, dataset balance is done by the reduction of the size of the ampleclass. This process is considered when the amount of data is adequate.

(b) Over Sampling:

In Over Sampling, dataset balance is done by increasing the size of the scarce samples.This process is considered when the amount of data is inadequate.

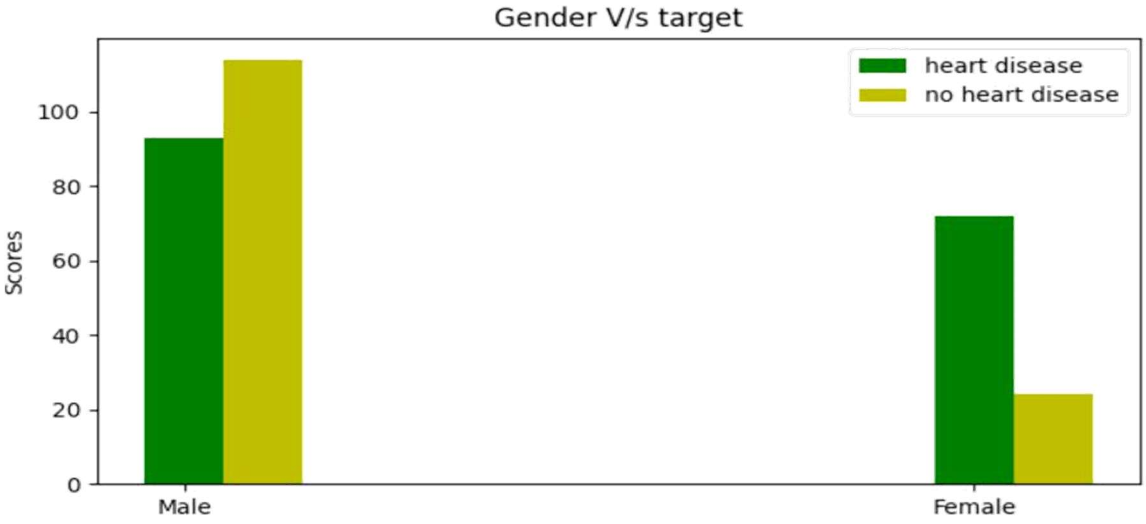


Figure: Data Balancing

3.2.5 Prediction of Disease

Various machine learning algorithms like SVM, Naive Bayes, Decision Tree, Random Tree, Logistic Regression, Ada-boost, Xg-boost are used for classification. Comparative analysis is performed among algorithms and the algorithm that gives the highest accuracy is used for heart disease prediction.

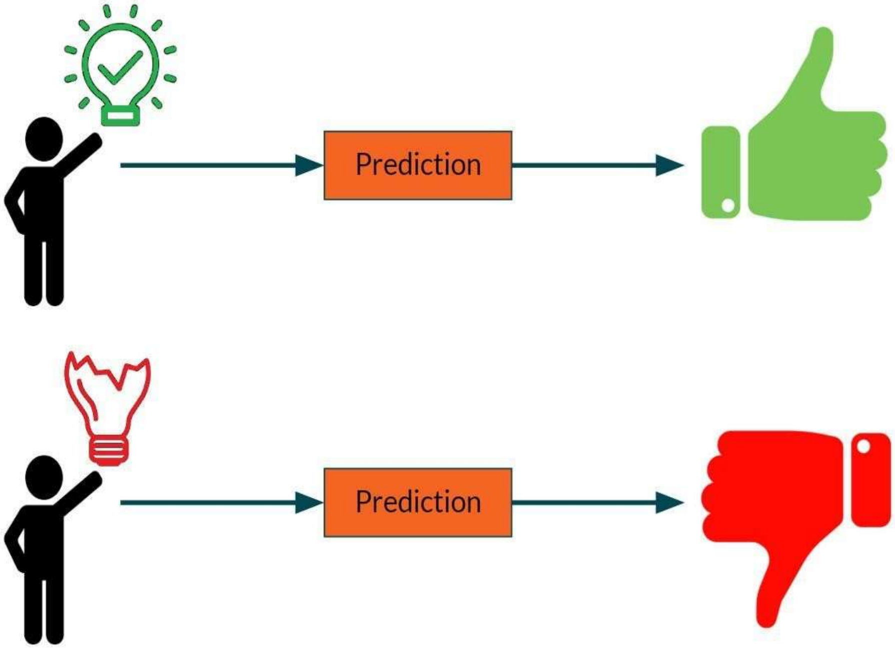
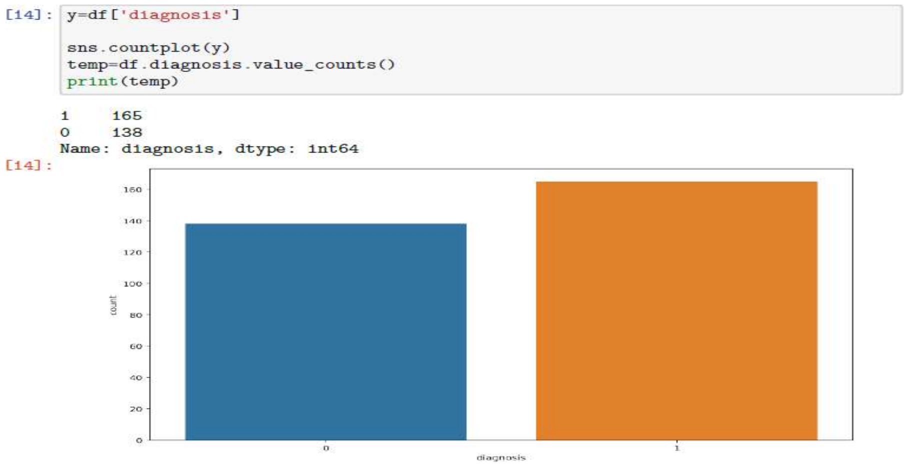


Figure: Prediction of Disease

3.2.6 Data Visualization- After data pre-processing we visualized our dataset. Data visualizationis required because it provides the visual summary of the information and makes it easier to understand the patterns and trends in the attributes. Data visualization is done to gain insights from the dataset. It helps to understand the relation between the attributes and how it is correlated to the target variable.

- **Countplot-** After importing the required libraries of Python, i.e, Pandas, matplotlib, and seaborn. We started with plotting the countplot of the target variable. It showed that the dataset contains 165 values with 1 and 130 values as 0. This shows that the person suffering from heart disease is more.



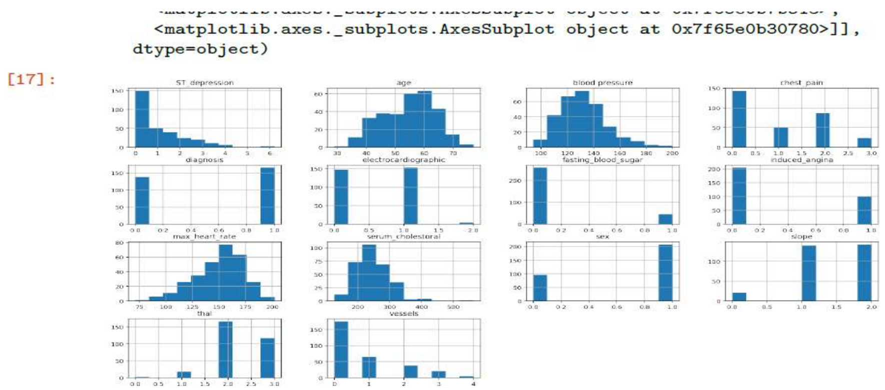
```
[14]: y=df['diagnosis']

      sns.countplot(y)
      temp=df.diagnosis.value_counts()
      print(temp)

      1    165
      0    138
      Name: diagnosis, dtype: int64
[14]:
```

Fig Count plot of target variable

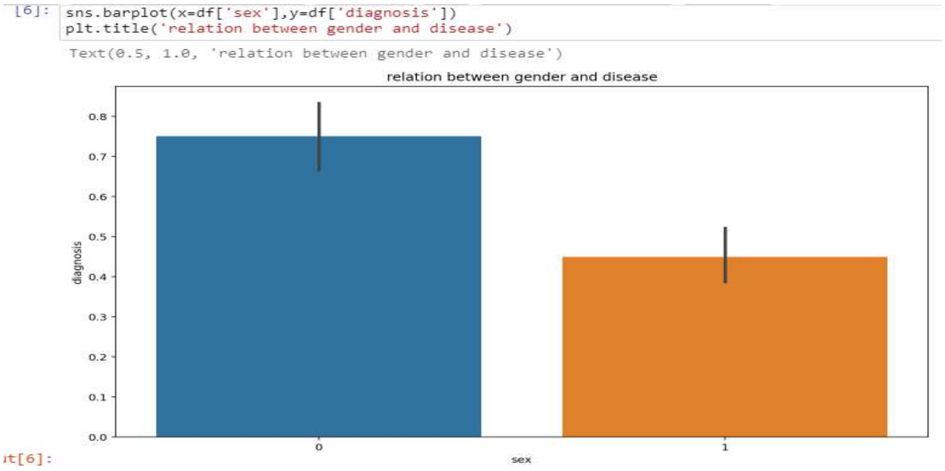
- **Univariate graph**- We plotted the univariate graph for all attributes using df.hist().This plotted the frequency of each value for every column in the dataset.



```
  <matplotlib.axes._subplots.AxesSubplot object at 0x7f65e0b30780>]],
 dtype=object)
[17]:
```

Fig- Univariate graph showing frequency of each value

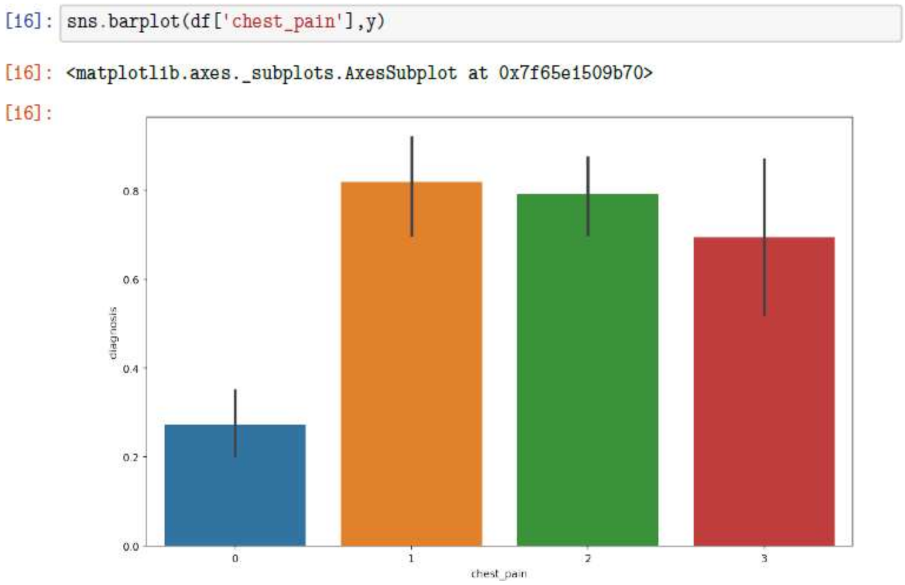
- **Barplot**- The barplot represents categorical data with rectangular bars with heights proportional to the values that they present. The barplot between gender and target variable showed that the females are suffering from heart disease more than men. Thebarplot between chest pain and the target variable showed that the person with chest pain value of 1 and 2 are the one suffering most from the heart disease compared to other values.



```
[6]: sns.barplot(x=df['sex'],y=df['diagnosis'])
     plt.title('relation between gender and disease')

     Text(0.5, 1.0, 'relation between gender and disease')
```

Fig- Barplot between gender and target

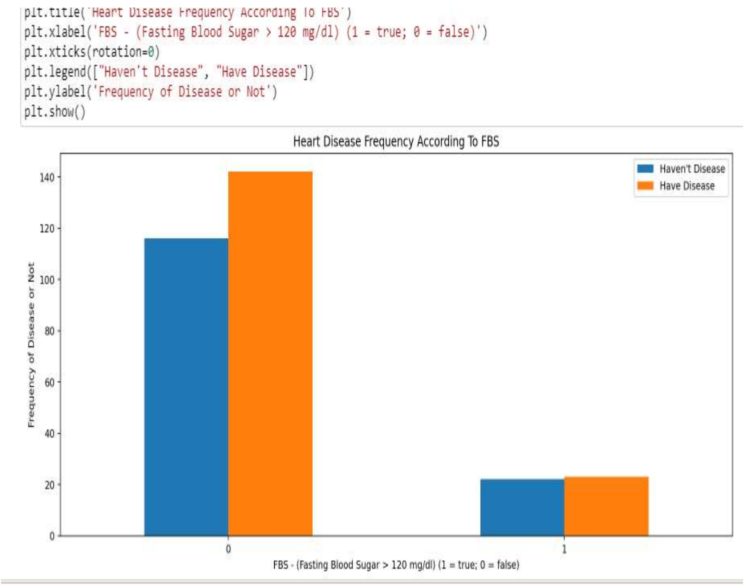


```
[16]: sns.barplot(df['chest_pain'],y)

[16]: <matplotlib.axes._subplots.AxesSubplot at 0x7f65e1509b70>
[16]:
```

Fig- Barplot between chest pain and target

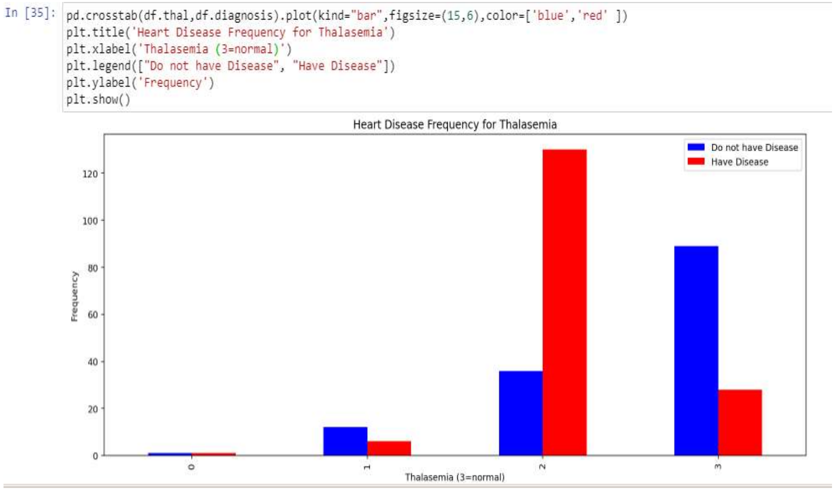
- **Crosstab**: The crosstab function in pandas is used to compute a simple cross tabulationof two or more factors. We plotted crosstab for age and target variable. It showed that the people between ages 41 to 55 are found to suffer more from heart disease than any other age group. The crosstab between fasting blood sugar and target variable showed that the person with fasting blood sugar value less than 120mg/dl are found to suffer more from heart disease. The crosstab between thalasemia and target variable showed that the thal value of 2 are found to suffer from heart disease.



```
plt.title('Heart Disease Frequency According To FBS')
plt.xlabel('FBS - (Fasting Blood Sugar > 120 mg/dl) (1 = true; 0 = false)')
plt.xticks(rotation=0)
plt.legend(["Haven't Disease", "Have Disease"])
plt.ylabel('Frequency of Disease or Not')
plt.show()
```

Heart Disease Frequency According To FBS
Haven't Disease
Have Disease
Frequency of Disease or Not
0
20
40
60
80
100
120
140
0
1
FBS - (Fasting Blood Sugar > 120 mg/dl) (1 = true; 0 = false)

Fig: Crosstab of fasting blood sugar

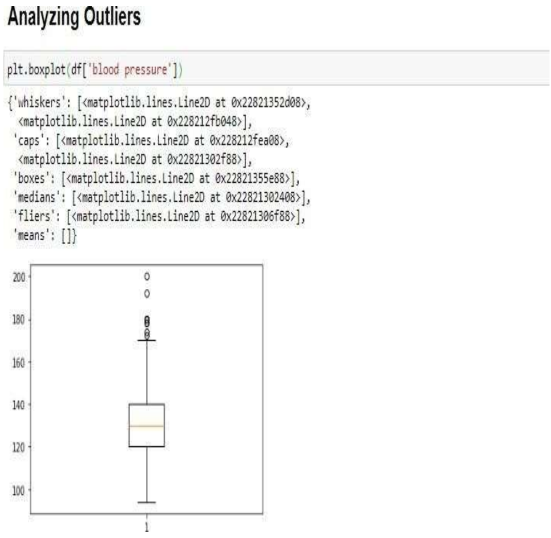


```
In [35]: pd.crosstab(df.thal,df.diagnosis).plot(kind="bar",figsize=(15,6),color=['blue','red' ])
         plt.title('Heart Disease Frequency for Thalasemia')
         plt.xlabel('Thalasemia (3=normal)')
         plt.legend(["Do not have Disease", "Have Disease"])
         plt.ylabel('Frequency')
         plt.show()
```

Fig

Fig- Crosstab for thalassemia

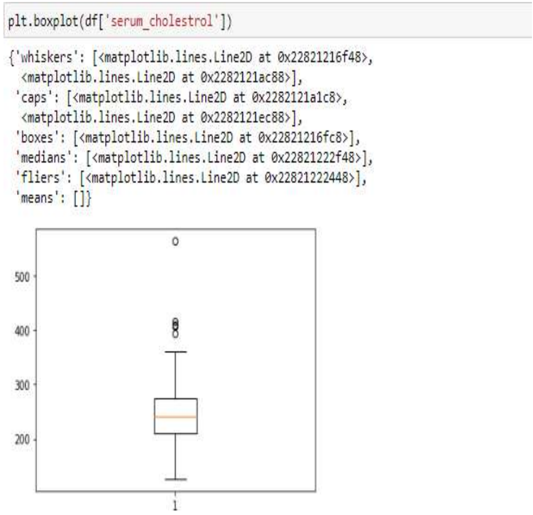
- **Outliers**: The outliers can be mistake during data collection. So it is important to remove outliers else they can affect our model fitting and result in poor prediction. Thebest way to identify outliers is to use boxplot. The plotting of boxlpot for all attributes showed that only two attributes blood pressure and cholesterol are having outliers. We removed outliers using standard deviation method.



Analyzing Outliers

```
plt.boxplot(df['blood pressure'])
```

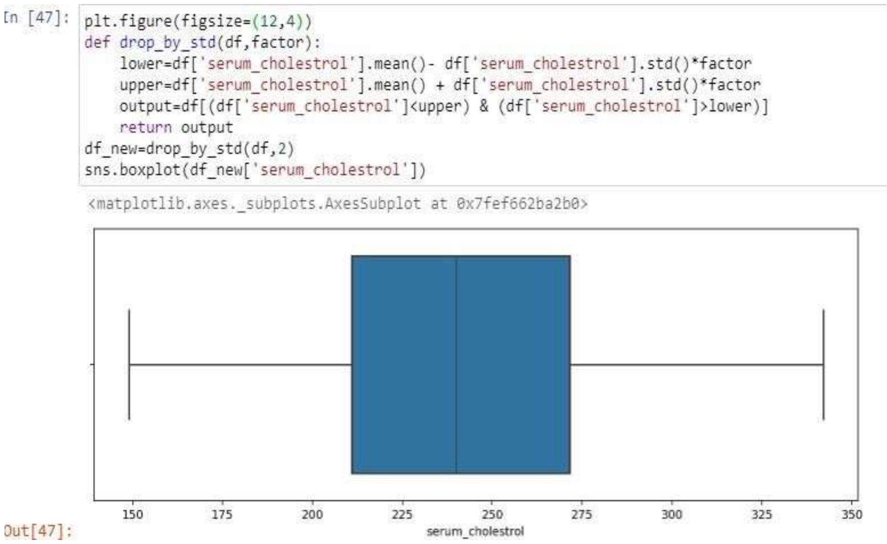
```
{'whiskers': [<matplotlib.lines.Line2D at 0x22821352d08>,
  <matplotlib.lines.Line2D at 0x228212fb048>],
 'caps': [<matplotlib.lines.Line2D at 0x228212fea08>,
  <matplotlib.lines.Line2D at 0x22821302f88>],
 'boxes': [<matplotlib.lines.Line2D at 0x22821355e88>],
 'medians': [<matplotlib.lines.Line2D at 0x22821302408>],
 'fliers': [<matplotlib.lines.Line2D at 0x22821306f88>],
 'means': []}
```



```
plt.boxplot(df['serum_cholestrol'])
```

```
{'whiskers': [<matplotlib.lines.Line2D at 0x22821216f48>,
  <matplotlib.lines.Line2D at 0x2282121ac88>],
 'caps': [<matplotlib.lines.Line2D at 0x2282121a1c8>,
  <matplotlib.lines.Line2D at 0x2282121ec88>],
 'boxes': [<matplotlib.lines.Line2D at 0x22821216fc8>],
 'medians': [<matplotlib.lines.Line2D at 0x22821222f48>],
 'fliers': [<matplotlib.lines.Line2D at 0x22821222448>],
 'means': []}
```

Fig- Identifying outlier



```
In [47]: plt.figure(figsize=(12,4))
         def drop_by_std(df,factor):
             lower=df['serum_cholestrol'].mean()- df['serum_cholestrol'].std()*factor
             upper=df['serum_cholestrol'].mean() + df['serum_cholestrol'].std()*factor
             output=df[(df['serum_cholestrol']<upper) & (df['serum_cholestrol']>lower)]
             return output
         df_new=drop_by_std(df,2)
         sns.boxplot(df_new['serum_cholestrol'])

Out[47]: <matplotlib.axes._subplots.AxesSubplot at 0x7fef662ba2b0>
```

Fig- Removing outliers

- **Correlation matrix**: A quick way to check the correlation between the columns is correlation matrix plot. We have plotted the correlation matrix. Here the dark coloredblue shows that the attributes are highly correlated where as the lightest color that is yellow shows negative correlation. The correlation matrix showed that none of the columns are highly correlated with each other. Though some of the attributes are correlated with each other. The attributes that shows positive correlation with each other are thalasemia and blood pressure, cholesterol and blood pressure, fasting bloodsugar and blood pressure. Some attributes shows negative correlation also.

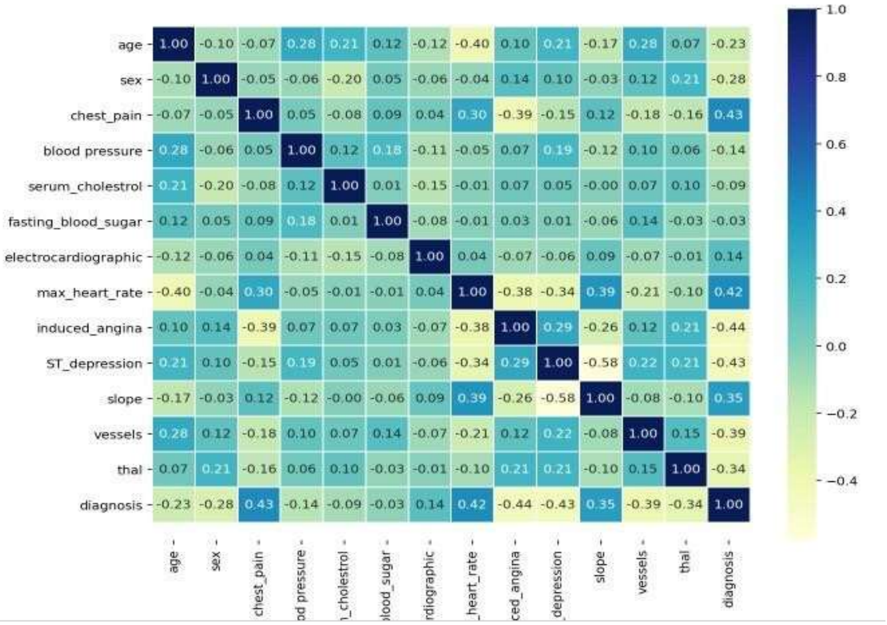


Fig: Correlation matrix

CHAPTER 4
WORKING OF SYSTEM

4.1 System Architecture

The system architecture gives an overview of the working of the system.

The working of this system is described as follows:

Dataset collection is collecting data which contains patient details. Attributes selection process selects the useful attributes for the prediction of heart disease. After identifying the available data resources, they are further selected, cleaned, made into the desired form. Different classification techniques as stated will be applied on preprocessed data to predict the accuracy of heart disease. Accuracy measure compares the accuracy of different classifiers.

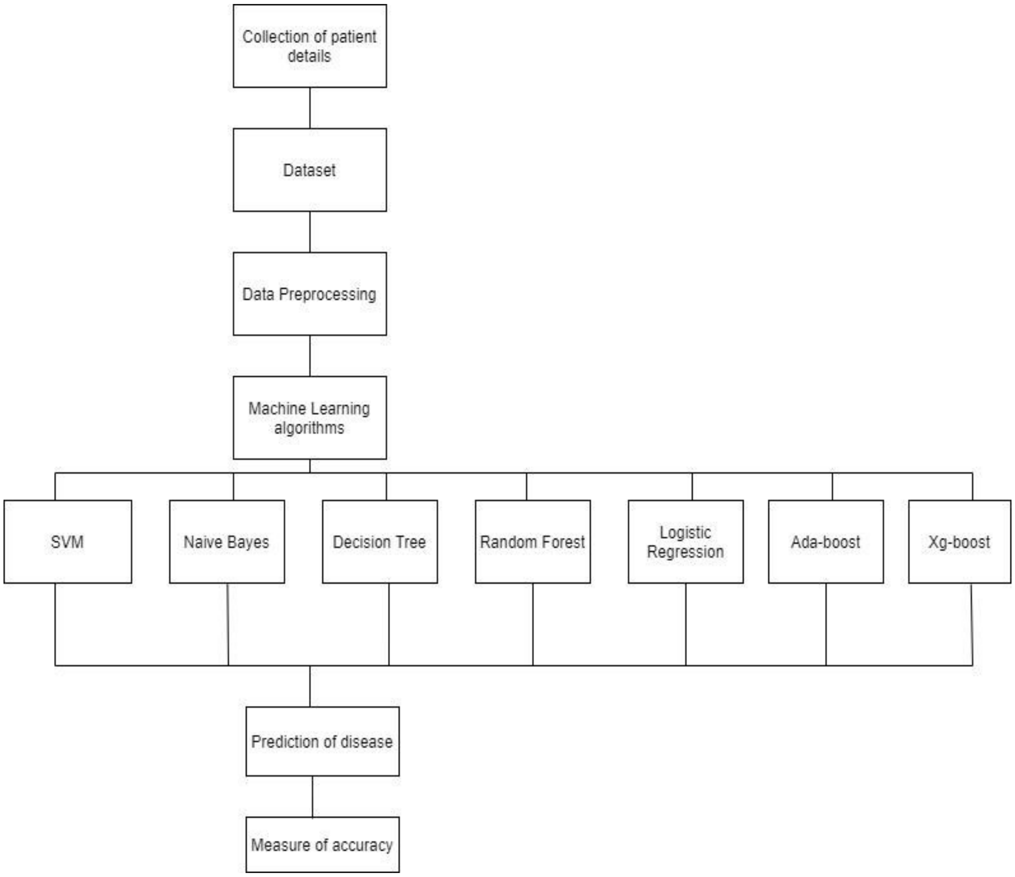


Figure: System Architecture

4.2 Machine Learning

In machine learning, classification refers to a predictive modeling problemwhere a class label is predicted for a given example of input data.

- **Supervised Learning**

  Supervised learning is the type of machine learning in which machines are trained using well "labelled" training data, and on the basis of that data, machines predict the output. The labelled data means some input data is already tagged with the correct output.

  In supervised learning, the training data provided to the machines work as the supervisor that teaches the machines to predict the output correctly. It applies the sameconcept as a student learns in the supervision of the teacher.

  Supervised learning is a process of providing input data as well as correct output data to the machine learning model. The aim of a supervised learning algorithm is to find a mapping function to map the input variable(x) with the output variable(y).

- **Unsupervised learning**

  Unsupervised learning cannot be directly applied to a regression or classification problem because unlike supervised learning, we have the input data but no corresponding output data. The goal of unsupervised learning is to find the underlying structure of dataset, group that data according to similarities, and represent that dataset in a compressed format.

  - Unsupervised learning is helpful for finding useful insights from the data.
  - Unsupervised learning is much similar to how a human learns to think by their own experiences, which makes it closer to the real AI.
  - Unsupervised learning works on unlabeled and uncategorized data which make unsupervised learning more important.
  - In real-world, we do not always have input data with the corresponding output so to solve such cases, we need unsupervised learning.

- **Reinforcement learning**

  Reinforcement learning is an area of Machine Learning. It is about taking suitable action to maximize reward in a particular situation. It is employed by various software and machines to find the best possible behaviour or path it should take in a specific situation. Reinforcement learning differs from supervised learning in a way that in supervised learning the training data has the answer key with it so the model is

trained with the correct answer itself whereas in reinforcement learning, there is no answer but the reinforcement agent decides what to do to perform the given task. In the absence of a training dataset, it is bound to learn from its experience.

4.3 Algorithms

4.3.1 Support Vector Machine (SVM):

Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems. However, primarily, it is used for Classification problems in Machine Learning.

The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data pointin the correct category in the future. This best decision boundary is called a hyperplane.SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called support vectors, and hence the algorithm istermed as Support Vector Machine.

Support vector machines (SVMs) are powerful yet flexible supervised machine learning algorithms which are used both for classification and regression. But generally, they are used in classification problems. In the 1960s, SVMs were first introduced but later they got refined in 1990. SVMs have their unique way of implementation as compared to other machine learning algorithms. Lately, they are extremely popular because of their ability to handle multiple continuous and categorical variables.
The followings are important concepts in SVM -

Support Vectors - Data Points that are closest to the hyperplane are called support vectors. Separating line will be defined with the help of these data points.

Hyperplane - As we can see in the above diagram, it is a decision plane or spacewhich is divided between a set of objects having different classes.

Margin - It may be defined as the gap between two lines on the closest data points of different classes. It can be calculated as the perpendicular distance from the line to the support vectors. Large margin is considered as a good margin and small margin is considered as a bad margin.

Types of SVM:

SVM can be of two types:

- **Linear SVM:** Linear SVM is used for linearly separable data, which means ifa dataset can be classified into two classes by using a single straight line, then such data is termed as linearly separable data, and classifier is used called as Linear SVM classifier.
- **Non-linear SVM:** Non-Linear SVM is used for non-linearly separated data, which means if a dataset cannot be classified by using a straight line, then such data is termed as non-linear data and classifier used is called as Non-linear SVM classifier.

The objective of the support vector machine algorithm is to find a hyperplane in an N-dimensional space (N - the number of features) that distinctly classifies the data points.

The advantages of support vector machines are:

- Effective in high dimensional spaces.
- Still effective in cases where the number of dimensions is greater than the number of samples.
- Uses a subset of training points in the decision function (called support vectors), so it is also memory efficient.
- Versatile: different kernel functions can be specified for the decision function. Common kernels are provided, but it is also possible to specify custom kernels.

The disadvantages of support vector machines include:

- If the number of features is much greater than the number of samples, avoid over-fitting in choosing Kernel functions and regularization term is crucial.

SVMs do not directly provide probability estimates, these are calculated using an expensive five-fold cross-validation.

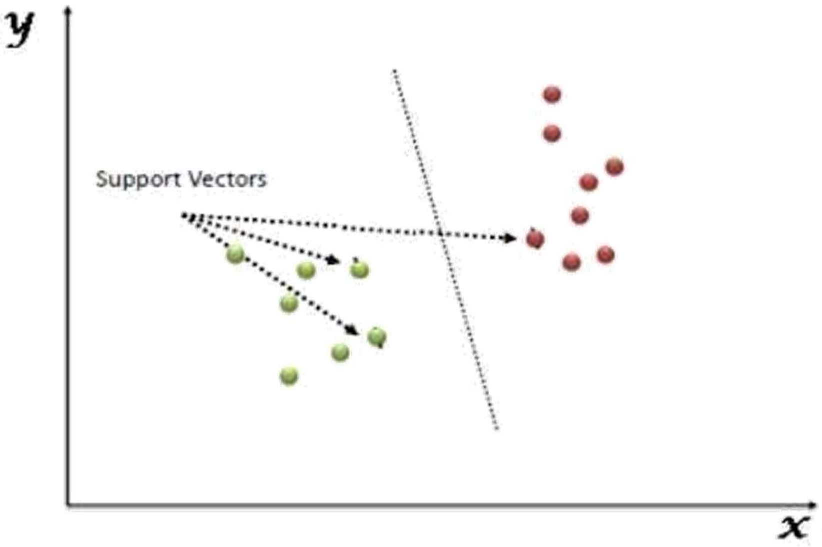


Figure: Support Vector Machine

4.3.2 Naive Bayes Algorithm:

Naive Bayes algorithm is a supervised learning algorithm, which is based on Bayes theorem and used for solving classification problems. It is mainly used in text classification that includes a high-dimensional training dataset.

Naive Bayes Classifier is one of the simple and most effective Classification algorithms which helps in building the fast machine learning models that can make quick predictions.

It is a probabilistic classifier, which means it predicts on the basis of the probability of an object. Some popular examples of Naïve Bayes Algorithm are spam filtration, Sentimental analysis, and classifying articles.

It is a classification technique based on Bayes' Theorem with an assumption ofindependence among predictors. In simple terms, a Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature.

The Naive Bayes model is easy to build and particularly useful for very large data sets. Along with simplicity, Naive Bayes is known to outperform even highly sophisticated classification methods.

The Naive Bayes algorithm is comprised of two words Naive and Bayes, Which can be described as:

- **Naive:** It is called Naive because it assumes that the occurrence of a certain feature is independent of the occurrence of other features. Such as if the fruit is identified on the basis of color, shape, and taste, then red, spherical, and sweet fruit is recognized as an apple. Hence each feature individually contributes to identify that it isan apple without depending on each other.
- **Bayes:** It is called Bayes because it depends on the principle of Bayes' Theorem.

Bayes’ theorem:

Bayes' theorem is also known as Bayes' Rule or Bayes' law, which is used to determine the probability of a hypothesis with prior knowledge. It depends on the conditional probability.

The formula for Bayes' theorem is given as:

$$\mathrm{P(A|B)}= \frac{P(B|A)P(A)}{P(B)}$$

Where,

P(A|B) is Posterior probability: Probability of hypothesis A on the observed event B.

P(B|A) is Likelihood probability: Probability of the evidence given that the probabilityof a hypothesis is true.

P(A) is Prior Probability: Probability of hypothesis before observing the evidence.P(B) is

Marginal Probability: Probability of Evidence.

Types of Naive Bayes model:

There are three types of Naive Bayes Model, which are given below:

- **Gaussian:** The Gaussian model assumes that features follow a normal distribution. This means if predictors take continuous values instead of discrete, then the model assumes that these values are sampled from the Gaussian distribution.
- **Multinomial:** The Multinomial Naïve Bayes classifier is used when the data is multinomial distributed. It is primarily used for document classification problems, it means a particular document belongs to which category such as

Sports, Politics, education, etc. The classifier uses the frequency of words for the predictors.

- **Bernoulli:** The Bernoulli classifier works similar to the Multinomial classifier, but the predictor variables are the independent Booleans variables. Such as if a particular word is present or not in a document. This model is also famous for document classification tasks.

4.3.3 Decision Tree Algorithm

Decision Tree is a Supervised learning technique that can be used for both classification and regression problems, but mostly it is preferred for solving classification problems. It is a tree-structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome. In a Decision Tree, there are two nodes, which are the Decision Node and Leaf Node.

Decision nodes are used to make any decision and have multiple branches, whereas Leaf nodes are the output of those decisions and do not contain any further branches. The decisions or the test are performed on the basis of features of the given dataset. It is a graphical representation for getting all the possible solutions to aproblem/decision based on given conditions. It is called a Decision Tree because, similar to a tree, it starts with the root node, which expands on further branches and constructs a tree-like structure. In order to build a tree, we use the CART algorithm, which stands for Classification and Regression Tree algorithm. A Decision Tree simply asks a question, and based on the answer (Yes/No), it further split the tree into subtrees.

The Decision Tree Algorithm belongs to the family of supervised machine learning algorithms. It can be used for both a classification problem as well as for a regression problem.

The goal of this algorithm is to create a model that predicts the value of a targetvariable, for which the decision tree uses the tree representation to solve the problemin which the leaf node corresponds to a class label and attributes are represented on theinternal node of the tree.

There are various algorithms in Machine learning, so choosing the best algorithm for the given dataset and problem is the main point to remember while creating a machine learning model. Below are the two reasons for using the Decision Tree:

- Decision Trees usually mimic human thinking ability while making a decision,so it is easy to understand.
- The logic behind the decision tree can be easily understood because it shows atree-like structure.

In Decision Tree the major challenge is to identify the attribute for the root node in each level. This process is known as attribute selection. We have two popular attribute selection measures:

1. **Information Gain:**

When we use a node in a Decision Tree to partition the training instances into smaller subsets, the entropy changes. Information gain is a measure of this change in entropy.

Entropy is the measure of uncertainty of a random variable, it characterizes the impurity of an arbitrary collection of examples.

The higher the entropy the more the information content.

2. **Gini Index:**

Gini Index is a metric to measure how often a randomly chosen element would be incorrectly identified. It means an attribute with lower Gini index should be preferred. Sklearn supports “Gini” criteria for Gini Index and by default, it takes “gini” value.

The most notable types of Decision Tree algorithms are:-

A) **IDichotomiser 3 (ID3):** This algorithm uses Information Gain to decide which attribute is to be used to classify the current subset of the data. For each level of the tree, information gain is calculated for the remaining data recursively.

B) **C4.5:** This algorithm is the successor of the ID3 algorithm. This algorithm uses either Information gain or Gain ratio to decide upon the classifyingattribute. It is a direct improvement from the ID3 algorithm as it can handle both continuous and missing attribute values.

C) **Classification and Regression Tree (CART):** It is a dynamic learning algorithm which can produce a regression tree as well as a classification tree depending upon the dependent variable.

Working:

In a Decision Tree, for predicting the class of the given dataset, the algorithm starts from the root node of the tree. This algorithm compares the values of the root attribute with the record (real dataset) attribute and, based on the comparison, follows the branch and jumps to the next node.

For the next node, the algorithm again compares the attribute value with the other sub-nodes and moves further. It continues the process until it reaches the leaf node of the tree. The complete process can be better understood using the below algorithm:

- Step-1: Begin the tree with the root node, says S, which contains the complete dataset.

- Step-2: Find the best attribute in the dataset using Attribute Selection Measure (ASM).

- Step-3: Divide the S into subsets that contains possible values for the best attributes.

- Step-4: Generate the Decision Tree node, which contains the best attribute.

- Step-5: Recursively make new decision trees using the subsets of the dataset created in step -3. Continue this process until a stage is reached where you cannot further classify the nodes and call the final node as a leaf node.

4.3.4 Random Forest Algorithm

Random Forest is a supervised learning algorithm. It is an extension of machine learning classifiers which include the bagging to improve the performance of Decision Tree. It combines tree predictors, and trees are dependent on a random vectorwhich is independently sampled. The distribution of all trees are the same. Random Forests splits nodes using the best among of a predictor subset that are randomly chosen from the node itself, instead of splitting nodes based on the variables. The time complexity of the worst case of learning with Random Forests is O(M(dnlogn)) , where M is the number of growing trees, n is the number of instances, and d is the datadimension.

It can be used both for classification and regression. It is also the most flexible and easy to use algorithm. A forest consists of trees. It is said that the more trees it has,the more robust a forest is. Random Forests create Decision Trees on randomlyselected data samples, get predictions from each tree and select the best solution by means of voting. It also provides a pretty good indicator of the feature importance.

Random Forests have a variety of applications, such as recommendation engines, image classification and feature selection. It can be used to classify loyal loan applicants, identify fraudulent activity and predict diseases. It lies at the base of the Boruta algorithm, which selects important features in a dataset.

Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of ensemble learning, which is a process ofcombining multiple classifiers to solve a complex problem and to improve the performance of the model.

As the name suggests, "Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset." Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output.

The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting.

Assumptions:

Since the random forest combines multiple trees to predict the class of the dataset, it is possible that some decision trees may predict the correct output, while others may not. But together, all the trees predict the correct output. Therefore, below are two assumptions for a better Random forest classifier:

- There should be some actual values in the feature variable of the dataset so thatthe classifier can predict accurate results rather than a guessed result.
- The predictions from each tree must have very low correlations.

Algorithm Steps:

It works in four steps:

- Select random samples from a given dataset.
- Construct a Decision Tree for each sample and get a predictionresult from each Decision Tree.
- Perform a vote for each predicted result.
- Select the prediction result with the most votes as the finalprediction.

Advantages:

- Random Forest is capable of performing both Classification andRegression tasks.
- It is capable of handling large datasets with high dimensionality.
- It enhances the accuracy of the model and prevents the overfitting issue.

Disadvantages:

Although Random Forest can be used for both classification and regression tasks, it isnot more suitable for Regression tasks.

4.3.5 Logistic Regression Algorithm

Logistic regression is one of the most popular Machine Learning algorithms,which comes under the Supervised Learning technique. It is used for predicting the categorical dependent variable using a given set of independent variables.

Logistic regression predicts the output of a categorical dependent variable. Therefore the outcome must be a categorical or discrete value. It can be either Yes or No, 0 or 1, true or False, etc. but instead of giving the exact value as 0 and 1, it gives the probabilistic values which lie between 0 and 1.

Logistic Regression is much similar to the Linear Regression except that how they are used. Linear Regression is used for solving Regression problems, whereas logistic regression is used for solving the classification problems.

In Logistic regression, instead of fitting a regression line, we fit an "S"shaped logistic function, which predicts two maximum values (0 or 1).

The curve from the logistic function indicates the likelihood of something suchas whether the cells are cancerous or not, a mouse is obese or not based on its weight, etc.

Logistic Regression is a significant machine learning algorithm because it has the ability to provide probabilities and classify new data using continuous and discrete datasets.

Advantages:

Logistic Regression is one of the simplest machine learning algorithms and is easy to implement yet provides great training efficiency in some cases. Also due to these reasons, training a model with this algorithm doesn't require high computation power.

The predicted parameters (trained weights) give inference about the importanceof each feature. The direction of association i.e. positive or negative is also given. So we can use Logistic Regression to find out the relationship between the features.

This algorithm allows models to be updated easily to reflect new data, unlike Decision Tree or Support Vector Machine. The update can be done using stochastic gradient descent.

Logistic Regression outputs well-calibrated probabilities along with classification results. This is an advantage over models that only give the final classification as results. If a training example has a 95% probability for a class, and another has a 55% probability for the same class, we get an inference about which training examples are more accurate for the formulated problem.

Disadvantages:

Logistic Regression is a statistical analysis model that attempts to predictprecise probabilistic outcomes based on independent features. On high dimensional datasets, this may lead to the model being over-fit on the training set, which means overstating the accuracy of predictions on the training set and thus the model may not be able to predict accurate results on the test set. This usually happens in the casewhen the model is trained on little training data with lots of features. So on high dimensional datasets, Regularization techniques should be considered to avoid over- fitting (but this makes the model complex). Very high regularization factors may even lead to the model being under-fit on the training data.

Non linear problems can't be solved with logistic regression since it has alinear decision surface. Linearly separable data is rarely found in real world scenarios. So the transformation of non linear features is required which can be done by increasing the number of features such that the data becomes linearly separable in higher dimensions.

Non-Linearly Separable Data:

It is difficult to capture complex relationships using logistic regression. More powerfuland complex algorithms such as Neural Networks can easily outperform this algorithm

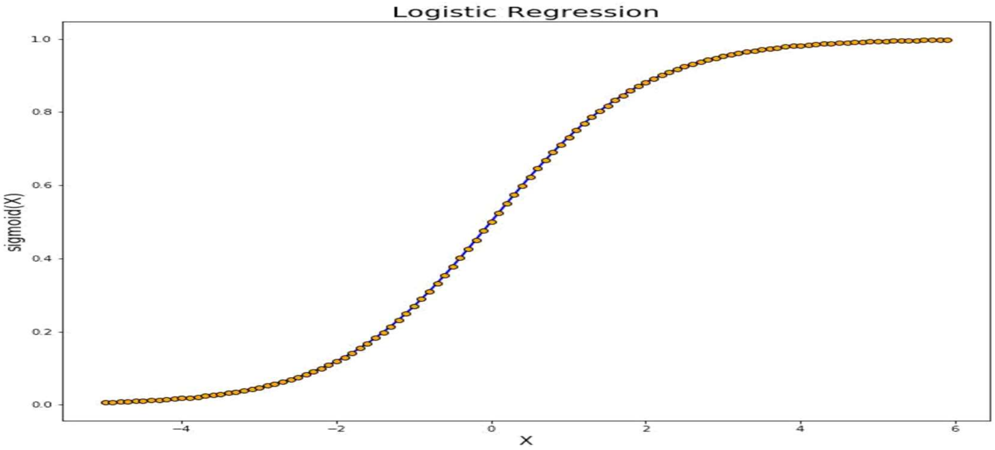


Figure: Logistic Regression

CHAPTER 5
EXPERIMENTAL ANALYSIS

5.1 SYSTEM CONFIGURATION

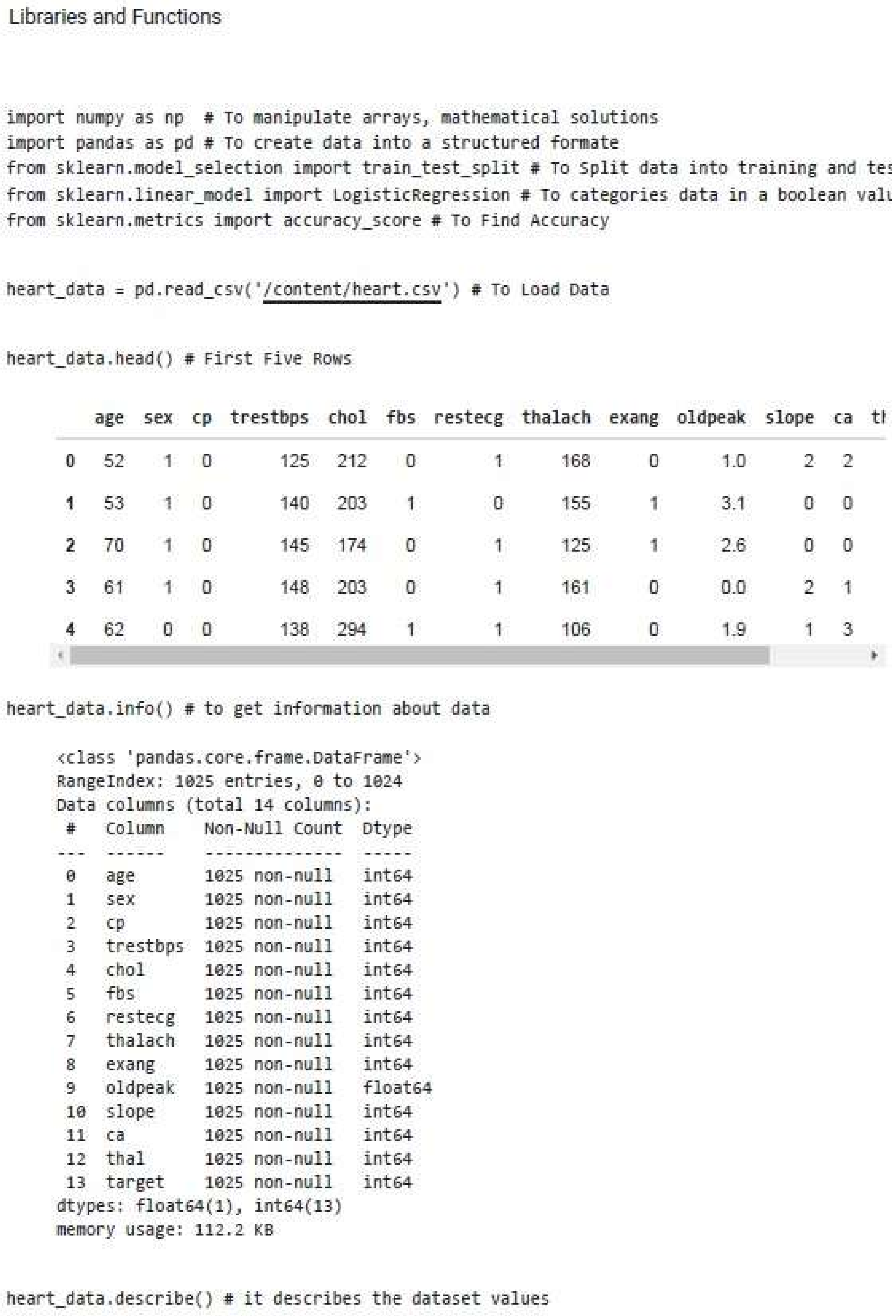
5.1.1 Hardware requirements:

Processer	:	Any Update Processer
Ram	:	Min 4GB
Hard Disk	:	Min 100GB

5.1.2 Software requirements:

Operating System	:	Windows family
Technology	:	Python3.7
IDE	:	Jupiter notebook

5.2 SAMPLE CODE



Libraries and Functions

```
import numpy as np  # To manipulate arrays, mathematical solutions
import pandas as pd # To create data into a structured formate
from sklearn.model_selection import train_test_split # To Split data into training and tes
from sklearn.linear_model import LogisticRegression # To categories data in a boolean valu
from sklearn.metrics import accuracy_score # To Find Accuracy
```

```
heart_data = pd.read_csv('/content/heart.csv') # To Load Data
```

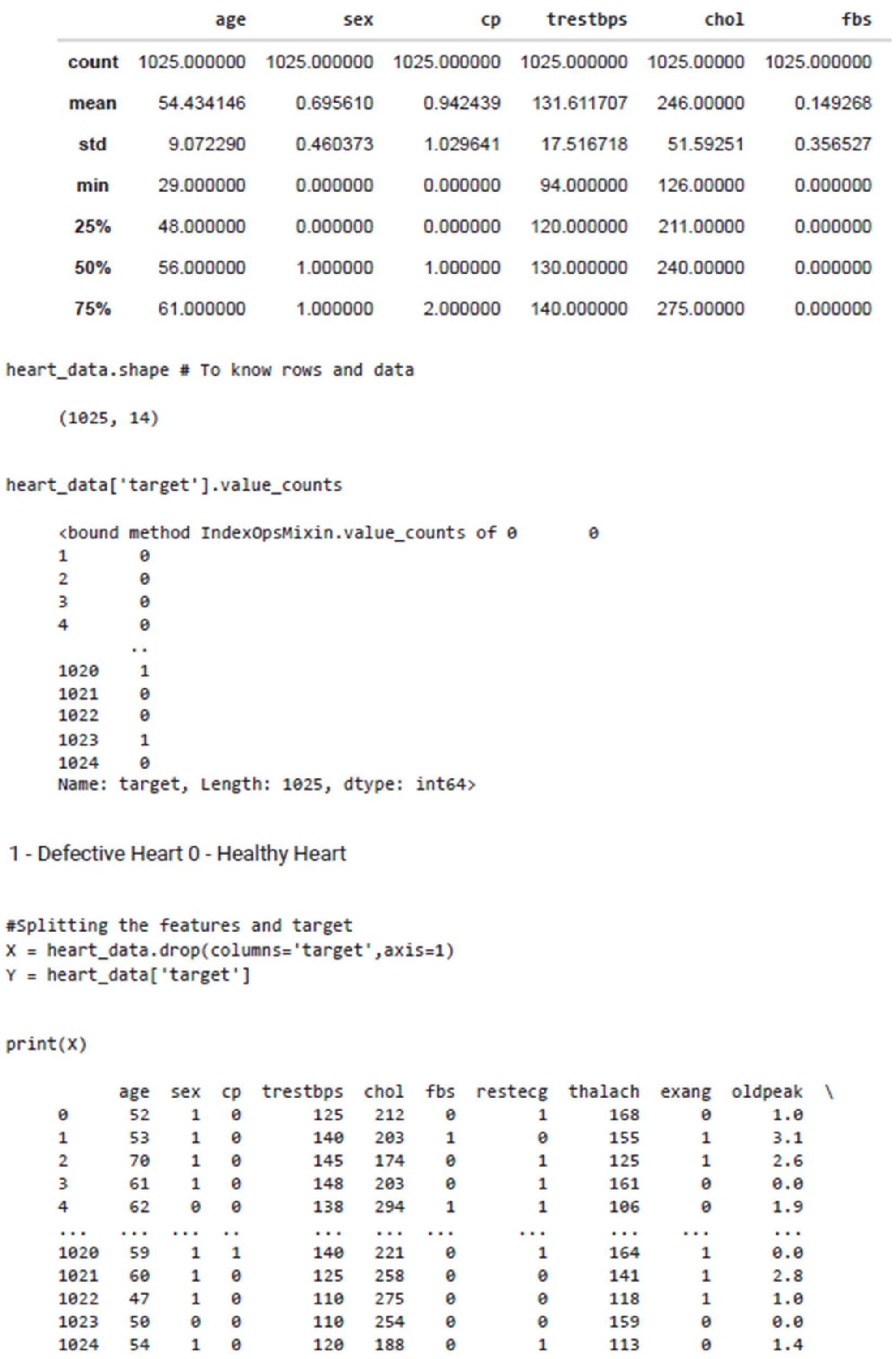
```
heart_data.head() # First Five Rows
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	tl
0	52	1	0	125	212	0	1	168	0	1.0	2	2	
1	53	1	0	140	203	1	0	155	1	3.1	0	0	
2	70	1	0	145	174	0	1	125	1	2.6	0	0	
3	61	1	0	148	203	0	1	161	0	0.0	2	1	
4	62	0	0	138	294	1	1	106	0	1.9	1	3	

```
heart_data.info() # to get information about data
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1025 entries, 0 to 1024
Data columns (total 14 columns):
 #   Column    Non-Null Count  Dtype  
---  ------    --------------  -----  
 0   age       1025 non-null   int64  
 1   sex       1025 non-null   int64  
 2   cp        1025 non-null   int64  
 3   trestbps  1025 non-null   int64  
 4   chol      1025 non-null   int64  
 5   fbs       1025 non-null   int64  
 6   restecg   1025 non-null   int64  
 7   thalach   1025 non-null   int64  
 8   exang     1025 non-null   int64  
 9   oldpeak   1025 non-null   float64
 10  slope     1025 non-null   int64  
 11  ca        1025 non-null   int64  
 12  thal      1025 non-null   int64  
 13  target    1025 non-null   int64  
dtypes: float64(1), int64(13)
memory usage: 112.2 KB
```

```
heart_data.describe() # it describes the dataset values
```



	age	sex	cp	trestbps	chol	fbs
count	1025.000000	1025.000000	1025.000000	1025.000000	1025.00000	1025.000000
mean	54.434146	0.695610	0.942439	131.611707	246.00000	0.149268
std	9.072290	0.460373	1.029641	17.516718	51.59251	0.356527
min	29.000000	0.000000	0.000000	94.000000	126.00000	0.000000
25%	48.000000	0.000000	0.000000	120.000000	211.00000	0.000000
50%	56.000000	1.000000	1.000000	130.000000	240.00000	0.000000
75%	61.000000	1.000000	2.000000	140.000000	275.00000	0.000000

```
heart_data.shape # To know rows and data
```

```
(1025, 14)
```

```
heart_data['target'].value_counts
```

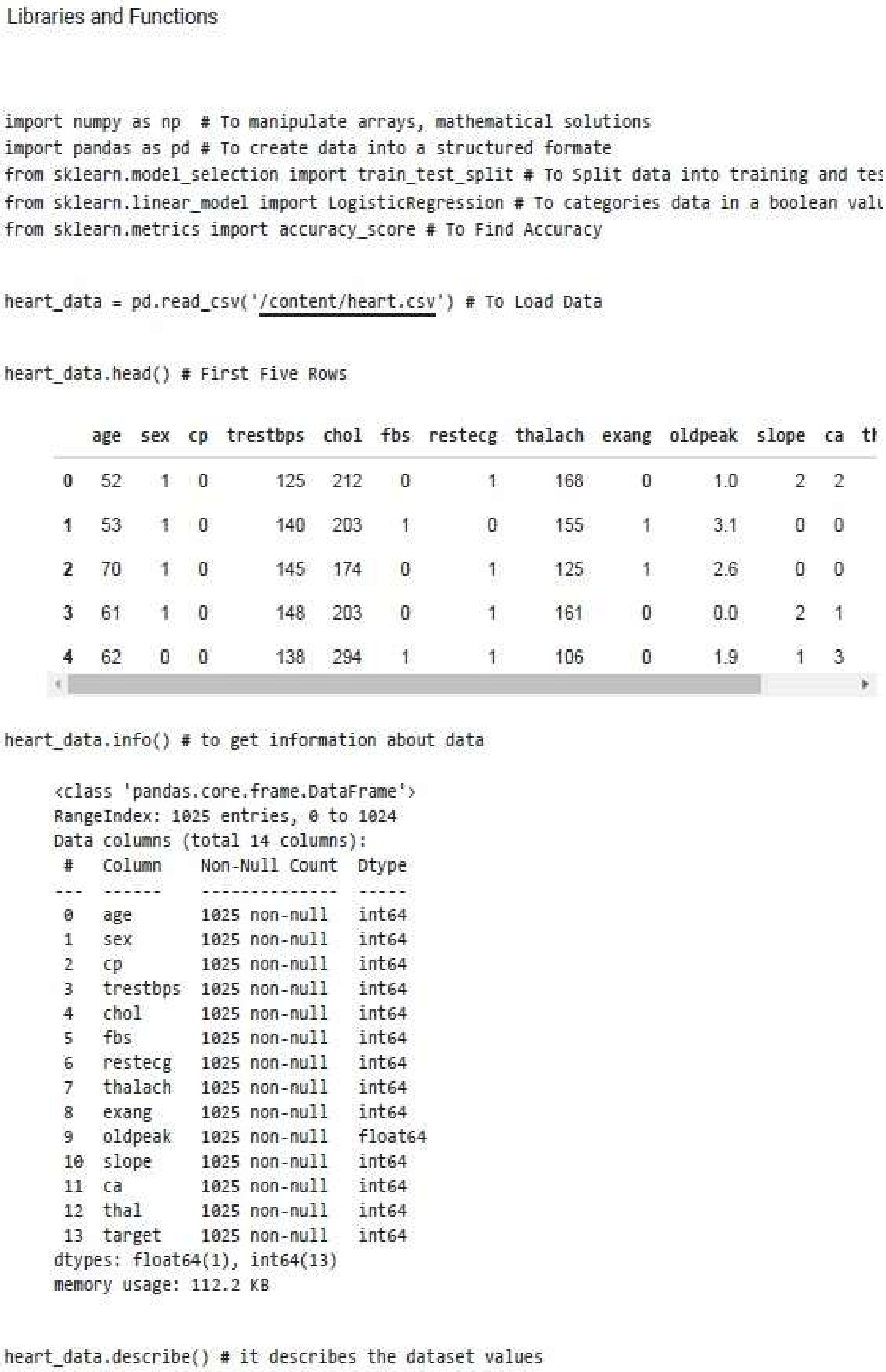
```
<bound method IndexOpsMixin.value_counts of 0       0
1       0
2       0
3       0
4       0
       ..
1020    1
1021    0
1022    0
1023    1
1024    0
Name: target, Length: 1025, dtype: int64>
```

1 - Defective Heart 0 - Healthy Heart

```
#Splitting the features and target
X = heart_data.drop(columns='target',axis=1)
Y = heart_data['target']
```

```
print(X)
```

```
      age  sex  cp  trestbps  chol  fbs  restecg  thalach  exang  oldpeak  \
0      52    1   0       125   212    0        1      168      0      1.0
1      53    1   0       140   203    1        0      155      1      3.1
2      70    1   0       145   174    0        1      125      1      2.6
3      61    1   0       148   203    0        1      161      0      0.0
4      62    0   0       138   294    1        1      106      0      1.9
...   ...  ...  ..       ...   ...  ...      ...      ...    ...      ...
1020   59    1   1       140   221    0        1      164      1      0.0
1021   60    1   0       125   258    0        0      141      1      2.8
1022   47    1   0       110   275    0        0      118      1      1.0
1023   50    0   0       110   254    0        0      159      0      0.0
1024   54    1   0       120   188    0        1      113      0      1.4
```



Libraries and Functions

```
import numpy as np  # To manipulate arrays, mathematical solutions
import pandas as pd # To create data into a structured formate
from sklearn.model_selection import train_test_split # To Split data into training and tes
from sklearn.linear_model import LogisticRegression # To categories data in a boolean valu
from sklearn.metrics import accuracy_score # To Find Accuracy
```

```
heart_data = pd.read_csv('/content/heart.csv') # To Load Data
```

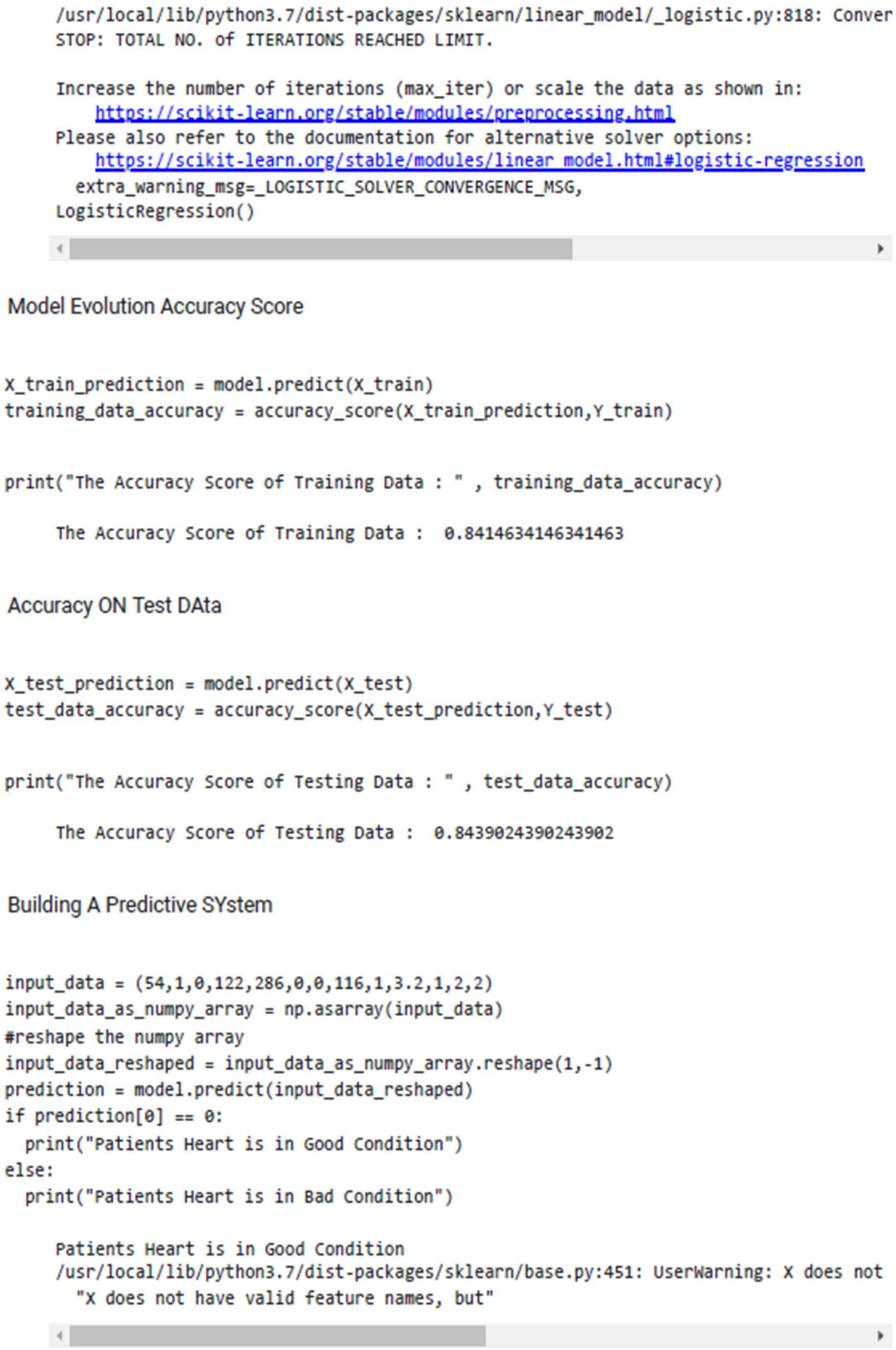
```
heart_data.head() # First Five Rows
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	tl
0	52	1	0	125	212	0	1	168	0	1.0	2	2	
1	53	1	0	140	203	1	0	155	1	3.1	0	0	
2	70	1	0	145	174	0	1	125	1	2.6	0	0	
3	61	1	0	148	203	0	1	161	0	0.0	2	1	
4	62	0	0	138	294	1	1	106	0	1.9	1	3	

```
heart_data.info() # to get information about data
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1025 entries, 0 to 1024
Data columns (total 14 columns):
 #   Column    Non-Null Count  Dtype  
---  ------    --------------  -----  
 0   age       1025 non-null   int64  
 1   sex       1025 non-null   int64  
 2   cp        1025 non-null   int64  
 3   trestbps  1025 non-null   int64  
 4   chol      1025 non-null   int64  
 5   fbs       1025 non-null   int64  
 6   restecg   1025 non-null   int64  
 7   thalach   1025 non-null   int64  
 8   exang     1025 non-null   int64  
 9   oldpeak   1025 non-null   float64
 10  slope     1025 non-null   int64  
 11  ca        1025 non-null   int64  
 12  thal      1025 non-null   int64  
 13  target    1025 non-null   int64  
dtypes: float64(1), int64(13)
memory usage: 112.2 KB
```

```
heart_data.describe() # it describes the dataset values
```



```
/usr/local/lib/python3.7/dist-packages/sklearn/linear_model/_logistic.py:818: Conver
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  extra_warning_msg=_LOGISTIC_SOLVER_CONVERGENCE_MSG,
LogisticRegression()
```

Model Evolution Accuracy Score

```
X_train_prediction = model.predict(X_train)
training_data_accuracy = accuracy_score(X_train_prediction,Y_train)
```

```
print("The Accuracy Score of Training Data : " , training_data_accuracy)
```

```
The Accuracy Score of Training Data :  0.8414634146341463
```

Accuracy ON Test DAta

```
X_test_prediction = model.predict(X_test)
test_data_accuracy = accuracy_score(X_test_prediction,Y_test)
```

```
print("The Accuracy Score of Testing Data : " , test_data_accuracy)
```

```
The Accuracy Score of Testing Data :  0.8439024390243902
```

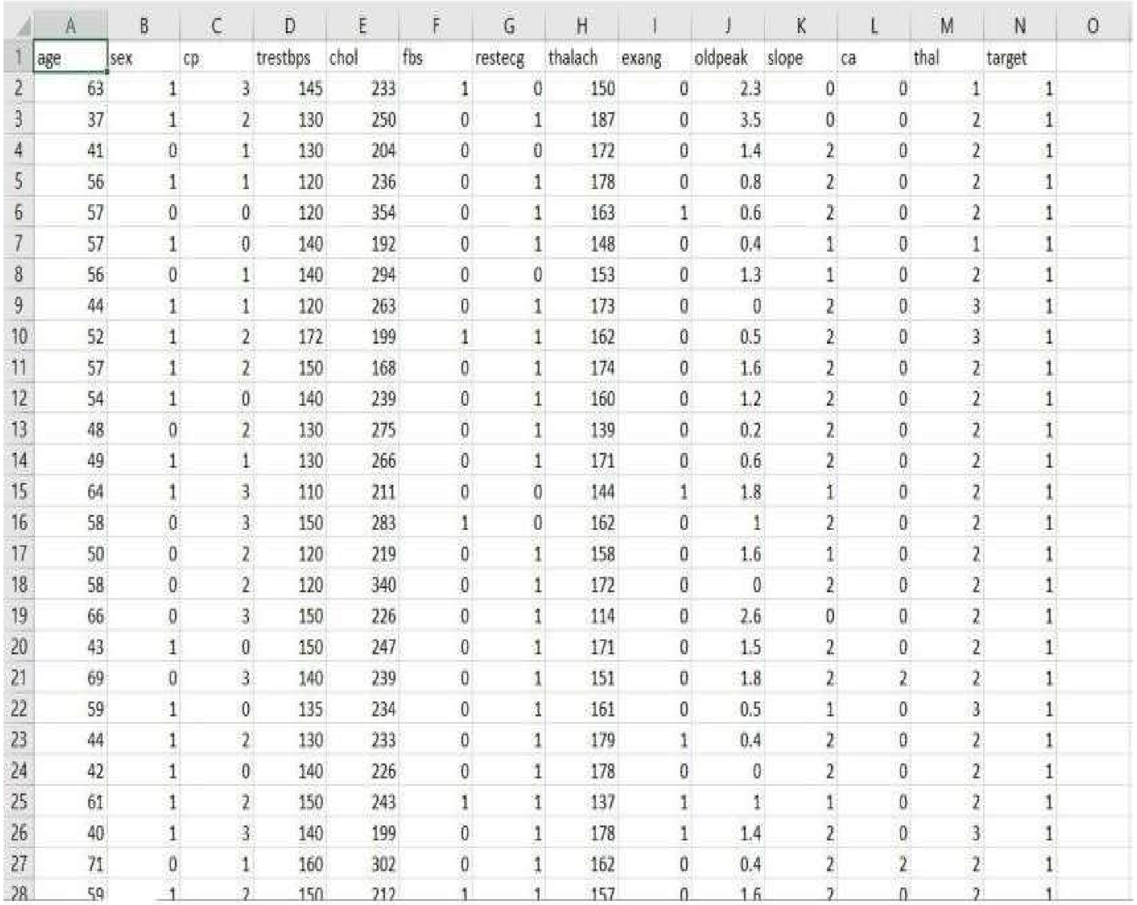
Building A Predictive SYstem

```
input_data = (54,1,0,122,286,0,0,116,1,3.2,1,2,2)
input_data_as_numpy_array = np.asarray(input_data)
#reshape the numpy array
input_data_reshaped = input_data_as_numpy_array.reshape(1,-1)
prediction = model.predict(input_data_reshaped)
if prediction[0] == 0:
  print("Patients Heart is in Good Condition")
else:
  print("Patients Heart is in Bad Condition")
```

```
Patients Heart is in Good Condition
/usr/local/lib/python3.7/dist-packages/sklearn/base.py:451: UserWarning: X does not
  "X does not have valid feature names, but"
```

5.3 DATASET DETAILS

- Of the 76 attributes available in the dataset,14 attributes are considered for the prediction of the output.
- Heart Disease UCI : https://archive.ics.uci.edu/ml/datasets/Heart+Disease



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target	
2	63	1	3	145	233	1	0	150	0	2.3	0	0	1	1	
3	37	1	2	130	250	0	1	187	0	3.5	0	0	2	1	
4	41	0	1	130	204	0	0	172	0	1.4	2	0	2	1	
5	56	1	1	120	236	0	1	178	0	0.8	2	0	2	1	
6	57	0	0	120	354	0	1	163	1	0.6	2	0	2	1	
7	57	1	0	140	192	0	1	148	0	0.4	1	0	1	1	
8	56	0	1	140	294	0	0	153	0	1.3	1	0	2	1	
9	44	1	1	120	263	0	1	173	0	0	2	0	3	1	
10	52	1	2	172	199	1	1	162	0	0.5	2	0	3	1	
11	57	1	2	150	168	0	1	174	0	1.6	2	0	2	1	
12	54	1	0	140	239	0	1	160	0	1.2	2	0	2	1	
13	48	0	2	130	275	0	1	139	0	0.2	2	0	2	1	
14	49	1	1	130	266	0	1	171	0	0.6	2	0	2	1	
15	64	1	3	110	211	0	0	144	1	1.8	1	0	2	1	
16	58	0	3	150	283	1	0	162	0	1	2	0	2	1	
17	50	0	2	120	219	0	1	158	0	1.6	1	0	2	1	
18	58	0	2	120	340	0	1	172	0	0	2	0	2	1	
19	66	0	3	150	226	0	1	114	0	2.6	0	0	2	1	
20	43	1	0	150	247	0	1	171	0	1.5	2	0	2	1	
21	69	0	3	140	239	0	1	151	0	1.8	2	2	2	1	
22	59	1	0	135	234	0	1	161	0	0.5	1	0	3	1	
23	44	1	2	130	233	0	1	179	1	0.4	2	0	2	1	
24	42	1	0	140	226	0	1	178	0	0	2	0	2	1	
25	61	1	2	150	243	1	1	137	1	1	1	0	2	1	
26	40	1	3	140	199	0	1	178	1	1.4	2	0	3	1	
27	71	0	1	160	302	0	1	162	0	0.4	2	2	2	1	
28	59	1	2	150	212	1	1	157	0	1.6	2	0	2	1	

Figure: Dataset Attributes

Input dataset attributes

- Gender (value 1: Male; value 0 : Female)
- Chest Pain Type (value 1: typical type 1 angina, value 2: typical type angina, value 3: non-angina pain; value 4: asymptomatic)
- Fasting Blood Sugar (value 1: > 120 mg/dl; value 0:< 120 mg/dl)
- Exang – exercise induced angina (value 1: yes; value 0: no)
- CA – number of major vessels colored by fluoroscopy (value 0 – 3)
- Thal (value 3: normal; value 6: fixed defect; value 7:reversible defect)
- Trest Blood Pressure (mm Hg on admission to the hospital)
- Serum Cholesterol (mg/dl)
- Thalach – maximum heart rate achieved
- Age in Year
- Height in cms
- Weight in Kgs.
- Cholestrol
- Restecg

S. No.	Attribute	Description	Type
1	Age	Patient’s age (29 to 77)	Numerical
2	Sex	Gender of patient(male-0 female-1)	Nominal
3	Cp	Chest pain type	Nominal
4	Trestbps	Resting blood pressure(in mm Hg on admission to hospital ,values from 94 to 200)	Numerical
5	Chol	Serum cholesterol in mg/dl, values from 126 to 564)	Numerical
6	Fbs	Fasting blood sugar>120 mg/dl, true-1 false-0)	Nominal
7	Resting	Resting electrocardiographics result (0 to 1)	Nominal
8	Thali	Maximum heart rate achieved(71 to 202)	Numerical
9	Exang	Exercise included agina(1-yes 0-no)	Nominal
10	Oldpeak	ST depression introduced by exercise relative to rest (0 to .2)	Numerical
11	Slope	The slop of the peak exercise ST segment (0 to 1)	Nominal
12	Ca	Number of major vessels (0-3)	Numerical
13	Thal	3-normal	Nominal
14	Targets	1 or 0	Nominal

TABLE 2: Attributes of the dataset

5.4 PERFORMANCE ANALYSIS

In this project, various machine learning algorithms like SVM, Naive Bayes, DecisionTree, Random Forest, Logistic Regression, Adaboost, XG-boost are used to predict heart disease. Heart Disease UCI dataset, has a total of 76 attributes, out of those only 14 attributes are considered for the prediction of heart disease. Various attributes of thepatient like gender, chest pain type, fasting blood pressure, serum cholesterol, exang, etc are considered for this project. The accuracy for individual algorithms has to measure and whichever algorithm is giving the best accuracy,that is considered for the heart disease prediction. For evaluating the experiment, various evaluation metrics likeaccuracy, confusion matrix, precision, recall, and f1-score are considered.

Accuracy- Accuracy is the ratio of the number of correct predictions to the totalnumber of inputs in the dataset. It is expressed as:

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

Confusion Matrix- It gives us a matrix as output and gives the total performance of thesystem.

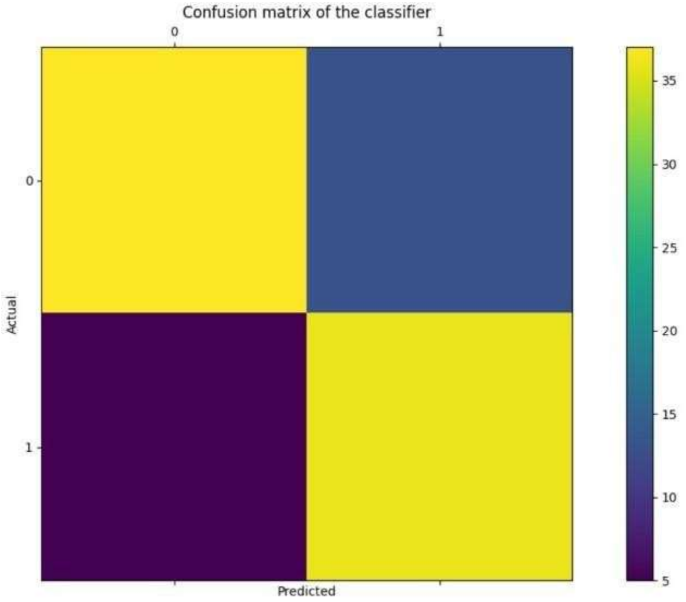


Figure: Confusion Matrix

Where

TP: True positive

FP: False Positive

FN: False Negative

TN: True Negative

Correlation Matrix: The correlation matrix in machine learning is used for feature selection. It represents dependency between various attributes.

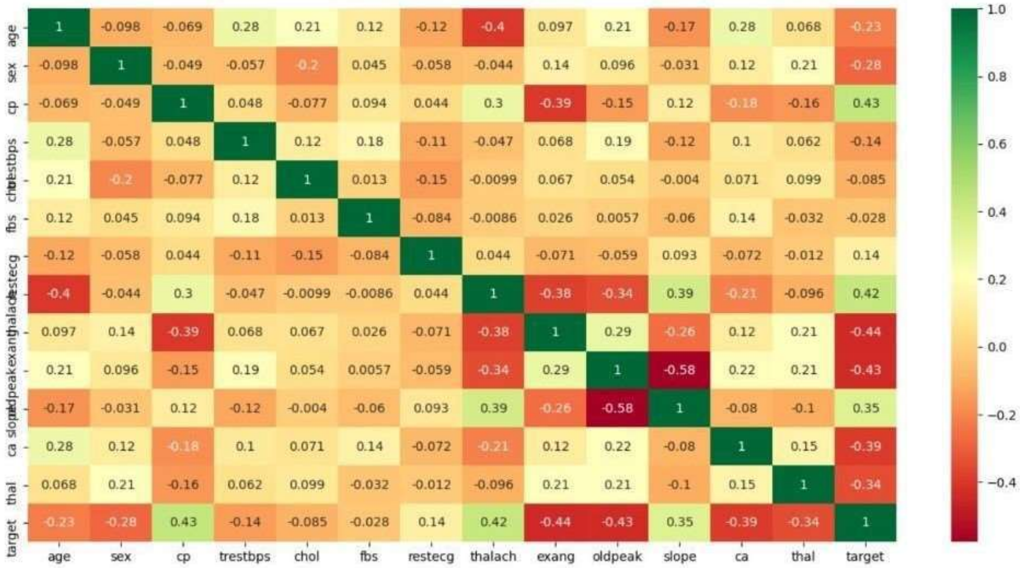


Fig: Correlation matrix

Precision- It is the ratio of correct positive results to the total number of positiveresults predicted by the system.

It is expressed as:

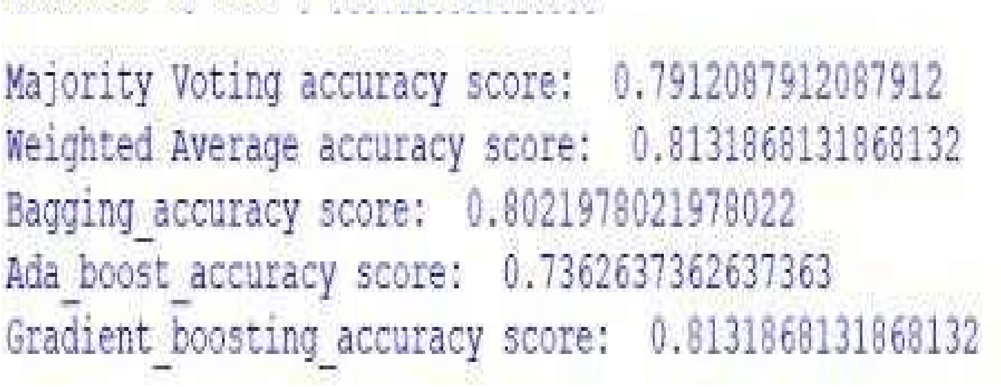
Recall-It is the ratio of correct positive results to the total number of positive results predicted by the system.

It is expressed as:

F1 Score-It is the harmonic mean of Precision and Recall. It measures the test accuracy. The range of this metric is 0 to 1.

5.5 PERFORMANCE MEASURES

- The highest accuracy is given by XG-boost.



```
Majority Voting accuracy score:  0.7912087912087912
Weighted Average accuracy score:  0.8131868131868132
Bagging_accuracy score:  0.8021978021978022
Ada_boost_accuracy score:  0.7362637362637363
Gradient_boosting_accuracy score:  0.8131868131868132
```

5.6 RESULT

After performing the machine learning approach for training and testing we find that accuracy of the XG-boost is better compared to other algorithms. Accuracy is calculated with the support of the confusion matrix of each algorithm, here the numbercount of TP, TN, FP, FN is given and using the equation of accuracy, value has been calculated and it is concluded that extreme gradient boosting is best with 81% accuracy and the comparison is shown below.
TABLE: Accuracy comparison of algorithms Algorithm Accuracy

Algorithm	Accuracy
XG-boost	81.3%
SVM	80.2%
Logistic Regression	79.1%
Random Forest	79.1%
Naive Bayes	76.9%
Decision Tree	75.8%
Adaboost	73.6%

TABLE 2: Accuracy Table

CHAPTER 6
CONCLUSION AND FUTURE WORK

Heart diseases are a major killer in India and throughout the world, applicationof promising technology like machine learning to the initial prediction of heart diseases will have a profound impact on society. The early prognosis of heart disease 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. Thenumber of people facing heart diseases is on a raise each year. This prompts for its early diagnosis and treatment. The utilization of suitable technology support in this regard can prove to be highly beneficial to the medical fraternity and patients. In this paper, the seven different machine learning algorithms used to measure the performance are SVM, Decision Tree, Random Forest, Naïve Bayes, Logistic Regression, Adaptive Boosting, and Extreme Gradient Boosting applied on the dataset.

The expected attributes leading to heart disease in patients are available in thedataset which contains 76 features and 14 important features that are useful to evaluate the system are selected among them. If all the features taken into the consideration then the efficiency of the system the author gets is less. To increase efficiency, attribute selection is done. In this n features have to be selected for evaluating the model which gives more accuracy. The correlation of some features inthe dataset is almost equal and so they are removed. If all the attributes present in thedataset are taken into account then the efficiency decreases considerably.

All the seven machine learning methods accuracies are compared based on which one prediction model is generated. Hence, the aim is to use various evaluationmetrics like confusion matrix, accuracy, precision, recall, and f1-score which predicts the disease efficiently. Comparing all seven the extreme gradient boosting classifier gives the highest accuracy of 81%.

APPENDIX

Python

Python is an interpreted, high-level, general purpose programming language createdby Guido Van Rossum and first released in 1991, Python's design philosophy emphasizes code Readability with its notable use of significant White space. Its language constructs and object oriented approach aim to help programmers write clear,logical code for small and large-scale projects. Python is dynamically typed andgarbage collected. It supports multiple programming paradigms, including procedural, object-oriented, and functional programming.

Sklearn

Scikit-learn (Sklearn) is the most useful and robust library for machine learning in Python. It provides a selection of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reductionvia a consistent interface in Python. This library, which is largely written in Python, isbuilt upon NumPy, SciPy and Matplotlib.

Numpy

NumPy is a library for the python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high level mathematical functions to operate on these arrays. The ancestor of NumPy, Numeric, was originally created by Jim with contributions from several other developers. In 2005, Travis created NumPy by incorporating features of the competing Numarray into Numeric, with extensive modifications. NumPy is open source software and has many contributors.

Matplotlib

Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy. It provides an object-oriented API for embedding plots into applications using general-purpose GUI toolkits like Tkinter, wxPython, Qt, or GTK. There is also a procedural "pylab" interface based on a statemachine (like OpenGL), designed to closely resemble that of MATLAB, though its use is discouraged.

Seaborn

Seaborn is a Python data visualization library based on matplotlib. It provides a high- level interface for drawing attractive and informative statistical graphics. Seaborn is a library in Python predominantly used for making statistical graphics. Seaborn is a data visualization library built on top of matplotlib and closely integrated with pandas data structures in Python. Visualization is the central part of Seaborn which helps in exploration and understanding of data.

SciPy

SciPy contains modules for optimization, linearalgebra, integration, interpolation, special functions, FFT, signal and imageprocessing, ODE solvers and other taskscommon in science and engineering. SciPy is also a family of conferences for usersand developers of these tools: SciPy (in the United States), EuroSciPy (in Europe) and SciPy.in (in India). Enthought originated the SciPy conference in the United States andcontinues to sponsor many of the international conferences as well as host the SciPy website. SciPy is a scientific computation library that uses NumPy underneath. It provides more utility functions for optimization, stats and signal processing.

REFERENCES

[1] Soni J, Ansari U, Sharma D & Soni S (2011). Predictive data mining for medical diagnosis: an overview of heart disease prediction. International Journal of Computer Applications, 17(8), 43-8

[2] Dangare C S & Apte S S (2012). Improved study of heart disease prediction system using data mining classification techniques. International Journal of Computer Applications, 47(10), 44-8.

[3] Ordonez C (2006). Association rule discovery with the train and test approach for heart disease prediction. IEEE Transactions on Information Technology in Biomedicine, 10(2), 334-43.

[4] Shinde R, Arjun S, Patil P & Waghmare J (2015). An intelligent heart disease prediction system using k-means clustering and Naïve Bayes algorithm. International Journal of Computer Science and Information Technologies, 6(1), 637-9.

[5] Bashir S, Qamar U & Javed M Y (2014, November). An ensemble-based decision support framework for intelligent heart disease diagnosis. In International Conference on Information Society (i-Society 2014) (pp. 259-64). IEEE. ICCRDA 2020 IOP Conf. Series: Materials Science and Engineering 1022 (2021) 012072 IOP Publishing doi:10.1088/1757-899X/1022/1/012072 9

[6] Jee S H, Jang Y, Oh D J, Oh B H, Lee S H, Park S W & Yun Y D (2014). A coronary heart disease prediction model: the Korean Heart Study. BMJ open, 4(5), e005025.

[7] Ganna A, Magnusson P K, Pedersen N L, de Faire U, Reilly M, Ärnlöv J & Ingelsson E (2013). Multilocus genetic risk scores for coronary heart disease prediction. Arteriosclerosis, thrombosis, and vascular biology, 33(9), 2267-72.

[8] Jabbar M A, Deekshatulu B L & Chandra P (2013, March). Heart disease prediction using lazy associative classification. In 2013 International Mutli-Conference on Automation, Computing,Communication, Control and Compressed Sensing (iMac4s) (pp. 40- 6). IEEE.

[9] Brown N, Young T, Gray D, Skene A M & Hampton J R (1997). Inpatient deaths from acute myocardial infarction, 1982-92: analysis of data in the Nottingham heart attack register. BMJ, 315(7101), 159-64.

[10] Folsom A R, Prineas R J, Kaye S A & Soler J T (1989). Body fat distribution and self-reported prevalence of hypertension, heart attack, and other heart disease in older women. International journal of epidemiologyy, 18(2), 361-7.

[11] Chen A H, Huang S Y, Hong P S, Cheng C H & Lin E J (2011, September). HDPS: Heart disease prediction system. In 2011 Computing in Cardiology (pp. 557-60). IEEE.

[12] Parthiban, Latha and R Subramanian. "Intelligent heart disease prediction system using CANFIS and genetic algorithm." International Journal of Biological, Biomedical and Medical Sciences 3.3 (2008).

[13] Wolgast G, Ehrenborg C, Israelsson A, Helander J, Johansson E & Manefjord H (2016). Wireless body area network for heart attack detection [Education Corner]. IEEE antennas and propagation magazine, 58(5), 84-92.

[14] Patel S & Chauhan Y (2014). Heart attack detection and medical attention using motion sensing device -kinect. International Journal of Scientific and Research Publications, 4(1), 1-4.

[15] Piller L B, Davis B R, Cutler J A, Cushman W C, Wright J T, Williamson J D & Haywood L J (2002). Validation of heart failure events in the Antihypertensive and Lipid Lowering Treatment to Prevent Heart Attack Trial (ALLHAT) participants assigned to doxazosin and chlorthalidone. Current controlled trials in cardiovascular medicine

[16] Raihan M, Mondal S, More A, Sagor M O F, Sikder G, Majumder M A & Ghosh K (2016, December). Smartphone based ischemic heart disease (heart attack) risk prediction using clinical data and data mining approaches, a prototype design. In 2016 19th International Conference on Computer and Information Technology (ICCIT) (pp. 299-303). IEEE.

[17] A. Aldallal and A. A. A. Al-Moosa, "Using Data Mining Techniques to Predict Diabetes and Heart Diseases", 2018 4th International Conference on Frontiers of Signal Processing (ICFSP), pp. 150-154, 2018, September.

[18] Takci H (2018). Improvement of heart attack prediction by the feature selection methods. Turkish Journal of Electrical Engineering & Computer Sciences, 26(1), 1-10.

[19] Ankita Dewan and Meghna Sharma, "Prediction of heart disease using a hybrid technique in data mining classification", 2015 2nd International Conference on Computing for Sustainable Global Development (INDIACom)

[20] Aditya Methaila, Prince Kansal, Himanshu Arya and Pankaj Kumar, "Early heart disease prediction using data mining techniques", Computer Science & Information Technology Journal, pp. 53-59, 2014.