VISVESVARAYA TECHNOLOGICAL UNIVERSITY

JNANA SANGAMA, BELAGAVI – 590 018

An Internship Project Report on

"HEART DISEASE PREDICTION"

Submitted in partial fulfillment of the requirements for the Final year degree of
***Bachelor of Engineering in Computer Science and Engineering** of*
Visvesvaraya Technological University, Belagavi

Submitted By

Namratha H V **1RN18CS067**

Carried out at

NASTECH

Under the guidance of:

Internal Guide:
Mrs. Heerah D
Assistant Professor
Department of CSE

External Guide:
Mr. Aman Upadhyay
NASTECH
Bengaluru

ESTD:2001
An Institute with a Difference

Department of Computer Science and Engineering
(NBA Accredited for academic years 2018-19, 2019-20, 2020-21, 2021-22)
RNS Institute of Technology
Dr. Vishnuvardhan Road, Channasandra, Bengaluru-560098

2021-2022

RNS INSTITUTE OF TECHNOLOGY

Dr. Vishnuvardhan Road, Rajarajeshwari Nagar post, Channasandra, Bengaluru - 560098

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

(NBA Accredited for academic years 2018-19, 2019-20, 2020-21, 2021-22)

CERTIFICATE

Certified that the Internship/Professional Practice work entitled ***"Heart Disease Prediction"*** has been successfully carried out at **"NASTECH"** by **Namratha H V (1RN18CS067)**, Bonafide student of **RNS Institute of Technology, Bengaluru** in partial fulfillment of the requirements of Final year degree in **Bachelor of Engineering in Computer Science and Engineering** of **Visvesvaraya Technological University, Belagavi** during academic year **2021-2022**. The internship report has been approved as it satisfies the academic requirements in respect of internship work for the said degree.

Mrs. Heerah D	**Dr. Kiran P**	**Dr. M K Venkatesha**
Internal Guide	Professor and HoD	Principal
Assistant Professor	Department of CSE	RNSIT
Department of CSE	RNSIT	

External Viva

Name of the Examiners	**Signature with Date**
1. _____	_____
2. _____	_____

CERTIFICATE

This is to certify that **NAMRATHA H V** from RNSIT has taken part in internship training on the **Artificial Intelligence & Machine Learning – Heart Disease Prediction** project conducted by New Age Solutions Technologies (NASTECH).

NAMRATHA H V has shown great enthusiasm and interest in the project. The incumbents' conduct and performance were found satisfactory during all phases of the project.

Thank you very much.

Sincerely Yours,

Deepak Garg

Founder

Address: Block No.10, Kailashnagar, Highland Residency, Kolshet Road, Thane (W)- 400608, Mumbai, Maharashtra.
E-mail: deeprgarg@nastech.in

ABSTRACT

Nowadays, health diseases are increasing day by day due to lifestyle, hereditary. Especially, heart diseases have become more common these days, i.e. the lives of people are at risk. Each individual has different values for Blood pressure, cholesterol and pulse rate. But according to medically proven results the normal values of Blood pressure is 120/90, Cholesterol is 100-129 mg/dL, Pulse rate is 72, Fasting Blood Sugar level is 100 mg/dL, Heart rate is 60-100 bpm, ECG is normal, Width of major vessels is 25 mm (1 inch) in the aorta to only 8 μm in the capillaries. This internship project gives the survey about different classification techniques used for predicting the risk level of each person based on age, gender, Blood pressure, cholesterol, pulse rate, etc.

"Heart Disease Prediction" system based on predictive modeling predicts whether a person is suffering from Heart Disease or not on the basis of the symptoms that the user provides as an input to the system. The system analyzes the symptoms provided by the user as input and gives whether a person is suffering from Heart Disease or not as an output. Heart Disease Prediction is done by implementing four techniques such as Random Forest, K-Nearest Neighbors, Decision Tree, Naive Bayes Algorithms. These techniques calculate the probability of heart disease.

The main motivation of doing this internship project is to present a heart disease prediction model for the prediction of occurrence of heart disease. Further, this internship project is aimed towards identifying the best classification algorithm for identifying the possibility of heart disease in a patient. This internship project work is justified by performing a comparative study and analysis using four classification algorithms namely Random Forest, K-Nearest Neighbors, Decision Tree, Naive Bayes are used at different levels of evaluations. Although these are commonly used machine learning algorithms, the heart disease prediction is a vital task involving highest possible accuracy. Hence, the four algorithms are evaluated at numerous levels and types of evaluation strategies. This will provide researchers and medical practitioners to establish a better understanding and help them identify a solution to identify the best method for predicting heart diseases.

ACKNOWLEDGMENT

At the very onset, I would like to place on record my gratitude to all those people who have helped me in making this Internship project work a reality. Our Institution has played a paramount role in guiding in the right direction.

I would like to profoundly thank **Sri. Satish R Shetty**, Managing Director, RNS Group of Companies, Bengaluru for providing such a healthy environment for the successful completion of this Internship project work.

I would like to thank our beloved Principal, **Dr. M K Venkatesha**, for providing the necessary facilities to carry out this Internship project work.

I am extremely grateful to **Dr. Kiran P**, Professor and Head of Department of Computer Science and Engineering for having accepted to patronize me in the right direction with all his wisdom.

I would like to express my sincere thanks to our Coordinator**, Mrs. Chethana H R,** Assistant Professor, Department of Computer Science and Engineering for her constant encouragement that motivated me for the successful completion of this Internship project work.

I would like to thank my guide **Mrs. Heerah D,** Assistant Professor, Department of Computer Science and Engineering for her continuous guidance and constructive suggestions for this Internship project work.

I thank **Mr. Aman Upadhyay, NASTECH** for providing the opportunity to be a part of the Internship program and having guided me to complete the same successfully.

Last but not the least, I am thankful to all the teaching and non-teaching staff members of the Computer Science and Engineering Department for their encouragement and support throughout this Internship project work.

Namratha H V

1RN18CS067

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1. ORGANIZATION/ INDUSTRY

1.1.1. Company Profile

NASTECH is formed with the purpose of bridging the gap between Academia and Industry. It is one of the leading Global Certification and Training service providers for technical and management programs for educational institutions. They collaborate with educational institutes to understand their requirements and form a strategy in consultation with all stakeholders to fulfill those by skilling, reskilling and upskilling the students and faculties on new age skills and technologies.

Nastech offers industry and project-oriented training programs which not only expose students to hands-on training experience but also make them practical oriented towards the industry readiness expected in today's time. Their programs are mapped to a certain Global Certification Exams that is after the students are done with their training, they will prove themselves on a global level via a global certification exam.

The company leads from the front in terms of costing of overall global certification and training programs.

1.1.2. Domain/ Technology

Machine learning (**ML**) is the study of computer algorithms that can improve automatically through experience and by the use of data. It is seen as a part of artificial intelligence. Machine learning algorithms build a model based on sample data, known as training data, in order to make predictions or decisions without being explicitly programmed to do so. Machine learning algorithms are used in a wide variety of applications, such as in medicine, email filtering, speech recognition, and computer vision, where it is difficult or unfeasible to develop conventional algorithms to perform the needed tasks.

In the past decade, machine learning has given us self-driving cars, practical speech recognition, effective web search, and a vastly improved understanding of the human

genome. Machine learning is so pervasive today that you probably use it dozens of times a day without knowing it. Many researchers also think it is the best way to make progress towards human-level AI.

Resurging interest in machine learning is due to the same factors that have made data mining and Bayesian analysis more popular than ever. Things like growing volumes and varieties of available data, computational processing that is cheaper and more powerful, and affordable data storage.

All of these things mean it's possible to quickly and automatically produce models that can analyze bigger, more complex data and deliver faster, more accurate results – even on a very large scale. And by building precise models, an organization has a better chance of identifying profitable opportunities or avoiding unknown risks.

1.2. Introduction to Heart Disease

In day to day life many factors affect a human heart. Many problems are occurring at a rapid pace and new heart diseases are rapidly being identified. In today's world of stress, Heart being an essential organ in the human body which pumps blood throughout the body for the blood circulation is essential and its health is to be conserved for a healthy living. The health of a human heart is based on the experiences in a person's life and is completely dependent on professional and personal behaviors of a person. There may also be several genetic factors through which a type of heart disease is passed down from generations. According to the World Health Organization, every year more than 12 million deaths are occurring worldwide due to the various types of heart diseases which is also known by the term cardiovascular disease. The term Heart disease includes many diseases that are diverse and specifically affect the heart and the arteries of a human being. Even young aged people around their 20-30 years of lifespan are getting affected by heart diseases. The increase in the possibility of heart disease among young people may be due to the bad eating habits, lack of sleep, restless nature, depression and numerous other factors such as obesity, poor diet, family history, high blood pressure, high blood cholesterol, idle behavior, family history, smoking and hypertension.

The diagnosis of heart diseases is very important and is itself the most complicated task in the medical field. All the mentioned factors are taken into consideration when analyzing and understanding the patients by the doctor through manual check-ups at regular intervals of time.

The symptoms of heart disease greatly depend upon which of the discomfort felt by an individual. Some symptoms are not usually identified by the common people. However, common symptoms include chest pain, breathlessness, and heart palpitations. The chest pain common to many types of heart disease is known as angina, or angina pectoris, and occurs when a part of the heart does not receive enough oxygen. Angina may be triggered by stressful events or physical exertion and normally lasts under 10 minutes. Heart attacks can also occur as a result of different types of heart disease. The signs of a heart attack are like angina except that they can occur during rest and tend to be more severe. The symptoms of a heart attack can sometimes resemble indigestion. Heartburn and a stomach ache can occur, as well as a heavy feeling in the chest. Other symptoms of a heart attack include pain that travels through the body, for example from the chest to the arms, neck,

back, abdomen, or jaw, lightheadedness and dizzy sensations, profuse sweating, nausea and vomiting.

Heart failure is also an outcome of heart disease, and breathlessness can occur when the heart becomes too weak to circulate blood. Some heart conditions occur with no symptoms at all, especially in older adults and individuals with diabetes. The term 'congenital heart disease' covers a range of conditions, but the general symptoms include sweating, high levels of fatigue, fast heartbeat and breathing, breathlessness, chest pain. However, these symptoms might not develop until a person is older than 13 years. In these types of cases, the diagnosis becomes an intricate task requiring great experience and high skill. A risk of a heart attack or the possibility of heart disease if identified early, can help the patients take precautions and take regulatory measures. Recently, the healthcare industry has been generating huge amounts of data about patients and their disease diagnosis reports are being especially taken for the prediction of heart attacks worldwide. When the data about heart disease is huge, the machine learning techniques can be implemented for the analysis.

1.2.1. What is Heart Disease?

Heart disease describes a range of conditions that affect your heart. Diseases under the heart disease umbrella include blood vessel diseases, such as coronary artery disease; heart rhythm problems (arrhythmias); and heart defects you're born with (congenital heart defects), among others. The term "heart disease" is often used interchangeably with the term "cardiovascular disease." Cardiovascular disease generally refers to conditions that involve narrowed or blocked blood vessels that can lead to a heart attack, chest pain (angina) or stroke. Other heart conditions, such as those that affect your heart's muscle, valves or rhythm, also are considered forms of heart disease. Heart failure is a serious condition with high prevalence (about 2% in the adult population in developed countries, and or than8%inpatients older than 75years). About 3 – 5% of hospital admissions are linked with heart failure incidents. Heart failure is the first cause of admission by healthcare professionals in their clinical practice. The costs are very high, reaching up to 2% of the total health costs in the developed countries. Building an effective disease management strategy requires analysis of large amounts of data, early detection of the disease, assessment of the severity and early prediction of adverse events. This will inhibit the progression of the

disease, will improve the quality of life of the patients and will reduce the associated medical costs.

1.2.1.1. Risk factors

Risk factors for developing heart disease include:

- **Age:** Aging increases your risk of damaged and narrowed arteries and weakened or thickened heart muscle.
- **Sex:** Men are generally at greater risk of heart disease. However, women's risk increases after menopause.
- **Family history:** A family history of heart disease increases your risk of coronary artery disease, especially if a parent developed it at an early age (before age 55 for a male relative, such as your brother or father, and 65 for a female relative, such as your mother or sister).
- **Smoking:** Nicotine constricts your blood vessels, and carbon monoxide can damage their inner lining, making them more susceptible to atherosclerosis. Heart attacks are more common in smokers than in nonsmokers.
- **Certain chemotherapy drugs and radiation therapy for cancer:** Some chemotherapy drugs and radiation therapies may increase the risk of cardiovascular disease.
- **Poor diet:** A diet that's high in fat, salt, sugar and cholesterol can contribute to the development of heart disease.
- **High blood pressure:** Uncontrolled high blood pressure can result in hardening and thickening of your arteries, narrowing the vessels through which blood flows.
- **High blood cholesterol levels:** High levels of cholesterol in your blood can increase the risk of formation of plaques and atherosclerosis.
- **Diabetes:** Diabetes increases your risk of heart disease. Both conditions share similar risk factors, such as obesity and high blood pressure.
- **Obesity:** Excess weight typically worsens other risk factors.
- **Physical inactivity:** Lack of exercise also is associated with many forms of heart

disease and some of its other risk factors, as well.

- **Stress:** Unrelieved stress may damage your arteries and worsen other risk factors for heart disease.

- **Poor hygiene:** Not regularly washing your hands and not establishing other habits that can help prevent viral or bacterial infections can put you at risk of heart infections, especially if you already have an underlying heart condition. Poor dental health also may contribute to heart disease.

1.2.1.2. Complications

Complications of heart disease include:

- **Heart failure:** One of the most common complications of heart disease, heart failure occurs when your heart can't pump enough blood to meet your body's needs. Heart failure can result from many forms of heart disease, including heart defects, cardiovascular disease, valvular heart disease, heart infections or cardiomyopathy.

- **Heart attack:** A blood clot blocking the blood flow through a blood vessel that feeds the heart causes a heart attack, possibly damaging or destroying a part of the heart muscle. Atherosclerosis can cause a heart attack.

- **Stroke:** The risk factors that lead to cardiovascular disease also can lead to an ischemic stroke, which happens when the arteries to your brain are narrowed or blocked so that too little blood reaches your brain. A stroke is a medical emergency brain tissue begins to die within just a few minutes of a stroke.

- **Aneurysm:** A serious complication that can occur anywhere in your body, an aneurysm is a bulge in the wall of your artery. If an aneurysm bursts, you may face life-threatening internal bleeding.

- **Peripheral artery disease:** Atherosclerosis also can lead to peripheral artery disease. When you develop peripheral artery disease, your extremities usually your legs don't receive enough blood flow. This causes symptoms, most notably leg pain when walking (claudication).

- **Sudden cardiac arrest:** Sudden cardiac arrest is the sudden, unexpected loss of

heart function, breathing and consciousness, often caused by an arrhythmia. Sudden cardiac arrest is a medical emergency. If not treated immediately, it is fatal, resulting in sudden cardiac death.

1.2.2. Features

Some of the attributes we used for Heart Disease Prediction and their correlation to CVD (Cardiovascular Diseases). This dataset consists of 13 features and a target variable. The detailed description of all the features are as follows:

- **Age:** Patients Age in years. (Numeric)
- **Sex:** Gender of patient. (Male - 1, Female - 0) (Nominal)
- **Chest Pain Type:** Type of chest pain experienced by patient categorized into:
  - Value 1: Typical angina
  - Value 2: Atypical angina
  - Value 3: Non-anginal pain
  - Value 4: Asymptomatic

  (Angina: Angina is caused when there is not enough oxygen-rich blood flowing to a certain part of the heart. The arteries of the heart become narrow due to fatty deposits in the artery walls. The narrowing of arteries means that blood supply to the heart is reduced, causing angina.) (Nominal)
- **Resting bps:** Level of blood pressure at resting mode in mm/HG. (Numerical)
- **Cholesterol:** Serum cholesterol in mg/dl. (Numeric) (Cholesterol means the blockage for blood supply in the blood vessels.)
- **Fasting blood sugar:** Blood sugar levels on fasting > 120 mg/dl represents as 1 in case of true and 0 as false. (Nominal) (blood sugar taken after a long gap between a meal and the test. Typically, it's taken before any meal in the morning.)
- **Resting ecg:** Result of electrocardiogram while at rest are represented in 3 distinct values (Nominal):
  - Value 0: Normal
  - Value 1: Having ST-T wave abnormality (T wave inversions and/or ST elevation or depression of > 0.05 mV)

- Value 2: Showing probable or definite left ventricular hypertrophy by Estes' criteria.

(ECG values taken while a person is on rest which means no exercise and normal functioning of heart is happening.)

- **Oldpeak:** Exercise induced ST-depression in comparison with the state of rest. (Numeric) (ST Depression is the difference between the value of ECG at rest and after exercise. An electrocardiogram records the electrical signals in your heart. It's a common and painless test used to quickly detect heart problems and monitor your heart's health.)

- **ST slope:** ST segment measured in terms of slope during peak exercise. (Nominal)
  - Value 1: Upsloping
  - Value 2: Flat
  - Value 3: Downsloping

- **ca:** Number of major blood vessels. (0-3)(Numeric)
  (Fluoroscopy is an imaging technique that uses X-rays to obtain real-time moving images of the interior of an object. In its primary application of medical imaging, a fluoroscope allows a physician to see the internal structure and function of a patient, so that the pumping action of the heart or the motion of swallowing, for example, can be watched.)

- **Exang:** Exercise induced angina. (1 = yes; 0 = no)(Nominal)
  (Exang is chest pain while exercising or doing any physical activity.)

- **Thal:** Thallium stress test. (Nominal)
  - Value 3: normal
  - Value 6: fixed defect
  - Value 7: reversible defect

- **Thalach:** Maximum heart rate achieved in bpm. (Numeric)

Target Variable:

- **Target:** It is the target variable which we have to predict, 2 means patient is suffering from heart risk and 1 means patient is normal. (1 = no disease; 2 = disease)

1.3. PROBLEM STATEMENT

Heart disease can be managed effectively with a combination of lifestyle changes, medicine and, in some cases, surgery. With the right treatment, the symptoms of heart disease can be reduced and the functioning of the heart improved. The predicted results can be used to prevent and thus reduce cost for surgical treatment and other expenses.

The overall objective of this internship project is to predict accurately with few tests and attribute the presence of heart disease. Attributes considered form the primary basis for tests and give accurate results more or less. Many more input attributes can be taken but our goal is to predict with few attributes and faster efficiency the risk of having heart disease. Decisions are often made based on doctors' intuition and experience rather than on the knowledge rich data hidden in the data set and databases. This practice leads to unwanted biases, errors and excessive medical costs which affects the quality of service provided to patients.

1.3.1. Existing System

1.3.1.1. Prediction system for heart disease using Naive Bayes and particle swarm

This section provides the basic concepts of classifiers as Naive Bayes and features subset selection methods as PSO.

1.3.1.1.1. Particle swarm optimization (PSO)

PSO is an Evolutionary Computation technique proposed by Kennedy et al. in 1995. PSO is motivated by social behaviors such as bird flocking and fish schooling. In PSO the population swarm consists of "n" particles, and the position of each particle stands for the potential solution in D-dimensional space. The particles change its condition based on three aspects: To keep its inertia; To change the condition according to its most optimist position; To change the condition according to the swarm's most optimist position[1]. In PSO, a population is encoded as particles in the search space dimensionality D. PSO starts with the random initialization of a population of particles. Based on the best experience of one particle (pbest) and its neighboring particles (gbest), PSO searches for the optimal solution by updating the velocity and the position of each

particle; PSO is used as feature subset selection method due to its advantages:

- Simple and easy to implement.
- Continuous optimization approach.

1.3.1.1.2. Naïve Bayes' Classifier

Naive Bayes classifiers are a family of simple probabilistic classifiers based on using Bayes theorem with strong (Naive) independence assumptions between the features. Naive Bayes classifiers are highly scalable by requiring several parameters linear for the number of features or predictors as variables in a learning problem.[2] It is the simplest and the fastest probabilistic classifier especially for the training phase.

Feature selection - It is a process of removing the irrelevant and redundant features from the dataset based on evaluation criteria which is used to improve accuracy. There are two approaches: individual evaluation and another one is subset evaluation. The process of feature selection is classified into three broad classes. One is filter and another one is wrapper and third one is embedded method based on how the feature selection is deployed by supervised learning algorithm. In this paper, they proposed a model which uses Naive Bayes as classifier and PSO as Feature subset selection measure for prediction of heart disease.[3]

Proposed system - In this section, we propose a methodology to improve the performance of Bayesian classifiers for prediction of heart disease. Algorithm for our proposed model is shown below:

Algorithm 1: Heart disease prediction by using Bayes classifier and PSO.

Input: Heart disease dataset.
Output: Classify patient dataset into heart disease or not (normal).
Step 1: Read the dataset.
Step 2: Apply particle swarm optimization for feature selection.
Step 3: Remove the features with low value of PSO.
Step 4: Apply Naive Bayes classifier on relevant features.
Step 5: Evaluate the performance of the NB+PSO model.

The above algorithm divided into two sections, section 1 (step 2 and step 3) performs processing and feature subset selection. In section 2 (step 4 and step 5) Naive Bayes is applied on relevant features data and evaluate the performance in terms of accuracy.[4]

Accuracy = (No. of objects correctly classified/Total no. of objects in test set) Cross validation technique used to split into training and testing data.

1.3.2. Literature Review

According to Ordonez [5] the heart disease can be predicted with some basic attributes taken from the patient and in their work have introduced a system that includes the characteristics of an individual human being based on totally 13 basic attributes like sex, blood pressure, cholesterol and others to predict the likelihood of a patient getting affected by heart disease. They have added two more attributes i.e. fat and smoking behavior and extended the research dataset. The data mining classification algorithms such as Decision Tree, Naive Bayes, and Neural Network are utilized to make predictions and the results are analyzed on the Heart disease database.

Yilmaz, [6] have proposed a method that uses least squares support vector machine (LS-SVM) utilizing a binary decision tree for classification of cardiotocogram to find out the patient condition.

Duff, et al. [7] have done research work involving five hundred and thirty-three patients who had suffered from cardiac arrest and they were integrated in the analysis of heart disease probabilities. They performed classical statistical analysis and data mining analysis using mostly Bayesian networks.

Frawley, et al. [8] have performed work on prediction of survival of Coronary heart disease (CHD) which is a challenging research problem for medical society. They also used 10-fold cross-validation methods to determine the impartial estimate of the three prediction models for performance comparison purposes.

Noh et al. [9] suggested a classification method which is an associative classifier that is constructed based on the efficient FP-growth method. Because the volume of patterns can be diverse and huge, they offered a rule to measure the cohesion and in turn allow a tough choice of pruning patterns in the pattern-generating process.

1.3.3. Proposed Solution

In this internship project we have used 4 algorithms to predict if the person is suffering from heart disease or not and create a model to get the maximum accuracy possible. For this we have used UCI dataset.[10]

From this dataset we have used 4 widely used algorithms Random forest, K-Nearest Neighbors, Decision Tree and Naive Bayes to create the model with the maximum accuracy possible. We have also explored precision score, recall score, F-score, false negative using confusion matrix for every algorithm used.

The user input is received by the web application using HTML forms. The Web application makes use of HTTP modules to send and receive the data to the API's.

Trained Random forest, K-Nearest Neighbors, Decision Tree and Naive Bayes Models in the form of pickle files are consumed by the flask file housing at the local system.

- The trained models are called by passing the user input JSON object.
- Prediction result is sent as a response to the API calls.

Trained Random forest, K-Nearest Neighbors, Decision Tree and Naive Bayes Models were used to predict whether a person is suffering from Heart Disease or not.

Pre-processed standard datasets were used to train the models post normalizing the dataset using Standard Scalar.

Post training the models, these models were extracted as pickle files and are stored at a local location which is used by the flask framework to call the trained model by passing in the user input.

The API was developed to predict heart disease, using the flask framework this API was hosted locally which will be consumed by the Front-End of the Heart Prediction Engine.

CHAPTER 2

REQUIREMENT ANALYSIS, TOOLS & TECHNOLOGIES

2.1. Hardware & Software Requirements

2.1.1. Hardware Requirements

- **Processors:** Intel® Core™i3 or i5 processor, 4/8 GB of RAM.
- **Operating systems:** Windows* 7 or later, macOS, and Linux.

2.1.2. Software Requirements

- **Python* versions:** 2.7.X, 3.6.X, 3.9.X
- **Included development tools:** conda*, conda-env, Jupyter Notebook* (IPython), Google Colaboratory.
- **Compatible tools:** Microsoft Visual Studio*, PyCharm*.
- **Included Python packages:** NumPy, SciPy, scikit-learn*, pandas, Matplotlib, Seaborn, Numba*, Intel® Threading Building Blocks, pyDAAL, Jupyter, mpi4py, PIP*, and others.
- **Flask Framework:** API Development.

2.2. Tools/ Languages/ Platform

- **Tools:** Flask, conda*, conda-env
- **Languages:** HTML, Python
- **Platform:** Google Colaboratory, Microsoft Visual Studio

CHAPTER 3

DESIGN AND IMPLEMENTATION

3.1. Architecture Model

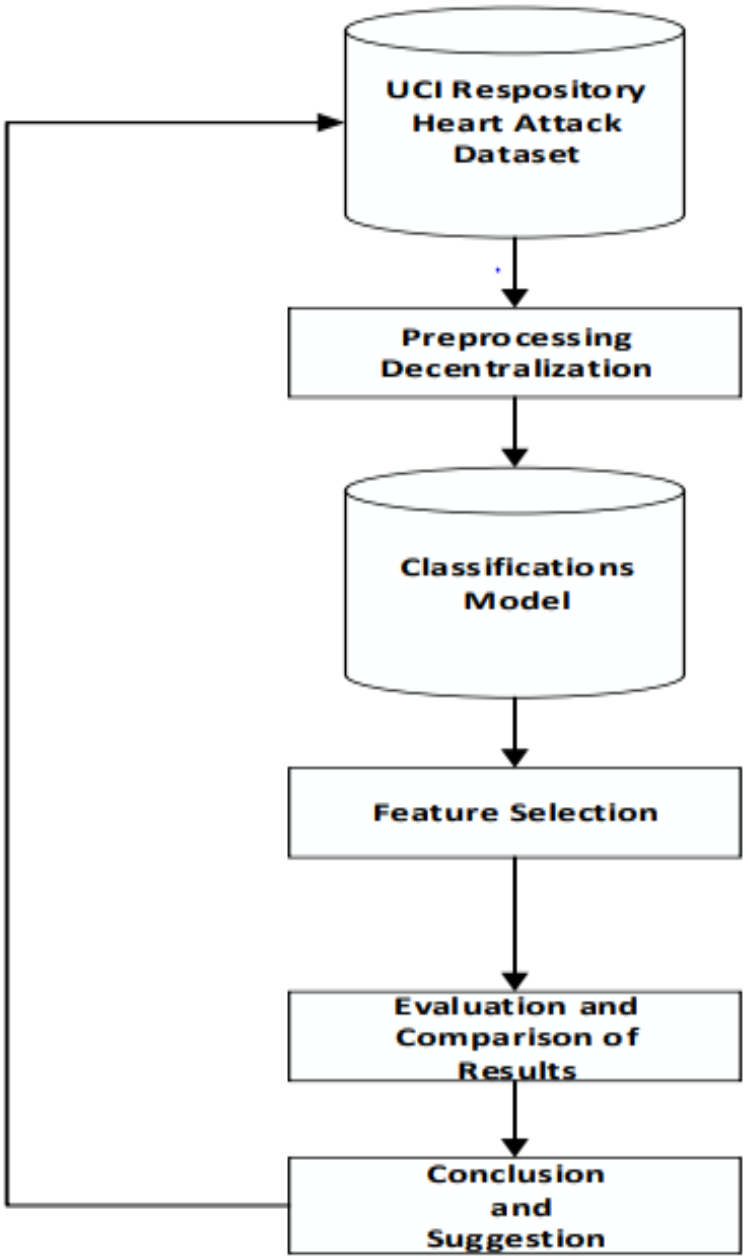

Fig: 3.1 Proposed Model

3.2. Flowchart

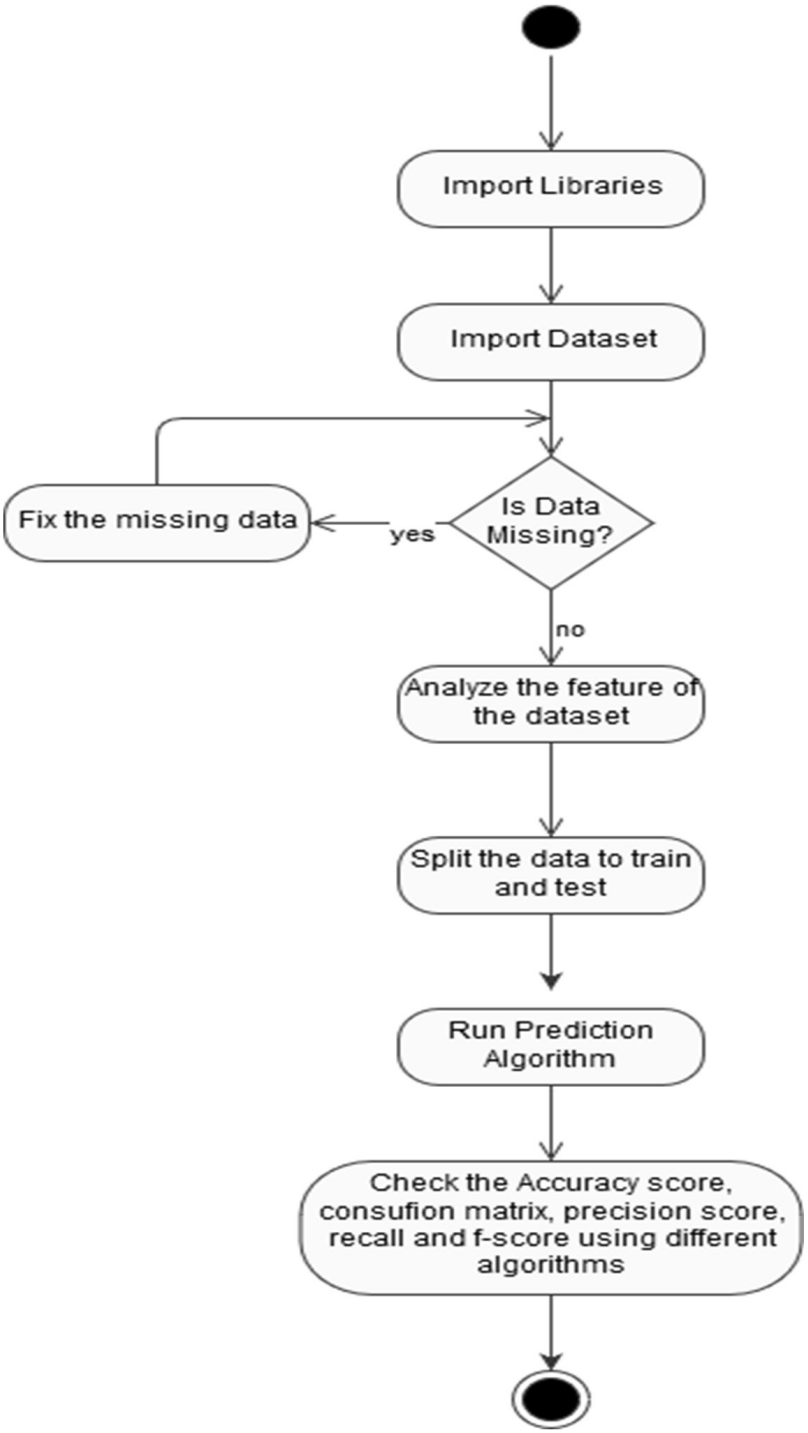

Fig: 3.2 Activity Diagram

3.3. Sequence diagram

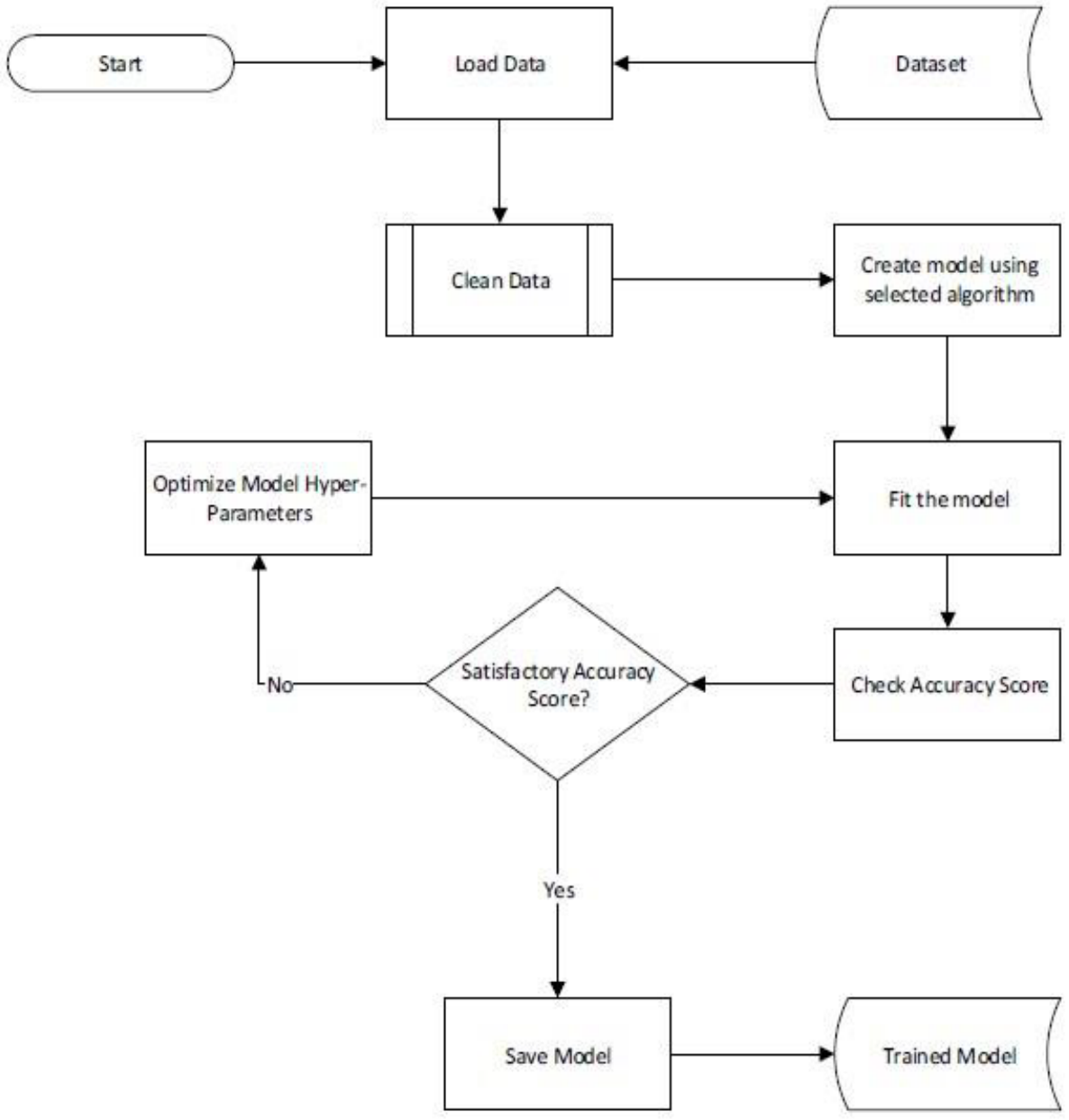

Fig: 3.3 Training Model Process

3.4. Algorithm & Code Segment

3.4.1. Random Forest

Random Forest is a supervised learning algorithm. Random forest can be used for both classification and regression problems, by using random forest regressor we can use random forest on regression problems. But we have used random forest on classification in this internship project so we will only consider the classification part.

3.4.1.1. Random Forest pseudocode

- Randomly select **"k"** features from total **"m"** features. Where **k << m**
- Among the **"k"** features, calculate the node **"d"** using the best split point.
- Split the node into **daughter nodes** using the **best split**.
- Repeat **1 to 3** steps until the "l" number of nodes has been reached.
- Build forest by repeating steps **1 to 4** for "n" number times to create **"n" number of trees**.

3.4.1.2. Random Forest prediction pseudocode

- Takes the **test features** and use the rules of each randomly created decision tree to predict the outcome and stores the predicted outcome (target).
- Calculate the **votes** for each predicted target.
- Consider the **highly voted** predicted target as the **final prediction** from the random forest algorithm.

Code:

```
max_accuracy = 0
for x in range(500):
    rf_classifier = RandomForestClassifier(random_state=x)
    rf_classifier.fit(X_train,Y_train)
    Y_pred_rf = rf_classifier.predict(X_test)
    current_accuracy =
round(accuracy_score(Y_pred_rf,Y_test)*100,2)
    if(current_accuracy>max_accuracy):
        max_accuracy = current_accuracy
        best_x = x
print(max_accuracy)
print(best_x)
rf_classifier =
RandomForestClassifier(random_state=best_x)
rf_classifier.fit(X_train,Y_train)
Y_pred_rf = rf_classifier.predict(X_test)
```

```
Y_pred_rf.shape
score_rf = round(accuracy_score(Y_pred_rf,Y_test)*100,2)
score_rf
```

3.4.2. K-Nearest Neighbors

We can implement a KNN model by following the below steps:

- Load the data
- Initialize the value of k
- For getting the predicted class, iterate from 1 to total number of training data points
  - Calculate the distance between test data and each row of training data. Here we will use Euclidean distance as our distance metric since it's the most popular method. The other metrics that can be used are Chebyshev, cosine, etc.
  - Sort the calculated distances in ascending order based on distance values
  - Get top k rows from the sorted array
  - Get the most frequent class of these rows
  - Return the predicted class

Code:

```
knn_classifier=
KNeighborsClassifier(n_neighbors=31,leaf_size=30)
knn_classifier.fit(X_train,Y_train)
Y_pred_knn = knn_classifier.predict(X_test)
score_knn = round(accuracy_score(Y_pred_knn,Y_test)*100,2)
score_knn
```

3.4.3. Decision Tree

Pseudocode:

- Place the best attribute of the dataset at the **root** of the tree.
- Split the training set into **subsets**. Subsets should be made in such a way that each subset contains data with the same value for an attribute.
- Repeat step 1 and step 2 on each subset until you find **leaf nodes** in all the branches of the tree.

Assumptions while creating a Decision Tree- At the beginning, the whole training set is considered as the root. Feature values are preferred to be categorical. If the values are continuous then they are discretized prior to building the model. Records are distributed recursively on the basis of attribute values. Order to place attributes as root or internal

node of the tree is done by using some statistical approach.

The popular attribute selection measures:

- Information gain
- Gini index

Attribute selection method- A dataset consists of **"n"** attributes then deciding which attribute to place at the root or at different levels of the tree as internal nodes is a complicated step. By just randomly selecting any node to be the root can't solve the issue. If we follow a random approach, it may give us bad results with low accuracy. To solve this attribute selection problem, researchers worked and devised some solutions. They suggested using some criterion like **information gain, Gini index,** etc. These criterions will calculate values for every attribute. The values are sorted, and attributes are placed in the tree by following the order i.e., the attribute with a high value (in case of information gain) is placed at the root. While using information Gain as a criterion, we assume attributes to be categorical, and for Gini index, attributes are assumed to be continuous. [11]

Gini Index - Gini Index is a metric to measure how often a randomly chosen element would be incorrectly identified. It means an attribute with a lower Gini index should be preferred.

$$Gini = 1 - \sum_{i=1}^{C} (p_i)^2$$

Code:

```
dt_classifier = DecisionTreeClassifier(
    max_depth=20,
    min_samples_split=2,
    min_samples_leaf=1,
    min_weight_fraction_leaf=0.00001,
    max_features='auto',
    random_state=46)
dt_classifier.fit(X_train, Y_train)
Y_pred_dt=dt_classifier.predict(X_test)
score_dt = round(accuracy_score(Y_pred_dt,Y_test)*100,2)
score_dt
```

3.4.4. Naïve Bayes

Bayes' Theorem is stated as:

$$P(h|d) = (P(d|h) * P(h)) / P(d)$$

P(h|d) is the probability of hypothesis h given the data d. This is called the posterior probability.

P(d|h) is the probability of data d given that the hypothesis h was true.

P(h) is the probability of hypothesis h being true (regardless of the data). This is called the prior probability of h.

P(d) is the probability of the data (regardless of the hypothesis).

We are interested in calculating the posterior probability of P(h|d) from the prior probability p(h) with P(D) and P(d|h). After calculating the posterior probability for a number of different hypotheses, we will select the hypothesis with the highest probability. This is the maximum probable hypothesis and may formally be called the (MAP) hypothesis.

This can be written as:

$$MAP(h) = max(P(h|d)) \text{ or}$$

$$MAP(h) = max((P(d|h) * P(h)) / P(d)) \text{ or}$$

$$MAP(h) = max(P(d|h) * P(h))$$

The P(d) is a normalizing term which allows us to calculate the probability. We can drop it when we are interested in the most probable hypothesis as it is constant and only used to normalize. Back to classification, if we have an even number of instances in each class in our training data, then the probability of each class (e.g. P(h)) will be equal. Again, this would be a constant term in our equation, and we could drop it so that we end up with:

$$MAP(h) = max(P(d|h))$$

Naive Bayes is a classification algorithm for binary (two-class) and multi-class classification problems. The technique is easiest to understand when described using binary or categorical input values. It is called Naive Bayes or Idiot Bayes because the calculation of the probabilities for each hypothesis are simplified to make their calculation tractable. Rather than attempting to calculate the values of each attribute value P (d1, d2, d3|h), they are assumed to be conditionally independent given the

target value and calculated as P(d1|h) * P(d2|H) and so on. This is a very strong assumption that is most unlikely in real data, i.e. that the attributes do not interact. Nevertheless, the approach performs surprisingly well on data where this assumption does not hold.

MAP(h) = max(P(d|h) * P(h))

Gaussian Naïve Bayes:

mean(x) = 1/n * sum(x)

Where n is the number of instances and x are the values for an input variable in your training data. We can calculate the standard deviation using the following equation:

standard deviation(x) = sqrt (1/n * sum(xi-mean(x)^2))

This is the square root of the average squared difference of each value of x from the mean value of x, where n is the number of instances, sqrt() is the square root function, sum() is the sum function, xi is a specific value of the x variable for the i'th instance and mean(x) is described above, and ^2 is the square. Gaussian PDF with a new input for the variable, and in return the Gaussian PDF will provide an estimate of the probability of that new input value for that class.

pdf (x, mean, sd) = (1 / (sqrt (2 * PI) * sd)) * exp (-((x-mean^2)/(2*sd^2)))

Where pdf(x) is the Gaussian Probability Density Function (PDF), sqrt () is the square root, mean and sd are the mean and standard deviation calculated above, Pi is the numerical constant, exp () is the numerical constant e or Euler's number raised to power and x is the input value for the input variable.

Code:

```
nb_classifier = GaussianNB( var_smoothing=1e-50)
nb_classifier.fit(X_train,Y_train)
nb_classifier.predict(X_test)
Y_pred_nb = nb_classifier.predict(X_test)
score_nb = round(accuracy_score(Y_pred_nb,Y_test)*100,2)
score_nb
```

3.4.5. Web App Code

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

# Load ML model
model = pickle.load(open('models.pkl', 'rb'))

# Create application
app = Flask(__name__)

# Bind home function to URL
@app.route('/')
def home():
    return
render_template('Heart_Disease_Classifier.html')

# Bind predict function to URL
@app.route('/predict', methods =['POST'])
def predict():

    # Put all form entries values in a list
    features = [float(i) for i in request.form.values()]
    # Convert features to array
    array_features = [np.array(features)]
    # Predict features
    prediction = model.predict(array_features)

    output = prediction

    # Check the output values and retrieve the result with
html tag based on the value
    if output == 1:
        return
render_template('Heart_Disease_Classifier.html',
                                result = 'The patient is
not likely to have heart disease!')
    else:
        return
render_template('Heart_Disease_Classifier.html',
                                result = 'The patient is
likely to have heart disease!')

if __name__ == '__main__':
#Run the application
    app.run()
```

3.5. Libraries used

Python has a vast reserve of inbuilt standard libraries which includes areas like web services tools, string operation, data analysis, and machine learning, etc. The complex programming tasks can be dealt with ease using these inbuilt libraries as it reduces the size of code with many inbuilt functions that do the job pretty well for its user.

3.5.1. Data Visualization

- **Matplotlib:**

  Matplotlib is a cross-platform, data visualization and graphical plotting library for Python and its numerical mathematics extension NumPy, a big data numerical handling resource.

  - pyplot
  - rcParams
  - rainbow

- **Seaborn:**

  Seaborn is an open-source Python library built on top of matplotlib. It is used for data visualization and exploratory data analysis. Seaborn works easily with dataframes and the Pandas library. The graphs created can also be customized easily.

3.5.2. Data Manipulation

- **NumPy:** The NumPy library in python is used for scientific computing and array manipulation. It can perform different operations such as indexing of an array, sequencing, and slicing, etc.

- **Pandas:** The Pandas library in python is used for structuring, manipulating, and organizing data in a tabular structure called the data frame which is further used for data analysis.

- **Scikit-learn:**
  - sklearn.model_selection
    - train_test_split
  - sklearn.preprocessing
    - StandardScaler
    - LabelEncoder

3.5.3. Data Modeling

- **Scikit-learn:**

  Scikit-learn is one of the most useful libraries that python offers. It has various statistical learning algorithms such as regression models (linear regression, logistic regression), SVM's, random forest for classification tasks and k-means for clustering, etc.

  - sklearn.ensemble.RandomForestClassifier
  - sklearn.neighbors.KNeighborsClassifier
  - sklearn.tree.DecisionTreeClassifier
  - sklearn.naive_bayes.GaussianNB

3.5.4. Data Validation

- **Scikit-learn-metrics:**

  The sklearn.metrics module implements several loss, score, and utility functions to measure classification performance.

  sklearn.metrics -

  log_loss, roc_auc_score, precision_score, f1_score, recall_score, roc_curve, auc, plot_roc_curve, classification_report, confusion_matrix, accuracy_score, fbeta_score, matthews_corrcoef

- **Mlxtend:**

  Mlxtend (machine learning extensions) is a Python library of useful tools for day-to-day data science tasks.

  mlxtend.plotting -

  plot_confusion_matrix

CHAPTER 4

OBSERVATIONS AND RESULTS

4.1. Testing

Testing is the process used to help identify the correctness, completeness, security and quality of the developed computer software. Testing is the process of technical investigation and includes the process of executing a program or application with the intent of finding errors.

In the training process, our model learns to associate a particular input (i.e. features) to the corresponding output (tag) based on the test samples used for training. Input features and tags (e.g. 1-normal 2-heart disease) are fed into the machine learning algorithm to generate a model.

A comparative analysis of different classifiers was performed for the classification of the Heart Disease dataset in order to correctly classify and predict Heart Disease cases with minimal attributes.

Input	Expected Output	Actual Output
Data Visualization	Various visual representations of the data to understand more about the relationship between various features.	Pass
Data Processing	Convert some categorical variables into dummy variables and scale all the values before training the Machine Learning models.	Pass

Dataset	Split the dataset into training and testing datasets.	Pass
Training dataset	Train the model using the training dataset.	Pass
Testing dataset	Tests if the model is accurate based on the output of the testing dataset.	Pass

Table 4.1 Training and subsequent testing

Input	**Expected Output**	**Actual Output**
No Heart Disease	Should be labeled as 1 (no heart disease) and should show output as “The patient is not likely to have heart disease”.	Pass
Heart Disease	Should be labeled as 2 (heart disease) and should show output as “The patient is likely to have heart disease”.	Pass

Table 4.2 Heart Disease Test

4.1.1. Model Evaluation

The most important evaluation metrics for this problem domain are Accuracy, Sensitivity, Specificity, Precision, F1-measure,Log Loss, ROC and Mathew correlation coefficient.

- **Accuracy:** which refers to how close a measurement is to the true value and can be calculated using the following formula:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

- **Precision:** which is how consistent results are when measurements are repeated and can be calculated using the following formula:

Precision = True Positive / (True Positive + False Positive)

- **Sensitivity:**

Sensitivity is a measure of the proportion of actual positive cases that got predicted as positive (or true positive). Sensitivity is also termed as Recall.

Sensitivity = True Positive / (True Positive + False Negative)

- **Specificity:**

Specificity is defined as the proportion of actual negatives, which got predicted as the negative (or true negative).

Specificity = True Negative / (True Negative + False Positive)

- **Mathew Correlation coefficient (MCC):**

The Matthews correlation coefficient (MCC), instead, is a more reliable statistical rate which produces a high score only if the prediction obtained good results in all of the four confusion matrix categories (true positives, false negatives, true negatives, and false positives), proportionally both to the size of positive elements and the size of negative elements in the dataset.

$$\mathrm{MCC} = \frac{\mathrm{TP} \cdot \mathrm{TN} - \mathrm{FP} \cdot \mathrm{FN}}{\sqrt{(\mathrm{TP} + \mathrm{FP}) \cdot (\mathrm{TP} + \mathrm{FN}) \cdot (\mathrm{TN} + \mathrm{FP}) \cdot (\mathrm{TN} + \mathrm{FN})}}$$

(worst value: −1; best value: +1)

- **Log Loss:**

  Logarithmic loss measures the performance of a classification model where the prediction input is a probability value between 0 and 1. The goal of our machine learning models is to minimize this value. A perfect model would have a log loss of 0. Log loss increases as the predicted probability diverges from the actual label. So predicting a probability of .012 when the actual observation label is 1 would be bad and result in a high log loss.

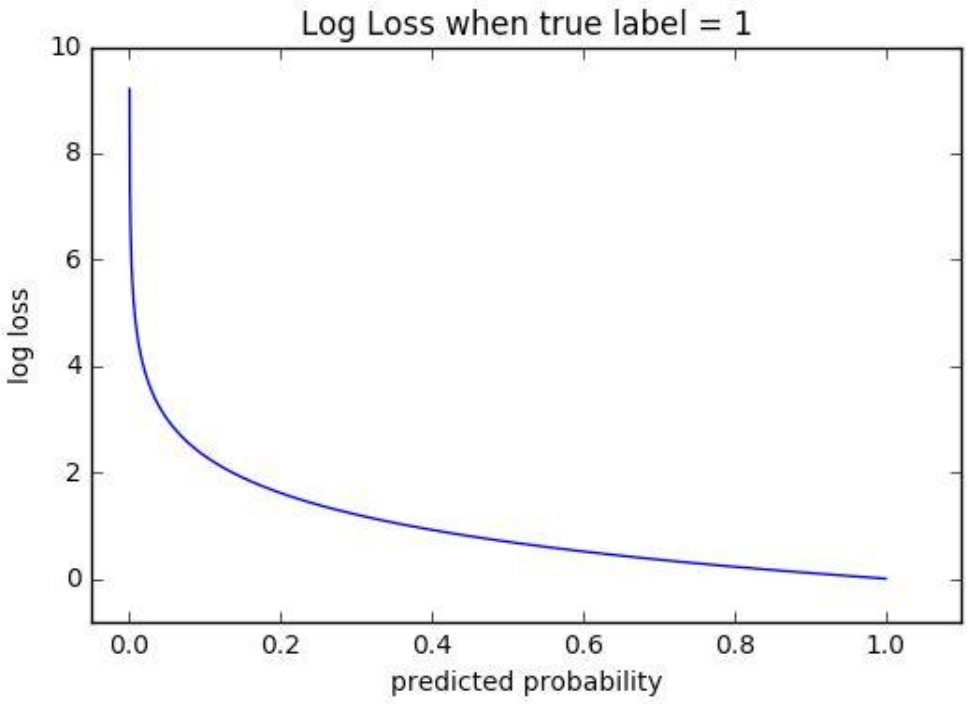

Fig: 4.1 Log Loss Graph

- **F1 Score:**

  F1 Score is the weighted average of Precision and Recall. Therefore, this score takes both false positives and false negatives into account. Intuitively it is not as easy to understand as accuracy, but F1 score is usually more useful than accuracy, especially if you have an uneven class distribution. Accuracy works best if false positives and false negatives have similar cost.

F1 Score = 2(Recall Precision) / (Recall + Precision)

- **ROC Curve:**

  An ROC curve (receiver operating characteristic curve) is a graph showing the performance of a classification model at all classification thresholds. This curve plots two parameters: True Positive Rate & False Positive Rate.

4.1.1.1. Random Forest Classifier

```
y_pred_rfe = rf_classifier.predict(X_test)

plt.figure(figsize=(10, 8))
CM=confusion_matrix(Y_test,y_pred_rfe)
sns.heatmap(CM, annot=True)

TN = CM[0][0]
FN = CM[1][0]
TP = CM[1][1]
FP = CM[0][1]
specificity = TN/(TN+FP)
loss_log = log_loss(Y_test, y_pred_rfe)
acc= accuracy_score(Y_test, y_pred_rfe)
roc=roc_auc_score(Y_test, y_pred_rfe)
prec = precision_score(Y_test, y_pred_rfe)
rec = recall_score(Y_test, y_pred_rfe)
f1 = f1_score(Y_test, y_pred_rfe)
mathew = matthews_corrcoef(Y_test, y_pred_rfe)

model_results =pd.DataFrame([['Random Forest',acc,
prec,rec,specificity, f1,roc, loss_log,mathew]],
                columns = ['Model',
'Accuracy','Precision', 'Sensitivity','Specificity', 'F1
Score','ROC','Log_Loss','mathew_corrcoef'])

model_results
```

	Model	Accuracy	Precision	Sensitivity	Specificity	F1 Score	ROC	Log_Loss	mathew_corrcoef
0	Random Forest	0.851852	0.866667	0.866667	0.866667	0.866667	0.85	19.188653	0.7

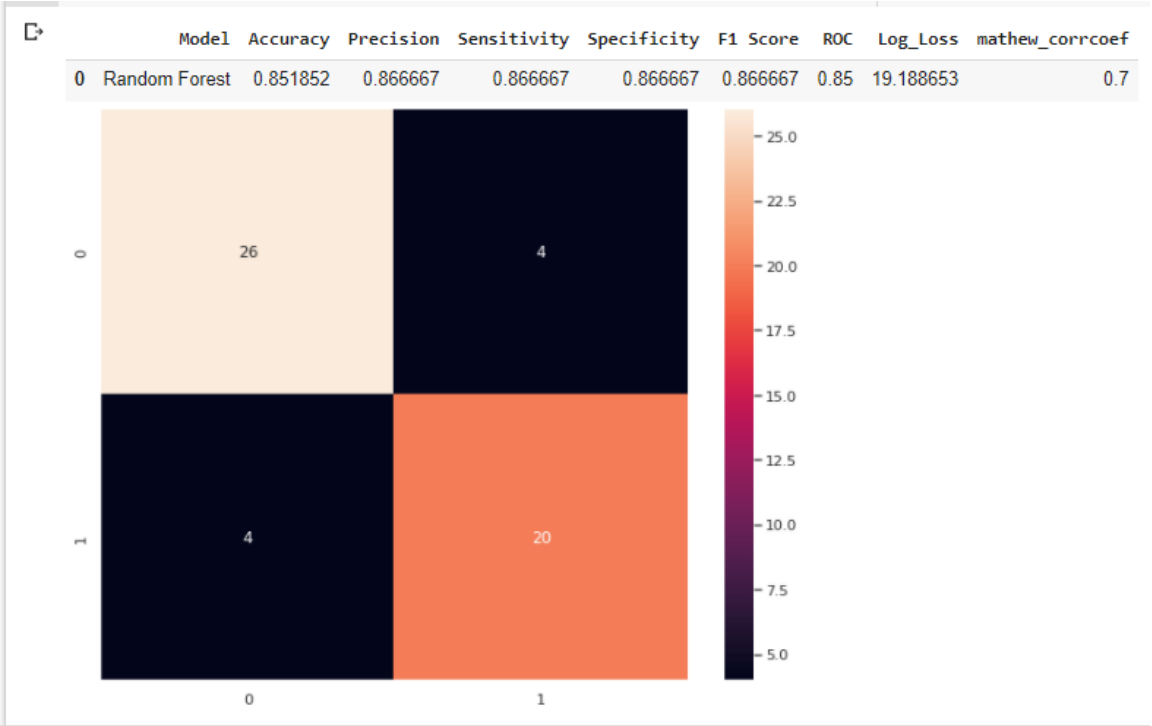

Fig: 4.2 Random Forest Confusion Matrix

```
Y_pred_rf = np.around(Y_pred_rf)
print(metrics.classification_report(Y_test,Y_pred_rf))
```

```
Y_pred_rf = np.around(Y_pred_rf)
print(metrics.classification_report(Y_test,Y_pred_rf))
```

```
              precision    recall  f1-score   support

           1       0.87      0.87      0.87        30
           2       0.83      0.83      0.83        24

    accuracy                           0.85        54
   macro avg       0.85      0.85      0.85        54
weighted avg       0.85      0.85      0.85        54
```

Table 4.3 Random Forest Classification Report

```
plot_roc_curve(rf_classifier,X_test,Y_test)
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver Operating Characteristic Curve')
```

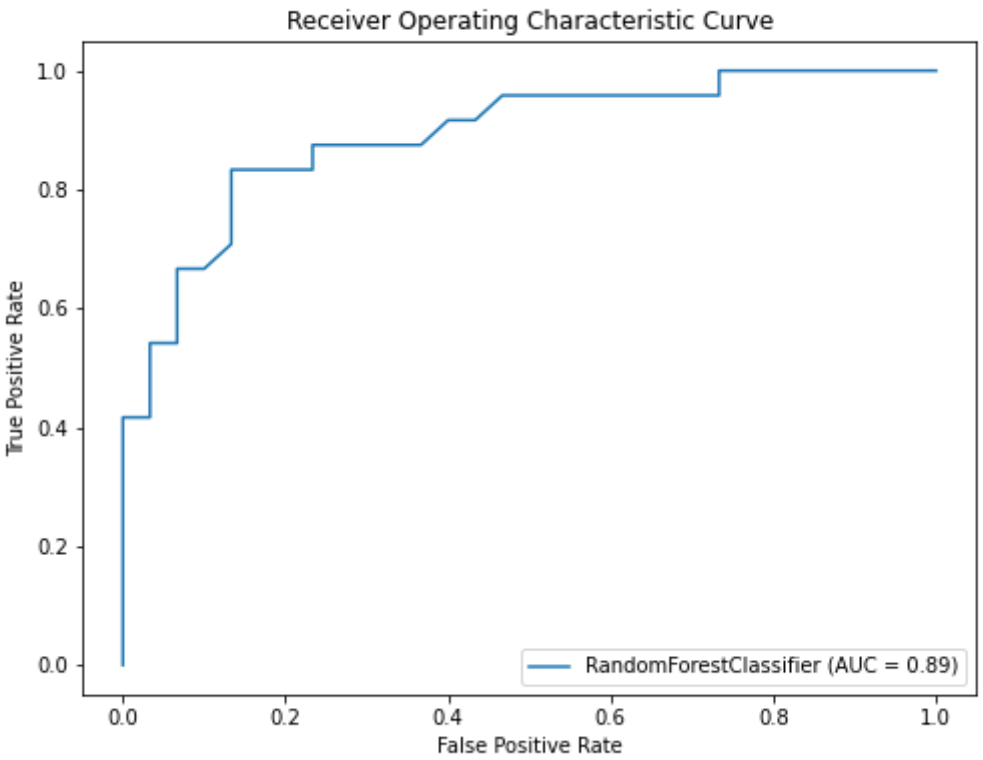

Fig: 4.3 Random Forest ROC Curve

4.1.1.2. K-Nearest Neighbors Classifier

```
y_pred_knne = knn_classifier.predict(X_test)

plt.figure(figsize=(10, 8))
CM=confusion_matrix(Y_test,y_pred_knne)
sns.heatmap(CM, annot=True)

TN = CM[0][0]
FN = CM[1][0]
TP = CM[1][1]
FP = CM[0][1]
specificity = TN/(TN+FP)
loss_log = log_loss(Y_test, y_pred_knne)
acc= accuracy_score(Y_test, y_pred_knne)
roc=roc_auc_score(Y_test, y_pred_knne)
prec = precision_score(Y_test, y_pred_knne)
rec = recall_score(Y_test, y_pred_knne)
f1 = f1_score(Y_test, y_pred_knne)
mathew = matthews_corrcoef(Y_test, y_pred_knne)

model_results =pd.DataFrame([['K-Nearest Neighbors
',acc, prec,rec,specificity, f1,roc, loss_log,mathew]],
                columns = ['Model',
'Accuracy','Precision', 'Sensitivity','Specificity', 'F1
Score','ROC','Log_Loss','mathew_corrcoef'])

model_results
```

	Model	Accuracy	Precision	Sensitivity	Specificity	F1 Score	ROC	Log_Loss	mathew_corrcoef
0	K-Nearest Neighbors	0.796296	0.787879	0.866667	0.866667	0.825397	0.7875	19.188653	0.586094

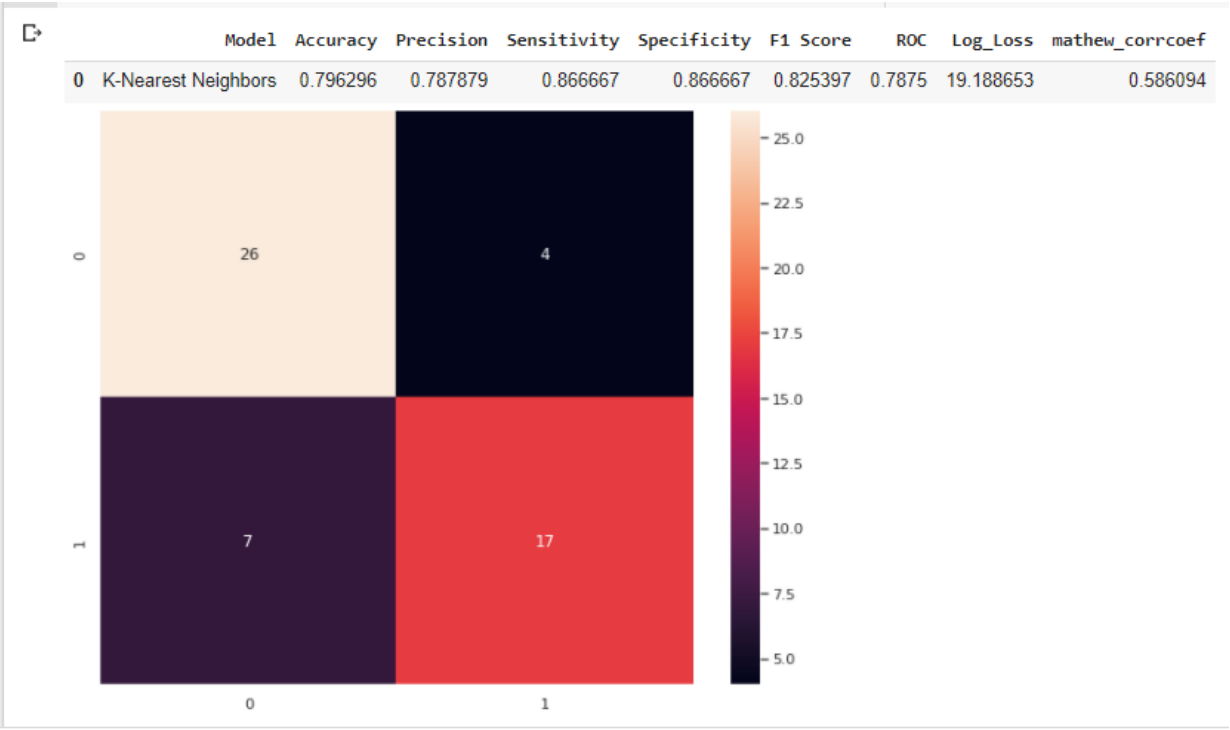

Fig: 4.4 K-Nearest Neighbors Confusion Matrix

```
Y_pred_knn = np.around(Y_pred_knn)
print(metrics.classification_report(Y_test,Y_pred_knn))
```

```
Y_pred_knn = np.around(Y_pred_knn)
print(metrics.classification_report(Y_test,Y_pred_knn))
```

```
              precision    recall  f1-score   support

           1       0.79      0.87      0.83        30
           2       0.81      0.71      0.76        24

    accuracy                           0.80        54
   macro avg       0.80      0.79      0.79        54
weighted avg       0.80      0.80      0.79        54
```

Table 4.4 K-Nearest Neighbors Classification Report

```
plot_roc_curve(knn_classifier,X_test,Y_test)
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver Operating Characteristic Curve')
```

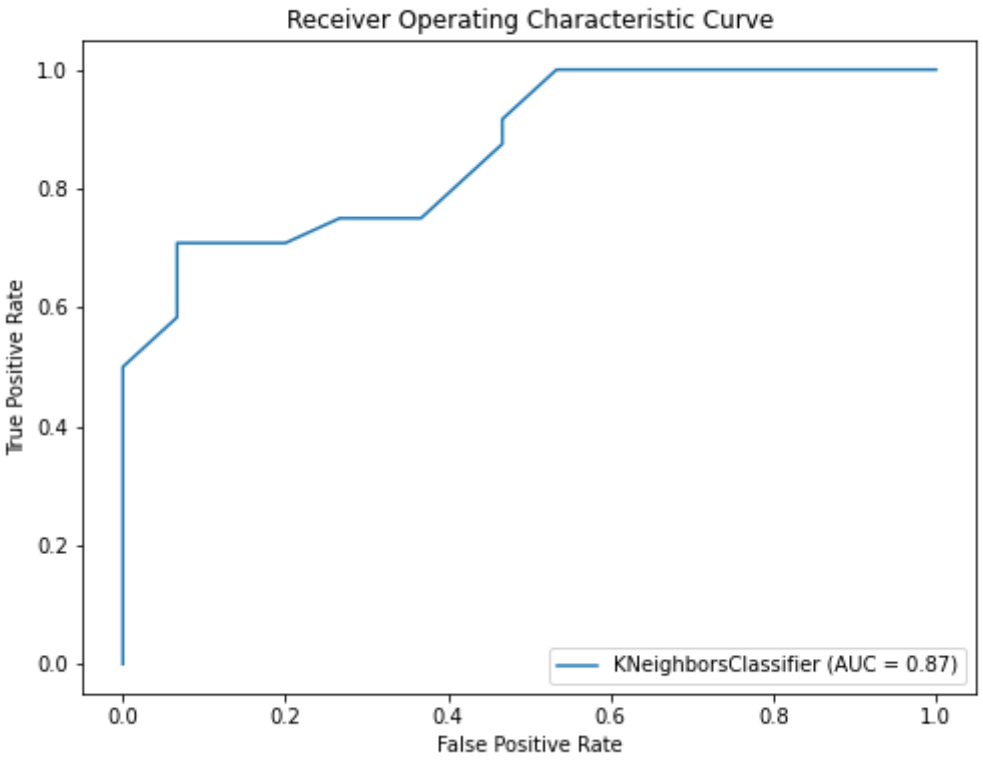

Fig: 4.5 K-Nearest Neighbors ROC Curve

4.1.1.3. Decision Tree Classifier

```
y_pred_dte = dt_classifier.predict(X_test)

plt.figure(figsize=(10, 8))
CM=confusion_matrix(Y_test,y_pred_dte)
sns.heatmap(CM, annot=True)

TN = CM[0][0]
FN = CM[1][0]
TP = CM[1][1]
FP = CM[0][1]
specificity = TN/(TN+FP)
loss_log = log_loss(Y_test, y_pred_dte)
acc= accuracy_score(Y_test, y_pred_dte)
roc=roc_auc_score(Y_test, y_pred_dte)
prec = precision_score(Y_test, y_pred_dte)
rec = recall_score(Y_test, y_pred_dte)
f1 = f1_score(Y_test, y_pred_dte)
mathew = matthews_corrcoef(Y_test, y_pred_dte)

model_results =pd.DataFrame([['Decision Tree',acc,
prec,rec,specificity, f1,roc, loss_log,mathew]],
                columns = ['Model',
'Accuracy','Precision', 'Sensitivity','Specificity', 'F1
Score','ROC','Log_Loss','mathew_corrcoef'])

model_results
```

	Model	Accuracy	Precision	Sensitivity	Specificity	F1 Score	ROC	Log_Loss	mathew_corrcoef
0	Decision Tree	0.703704	0.769231	0.666667	0.666667	0.714286	0.708333	19.188653	0.414371

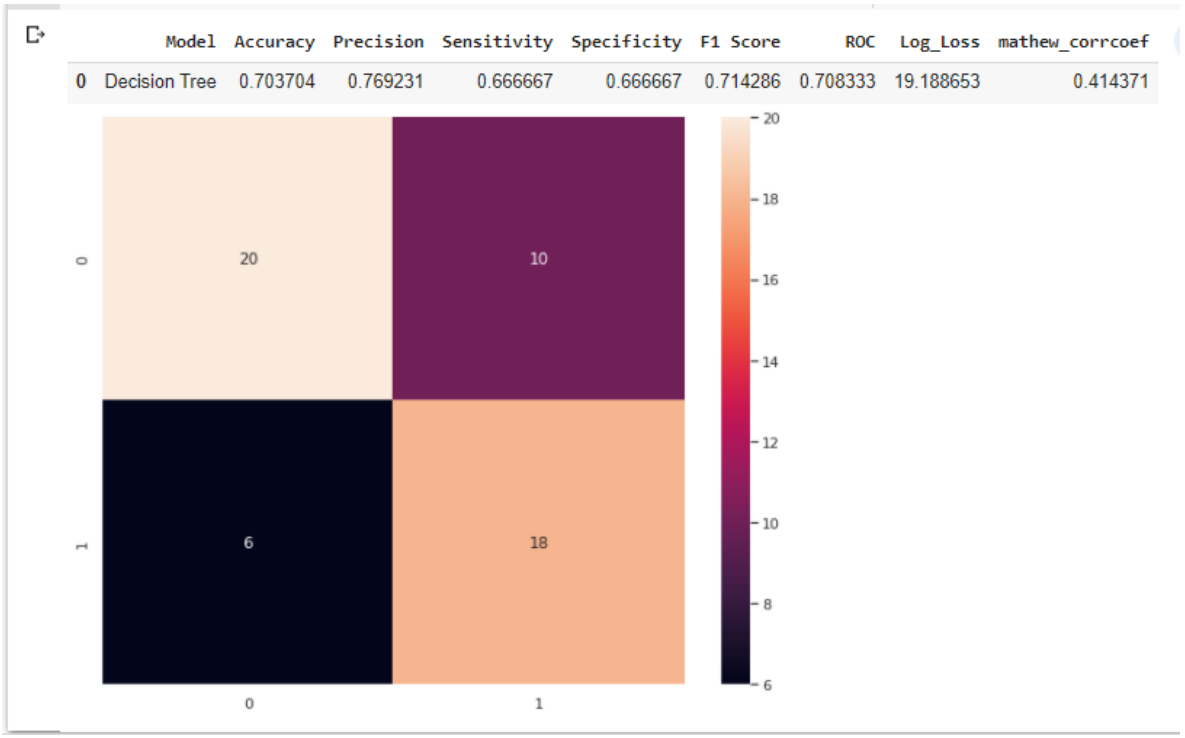

Fig: 4.6 Decision Tree Confusion Matrix

```
Y_pred_dt = np.around(Y_pred_dt)
print(metrics.classification_report(Y_test,Y_pred_dt))
```

```
Y_pred_dt = np.around(Y_pred_dt)
print(metrics.classification_report(Y_test,Y_pred_dt))
```

```
              precision    recall  f1-score   support

           1       0.77      0.67      0.71        30
           2       0.64      0.75      0.69        24

    accuracy                           0.70        54
   macro avg       0.71      0.71      0.70        54
weighted avg       0.71      0.70      0.70        54
```

Table 4.5 Decision Tree Classification Report

```
plot_roc_curve(dt_classifier,X_test,Y_test)
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver Operating Characteristic Curve')
```

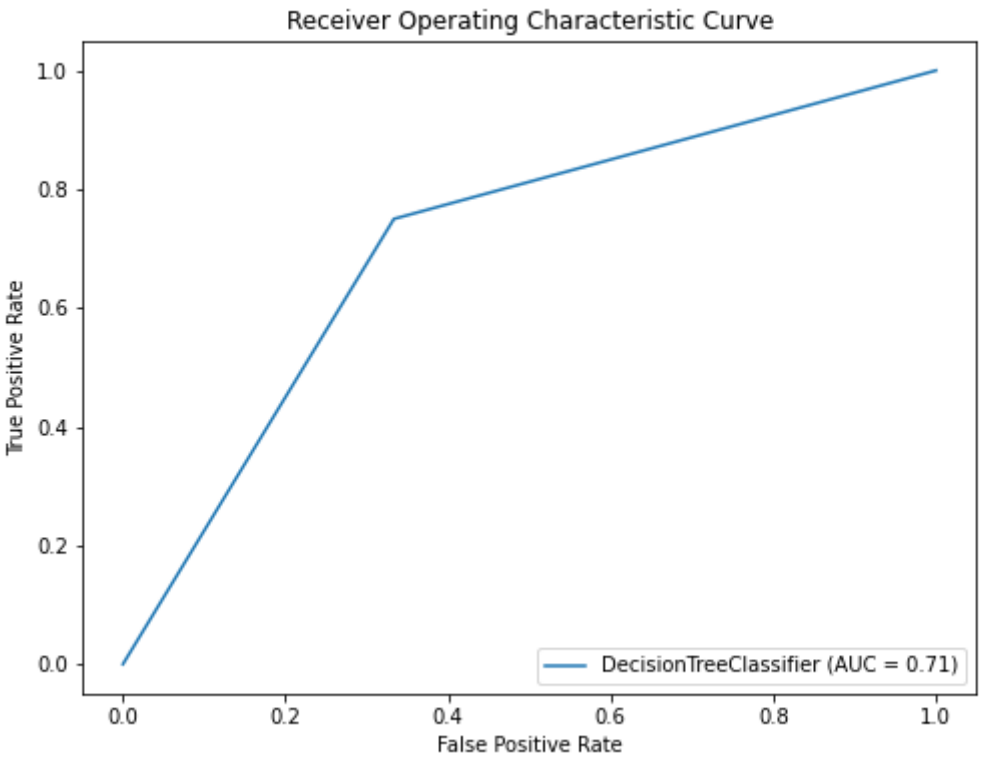

Fig: 4.7 Decision Tree ROC Curve

4.1.1.4. Naive Bayes Classifier

```
y_pred_nbe = nb_classifier.predict(X_test)

plt.figure(figsize=(10, 8))
CM=confusion_matrix(Y_test,y_pred_nbe)
sns.heatmap(CM, annot=True)

TN = CM[0][0]
FN = CM[1][0]
TP = CM[1][1]
FP = CM[0][1]
specificity = TN/(TN+FP)
loss_log = log_loss(Y_test, y_pred_nbe)
acc= accuracy_score(Y_test, y_pred_nbe)
roc=roc_auc_score(Y_test, y_pred_nbe)
prec = precision_score(Y_test, y_pred_nbe)
rec = recall_score(Y_test, y_pred_nbe)
f1 = f1_score(Y_test, y_pred_nbe)
mathew = matthews_corrcoef(Y_test, y_pred_nbe)

model_results =pd.DataFrame([['Naive Bayes ',acc,
prec,rec,specificity, f1,roc, loss_log,mathew]],
                columns = ['Model',
'Accuracy','Precision', 'Sensitivity','Specificity', 'F1
Score','ROC','Log_Loss','mathew_corrcoef'])

model_results
```

	Model	Accuracy	Precision	Sensitivity	Specificity	F1 Score	ROC	Log_Loss	mathew_corrcoef
0	Naive Bayes	0.666667	0.714286	0.666667	0.666667	0.689655	0.666667	19.188653	0.331497

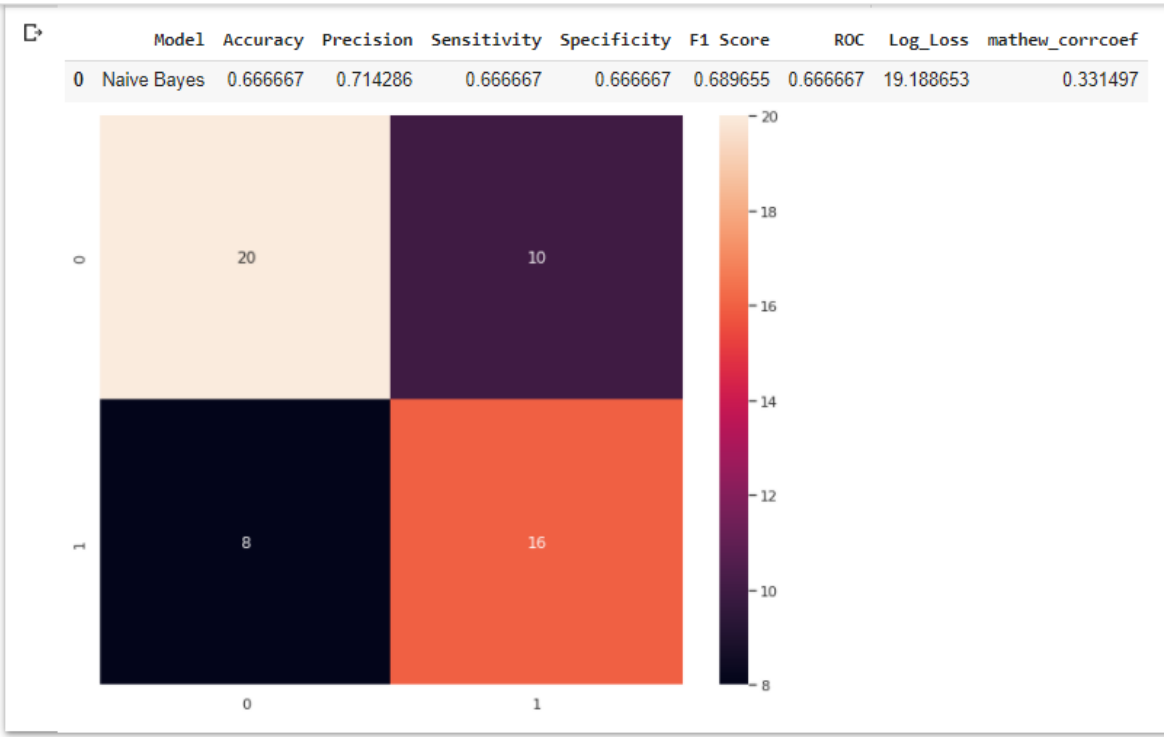

Fig: 4.8 Naive Bayes Confusion Matrix

```
Y_pred_nb = np.around(Y_pred_nb)
print(metrics.classification_report(Y_test,Y_pred_nb))
```

```
Y_pred_nb = np.around(Y_pred_nb)
print(metrics.classification_report(Y_test,Y_pred_nb))
```

```
              precision    recall  f1-score   support

           1       0.71      0.67      0.69        30
           2       0.62      0.67      0.64        24

    accuracy                           0.67        54
   macro avg       0.66      0.67      0.66        54
weighted avg       0.67      0.67      0.67        54
```

Table 4.6 Naive Bayes Classification Report

```
plot_roc_curve(nb_classifier,X_test,Y_test)
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver Operating Characteristic Curve')
```

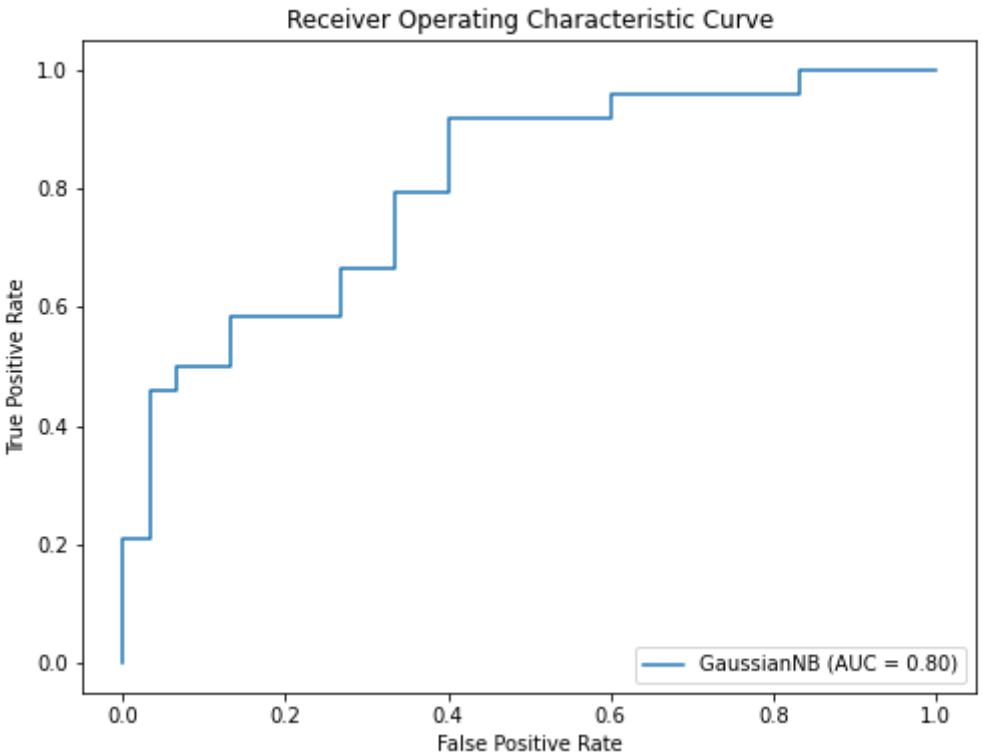

Fig: 4.9 Naive Bayes ROC Curve

4.2. Results

Model	Accuracy	Precision	Sensitivity	Specificity	F1 Score	ROC	Log_Loss	Mathew_ correcoef
Random Forest	0.8519	0.8667	0.8667	0.8667	0.8667	0.85	19.1886	0.7
KNN	0.7963	0.7879	0.8667	0.8667	0.8254	0.7875	19.1886	0.5861
Decision Tree	0.7037	0.7692	0.6667	0.6667	0.7142	0.7083	19.1886	0.4144
Naive Bayes	0.6667	0.7143	0.6667	0.6667	0.6896	0.6667	19.1886	0.3315

Table 4.7 Final Result

```
scores = [score_rf,score_knn,score_dt,score_nb]
Models = ["Random Forest Classifier"," K-Nearest Neighbors Classifier",
          "Decision Tree Classifier","Naive Bayes Classifier"]

for i in range(len(Models)):
    print("The accuracy score achieved using "+Models[i]+" is: "+str(scores[i])+" %")
```

```
The accuracy score achieved using Random Forest Classifier is: 85.19 %
The accuracy score achieved using  K-Nearest Neighbors Classifier is: 79.63 %
The accuracy score achieved using Decision Tree Classifier is: 70.37 %
The accuracy score achieved using Naive Bayes Classifier is: 66.67 %
```

Fig: 4.10 Final Accuracy Score

4.3. Graphs

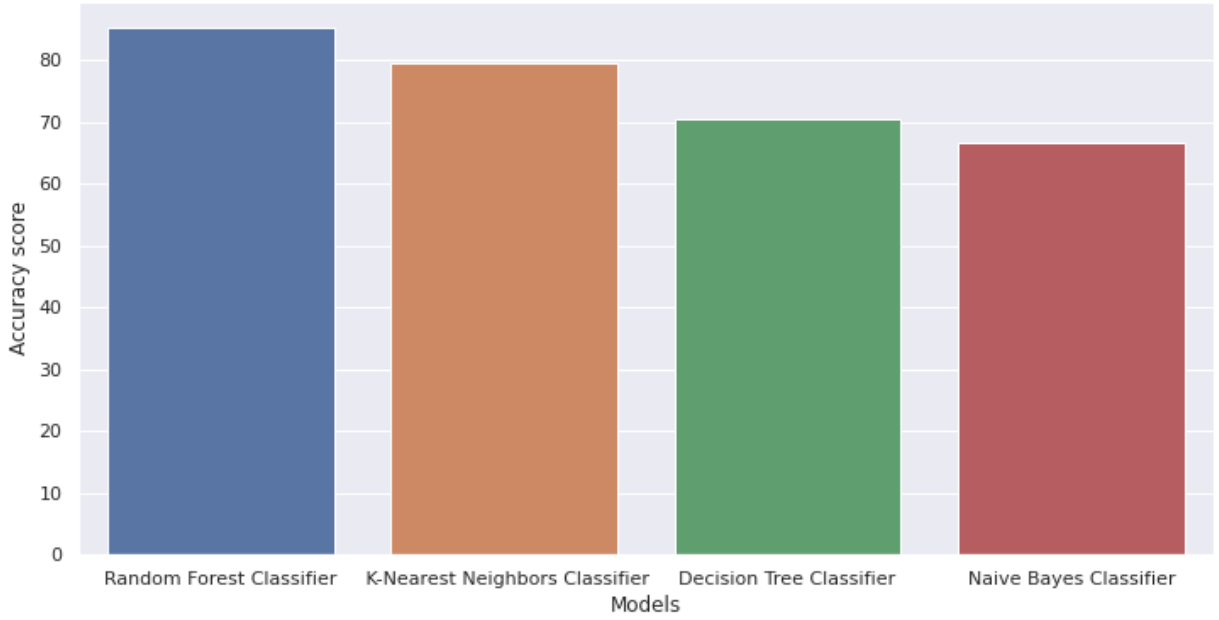

Fig: 4.11 Accuracy Score Bar Graph

4.4. Snapshots

```
Input = (67, 0, 3, 115, 564, 0, 2, 160, 0, 1.6, 2, 0, 7)

Input_array= np.asarray(Input)
Input_reshaped = Input_array.reshape(1,-1)

prediction = rf_classifier.predict(Input_reshaped)
prediction = np.around(prediction)

print(prediction)

if (prediction[0]== 1):
  print('The Person does not have a Heart Disease')
else:
  print("The Person is likely to have Heart Disease by %f "%(prediction))
```

```
[1]
The Person does not have a Heart Disease
```

Fig: 4.12 Sample Test 1

```
Input = (70,1,4,130,322,0,2,109,0,2.4,2,3,3)

Input_array= np.asarray(Input)
Input_reshaped = Input_array.reshape(1,-1)

prediction = rf_classifier.predict(Input_reshaped)
prediction = np.around(prediction)

print(prediction)

if (prediction[0]== 1):
  print('The Person does not have a Heart Disease')
else:
  print("The Person is likely to have Heart Disease by %f "%(prediction))
```

```
[2]
The Person is likely to have Heart Disease by 2.000000
```

Fig: 4.13 Sample Test 2

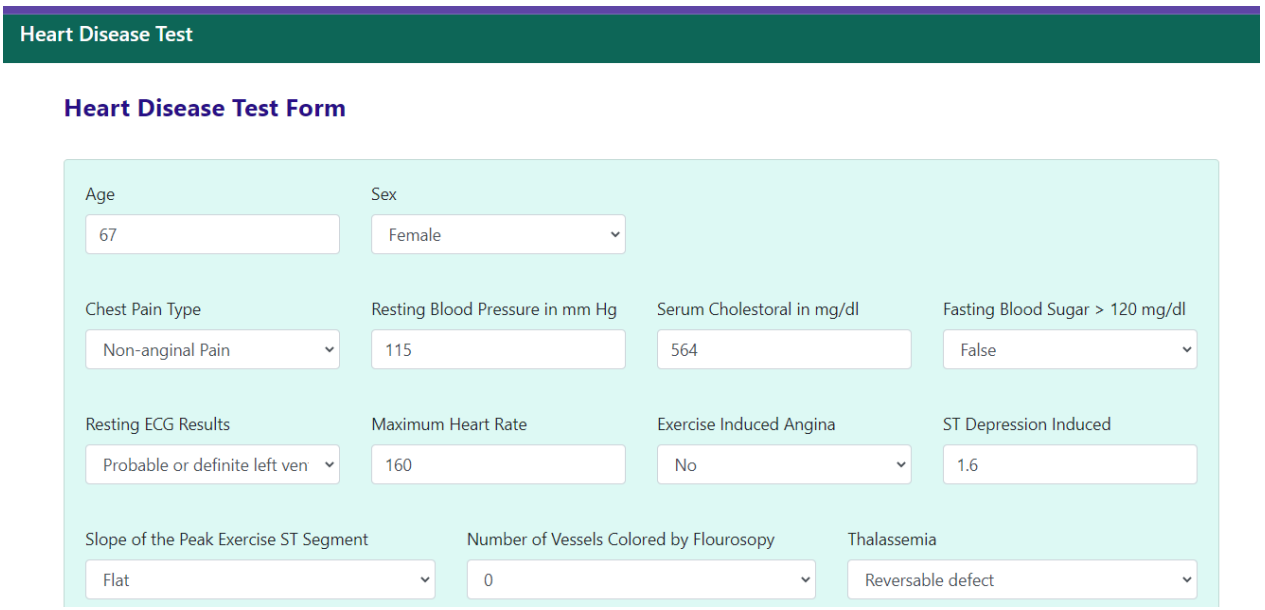

Fig: 4.14 Heart Disease Test 1

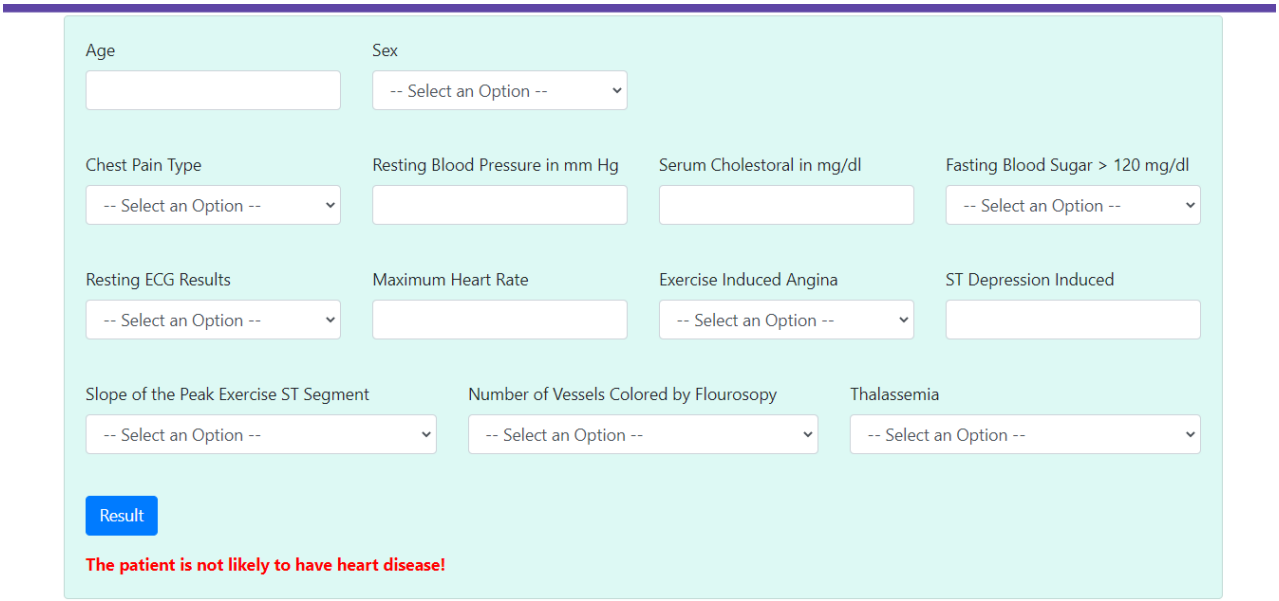

Fig: 4.15 Heart Disease Test 1 Result

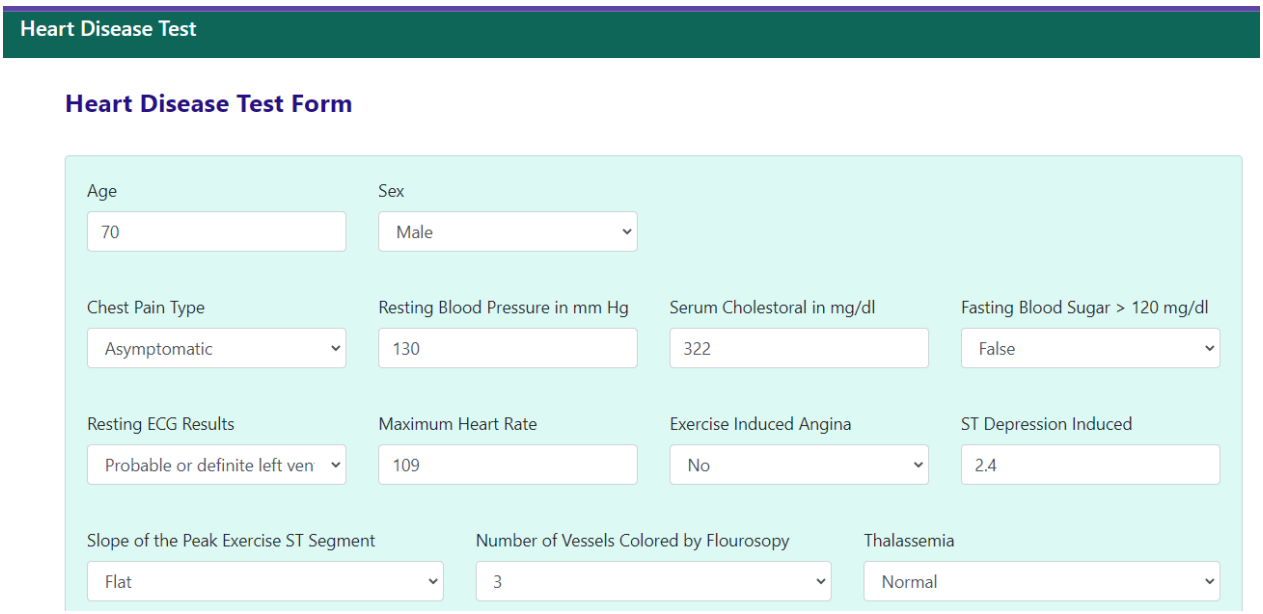

Fig: 4.16 Heart Disease Test 2

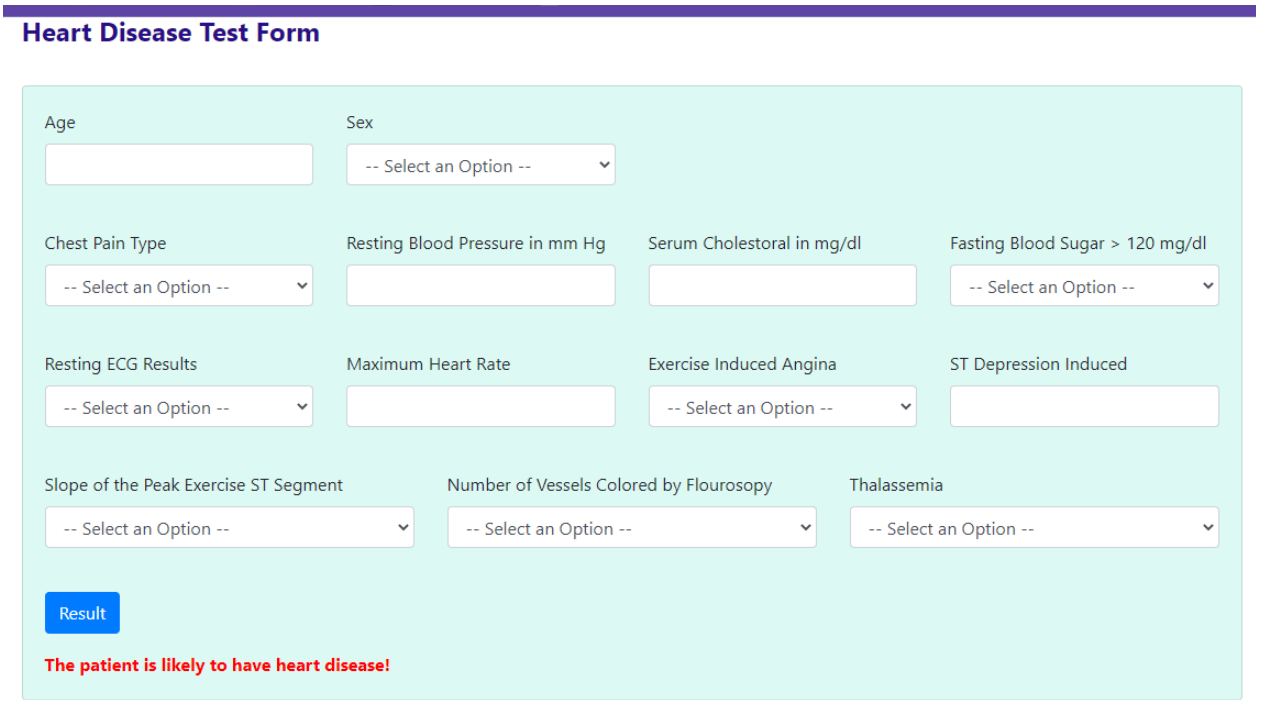

Fig: 4.17 Heart Disease Test 2 Result

CHAPTER 5

CONCLUSION AND FUTURE WORK

5.1. Conclusion

The overall objective of our internship project is to predict accurately with fewer tests and attribute the presence of heart disease. In this internship project, fourteen attributes are considered which form the primary basis for tests and give accurate results more or less. Many more input attributes can be taken but our goal is to predict with less number of attributes and faster efficiency to predict the risk of having heart disease at a particular age span. Four classification techniques were applied namely Random Forest, K-Nearest Neighbors, Decision Tree, Naive Bayes. It is shown that Random Forest has better accuracy than the other techniques.

This is the most effective model to predict patients with heart disease. This internship project could answer complex queries, each with its own strength with respect to ease of model interpretation, access to detailed information and accuracy.

This internship project is presented using four classification techniques. Random Forest, K-Nearest Neighbors, Decision Tree, Naive Bayes are used to develop the system. Random Forest proves the better results and assists the domain experts and even the person related to the medical field to plan for a better and early diagnosis for the patient. This system performs realistically well even without retraining.

5.2. Limitations

The Algorithms used in our internship project do not give a 100% accuracy, so the prediction is not 100% feasible. Clinical diagnosis and diagnosis using our internship project may differ slightly because the prediction is not 100% accurate. 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 the doctor's intuition and experience rather than on the knowledge rich data collected from the dataset.

5.3. Future Work

We are planning to introduce an efficient disease prediction system to predict heart disease with better accuracy using Support Vector Machine (SVM). Our internship project aims to provide a web platform to predict the occurrences of disease based on various symptoms. The user can select various symptoms and can find the diseases with their probabilistic figures.

Our internship project can be improved by implementing medicine suggestions to the patient along with the results. We can implement feedback from the experienced doctors who can give their views and opinions about certain medicines/practices done by the doctor on the patient. We can implement a live chat option where the patient can chat with a doctor available regarding medication for the respective result for their symptoms. Our internship project could be used as a training tool for Nurses and Doctors who are freshly introduced in the field related to heart diseases. The patient can have a choice in choosing the medicines he/she should take in order to have a healthier life. Moreover, if implemented on a large scale it can be used in medical facilities like hospital, clinics where a patient wouldn't have to wait in long queues for treatment if he is feeling symptoms related to heart disease.

REFERENCES

[1] Halaudi Daniel M., "Prediction of heart disease using classification algorithms.," WCSECS, pp. 22–24, 2014.

[2] U. N. Dulhare and M. Ayesha, "Extraction of action rules for chronic kidney disease using Naïve bayes classifier," in 2016 IEEE International Conference on Computational Intelligence and Computing Research, ICCIC 2016, 2017.

[3] T. J. Peter and K. Somasundaram, "Study and Development of Novel Feature Selection Framework for Heart Disease Prediction," Int. J. Sci. Res. Publ., 2012.

[4] B. Xue, M. Zhang, and W. N. Browne, "Particle swarm optimization for feature selection in classification: A multi-objective approach," IEEE Trans. Cybern., 2013.

[5] C. Ordonez, "Improving Heart Disease Prediction using Constrained Association Rules," Tech. Semin. Present. Univ. Tokyo, 2004.

[6] M. C. and P. M. Franck Le Duff, CristianMunteanb, "Predicting Survival Causes After Out of Hospital Cardiac Arrest using Data Mining Method," Stud. Health Technol. Inform., vol. Vol. 107, no. 2, p. No. 2, pp. 1256–1259, 2004.

[7] W. J. F. and G. Piatetsky-Shapiro, "Knowledge Discovery in Databases: An Overview," AI Mag., vol. Vol. 13, N, no. 3, pp. 57–70, 1996.

[8] K. Y. N. and K. H. R. Heon Gyu Lee, "Mining Bio Signal Data: Coronary Artery Disease Diagnosis using Linear and Nonlinear Features of HRV," Proc. Int. Conf. Emerg. Technol. Knowl. Discov. Data Min., p. pp. 56–66, 2007.

[9] L. P. and R. Subramanian, "Intelligent Heart Disease Prediction System using CANFIS and Genetic Algorithm," Int. J. Biol. Biomed. Med. Sci., vol. Vol. 3, no. No. 3, pp. 1-8, 2008.

[10] UCI Dataset (https://archive.ics.uci.edu/ml/datasets/statlog+(heart)).

[11] M. Kukar, I. Kononenko, C. Grošelj, K. Kralj, and J. Fettich, "Analyzing and improving the diagnosis of ischaemic heart disease with machine learning," Artif. Intell. Med., vol. 16, no. 1, pp. 25–50, May 1999.