Predicting Heart Disease using Machine Learning

Submitted in partial fulfillment of the requirements

for the degree of

Bachelor of Engineering

by

Sushant Kumar Mishra

Roll No.40

Shubham Ingle

Roll No.20

Sarthak Jadhav

Roll No.23

Under the Supervision of

Prof. Poonam B. Lad

DEPARTMENT OF INFORMATION TECHNOLOGY

KONKAN GYANPEETH COLLEGE OF ENGINEERING

KARJAT-410201

May 2021

Certificate

This is to certify that the project entitled **Predicting Heart Disease using Machine Learning** is a bonafide work of **Shubham Ingle (Roll No. 20),Sarthak Jadhav(Roll No.23) and Sushant Kumar Mishra(Roll No.40)** submitted to the University of Mumbai in partial fulfillment of the requirement for the award of the degree of **Undergraduate** in **DEPARTMENT OF INFORMATION TECHNOLOGY**.

Prof. Poonam B. Lad

Assistant Professor

Department of Information Technology

Dr. M. J. Lengare

Head of Department

Department of Information Technology

Principal

Konkan Gyanpeeth College of Engineering

Project Report Approval for B.E.

This project report entitled **Predicting Heart Disease using Machine Learning** by **Shubham Ingle (Roll No. 20),Sarthak Jadhav(Roll No.23) and Sushant Kumar Mishra(Roll No.40)** is approved for the degree of **DEPARTMENT OF INFORMATION TECHNOLOGY**.

Examiners

1.....

2.....

Date.

Place.

Declaration

I declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Signature
Sushant Kumar Mishra (Roll No 40)

Signature
Sarthak Jadhav (Roll No 23)

Signature
Shubham Ingle (Roll No 20)

Date.

Abstract

Predicting and detection of heart disease has always been a critical and challenging task for healthcare practitioners. Hospitals and other clinics are offering expensive therapies and operations to treat heart diseases. So, predicting heart disease at the early stages will be useful to the people around the world so that they will take necessary actions before getting severe. Heart disease is a significant problem in recent times; the main reason for this disease is the intake of alcohol, tobacco, and lack of physical exercise. Over the years, machine learning shows effective results in making decisions and predictions from the broad set of data produced by the health care industry. Some of the supervised machine learning techniques used in this prediction of heart disease are Logistic Regression , random forest (RF), ,knearest neighbour algorithm. Furthermore,system made on the performances of these algorithms is summarized

Acknowledgements

We wish to express our profound and sincere gratitude to **Prof. P. B. Lad**, Department Information Technology, KGCE, Karjat, who guided us into the intricacies of this project with matchless magnanimity. We thank Head of the Dept. of Information Technology, KGCE Karjat and **Dr. M. J. LENGARE**, Principal, KGCE Karjat for extending their support during the Course of this investigation. We would be failing in our duty if we don't acknowledge the co-operation Rendered during various stages of image interpretation by. We are highly grateful to who evinced keen interest and invaluable support in the progress and successful completion of our project work. We are indebted to for their constant encouragement, co-operation and help. Words of Gratitude are not enough to describe the accommodation and fortitude which they have shown throughout my endeavor.

Contents

List of Figures

List of Tables

Abbreviations

CVD	**C**ardio **V**ascular **D**iseas
EDA	**E**xploratory **D**ata **A**nalysis
CAD	**C**oronary **A**rtery **D**iseas

Chapter 1

INTRODUCTION

1.1 Introduction

The heart is one of the main parts of the human body after the brain. The primary function of the heart is to pumping blood to the whole body parts. Any disorder that can lead to disturbing the functionality of the heart is called heart disease. Several types of heart disease are there in the world; CAD, HF are the most common heart diseases that are present. The main reason behind the coronary heart disease CAD is blockage or narrowing down of the coronary arteries Coronary arteries also responsible for supplying blood to the heart. CAD is the leading cause of death over 26 million people are suffering from coronary heart disease(CAD) around the world, and it is increasing 2% annually . In the growing world, 2% of the population around the world is suffering from CAD, and 10% of the people are older than 65 years. Approximately 2% of the annual healthcare budget spent only to treat CAD disease.

Heart disease is the leading cause of death among all others diseases ,even cancers.One in 4 deaths in India are now because of CVDs with ischemic heart disease and stroke.The diagnosis is often made, based on doctor's intutions and experience ,this may lead to an unwanted result and excessive medical cost. Heart disease is a significant issue, so there is a need for diagnosis or prediction of heart disease there areseveral methods to diagnose heart disease among them. Angiography is the trending method which is used by most of the physicians across the world. However, there are some drawbacks associated with angiography technique. It is an expensive procedure and physicians have to analyze so many factors to diagnose a patient hence this process makes physician job very difficult, so these limitations motivate to develop a non-invasive method for prediction of heart

disease. These conventional methods deal with medical reports of the patients moreover these conventional methods are time-consuming, and it may give erroneous results because these conventional methods are performed by humans. To avoid these errors and to achieve better and faster results, we need an automated system. Over the past years, researchers find out that machine learning algorithms perform very well in analyzing medical data sets. These data sets will be directly given to machine learning algorithms, and machine learning algorithms will perform according to their nature, and those algorithms will give some outputs.

1.2 Objectives

Objectives of this project is :

1. to go through the data science lifecycle steps in order to build a heart disease classification web application by using a historical dataset
2. to use flask API to deploy the model and build the web application
3. to Help avoid human biasness.
4. to Reduce the cost of medical tests.

1.3 Purpose, Scope, and Applicability

Purpose, Scope and Applicability: The description of Purpose, Scope, and Applicability are given below:

1.3.1 Purpose

According to the World Health Organization, every year 12 million deaths occur worldwide due to Heart Disease. The load of cardiovascular disease is rapidly increasing all over the world from the past few years. Many researches have been conducted in attempt to pinpoint the most influential factors of heart disease as well as accurately predict the overall risk. Heart Disease is even highlighted as a silent killer which leads to the death of the person without obvious symptoms.The early diagnosis of heart disease plays a vital role in making decisions on lifestyle changes in high-risk patients and in turn reduce the

complications. 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

1.3.2 Scope

Machine learning techniques have been around us and has been compared and used for analysis for many kinds of data science applications.Heart disease is the leading cause of death among all others diseases ,even cancers.One in 4 deaths in India are now because of CVDs with ischemic heart disease and stroke.Prediction of heart diseases is a difficult and risky task. Since it is directly dependent on people's health, accuracy is a major factor. If not predicted accurately it can be disastrous.This project therefore focuses on the comparison of different data mining techniques to predict it. It shows the comparative analysis of the different methods. Cross validation error is used to compare the techniques. We choose Logical Regression, Random forest, K-Nearest Neighbors,as they are the most widely used techniques in determining diseases.

1.3.3 Applicability

Machine learning is widely used now a days in many business applications like e commerce and many more. Prediction is one of area where this machine learning used, our topic is about prediction of heart disease by processing patient's data set and a data of patients to whom we need to predict the chance of occurrence of a heart disease

1.4 Achievements

Achievements: Explain what knowledge you achieved after the completion of your work. What contributions has your project made to the chosen area? Goals achieved describe the degree to which the findings support the original objectives laid out by the project. The goals may be partially or fully achieved, or exceeded.

1.5 Organisation of Report

In introduction we studied about basic over view of the project in which we learned various outcomes of the project. how the project is going to the implemented what are the prerequisites require to develop the project etc.

In literature survey we will see various presents system the which are previously developed by various experts and we had compared system which are previously design.

in survey of technologies we will see various technologies which can be use to complete the system.

In requirements and analysis we will analyse various requirements pre-quest for developing the system also see various functional requirements software requirements and hardware requirements of the system.

In system design we see basic overall design of the system.

In the implementation chapter we will see how the system is implemented.This chapter also contain implementation strategies and test strategies

In Results we will see test results which we have been obtained by testing various algorithms and methods.

In final chapters we will see a conclusion of the project it also describes limitations and future scope of the system.

Chapter 2

LITERATURE SURVEY

Literature Survey : Numerous studies have been done that have focus on diagnosis of heart disease. They have applied different data mining techniques for diagnosis and achieved different probabilities for different methods.

2.0.1 An Optimized Stacked Support Vector Machines Based Expert System for The Effective Prediction Of Heart Failure

Authors: Liaquat Ali et al
Year: 2019

This paper recommended a model which is consists of two methods one is X^2 statistical and deep neural network(DNN). Feature refinement is done by X^2 statistical model and classification is done by a deep neural network(DNN). In their study, they have used the Cleveland dataset. There are 303 instances in that dataset, among them, 297 have no missing data, and the remaining 6 have missing data. Among 297, 207 instances are used for training data, and the remaining 90 are used as testing data. This model gives better results compared to conventional ANN models which are present earlier. As a result of this using this proposed model, they have got 93.33 % classification accuracy using DNN. It is 3.33 percent more than that of the conventional ANN model.

2.0.2 Effective Heart Disease Prediction System

Authors: Dr. Kanak Saxena et.al
Year: 2016

This paper developed a data mining model to predict heart disease efficiently. It mainly helps the medical practitioners to make efficient decisions way based on the given parameters. The author has used Cleveland dataset from UCI, and they have used age, sex, resting blood pressure, chest pain, serum cholesterol, fasting blood sugar, etc. as attributes. Furthermore, they have divided the datasets into two parts one is for testing, and the other one is for training. They have used a 10-fold method to find accuracy

2.0.3 An System Based On Support Vector Machines For Effective Dignosis Of Heart Disease

Authors : Awais Nimat et al
Year :2016

This paper proposed an expert system based on two support vector machines(SVM) to predict heart disease efficiently. These tow SVM's have their purpose; first, one is used to remove the unnecessary features, and the second one is used for prediction. Moreover, they have used the HGSA (hybrid gird search algorithm) to optimize the two methods. By using this model, they have achieved 3.3 % better accuracy than the conventional SVM models that are present earlier.

2.0.4 An Intelligent Learning System Based O Random Search Algorithm And Optimized Random Forest Model For Improved Heart Disease Detection

Authors : Ashir Javeed et al
Year :2019

This paper developed a model to improve the prediction of heart disease by overcoming the problem of overfitting; overfitting means the proposed model performs and gives better accuracy on testing data and gives unfortunate accuracy result for training data while predicting the heart disease. To solve this problem, they have developed a model that will give the best accuracy on both training and testing data. That model consists of two

algorithms one is RAS (Random search algorithm) other one is a random forest algorithm that is used to predict the model. This proposed model gave them better results in training data as well as testing data.

Paper Comparison

S.R. NO.	**Authors**	**Techniques Used**	**Accuracy**
1	Liaqat Ali et al.	X^2 statistical model, deep neural network	93.33%(holdout) 91.57%(k-fold
2	Dr.Kanak Saxena et.al	Decision tree	86.3% (testing phase) 87.3% (training phase)
3	Awais Nimat et al.	Support vector machine, Hybrid grid search algorithm (HGSA)	92.22% (L1 linear SVM+L2 linear & RBF SVM)
4	Ashir Javeed et al.	Random search algorithm (RSA), Random forest.	93.33% (RSA+RF)

TABLE 2.1: Literature survey comparison

Chapter 3

REQUIREMENTS AND ANALYSIS

3.1 Problem Definition

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

3.2 Requirements Specification

we will need a historic data set to study its hidden patterns and to train the models which we are going to use for prediction.and then after building the user interface of the system ,users will need health related reports containing blood sugar level,serum cholesterol,number of major vessels effected ,maximum heart rate achieved etc.

3.3 Planning and Scheduling

Planning and scheduling is a complicated part of software development. Planning, for our purposes, can be thought of as determining all the small tasks that must be carried out in order to accomplish the goal. Planning also takes into account, rules, known as constraints, which, control when certain tasks can or cannot happen. Scheduling can be thought of as determining whether adequate resources are available to carry out the plan. You should show the Gantt chart and Program Evaluation Review Technique (PERT).

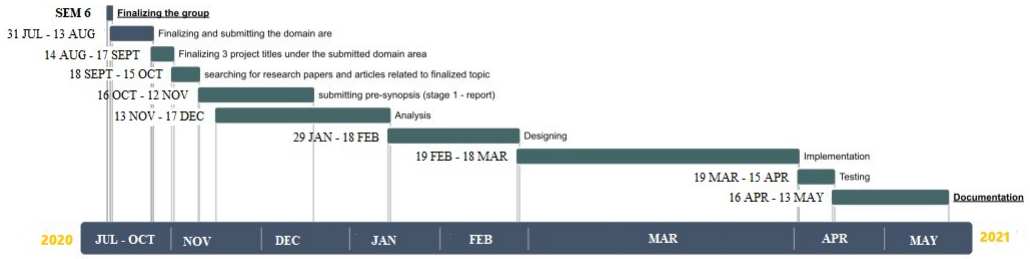

FIGURE 3.1: gantt chart

3.4 Software and Hardware Requirements

Hardware Requirement:

- 1.5 gigahertz (GHz) dual-core C.P.U
- 4 GB RAM
- 1024x768 minimum screen resolution
- 10GB Of hard disk space

Software Requirements:

- A miniconda environment
- jupyter notebook
- Python 3.8 (recommended)

- Flask
- Heroku cloud platform
- Heroku CLI or github
- Git
- Visual studio code and sublime text
- Creatly and Star ml

3.5 Preliminary Product Description

In this project we are using data set from cleaveland database from UCI machine learning Repository. however, we have downloaded it in a formatted way from kaggle.we are going to use 14 important attributes from total 76 attributes from database .we are using three algorithms i.e. logistic regression,k-nearest neighbor and Random forest for make our best model of prediction.we are using Flask API for deployment of our model through heroku platform into web application.

3.6 Conceptual Models

3.6.1 Data flow diagram:1

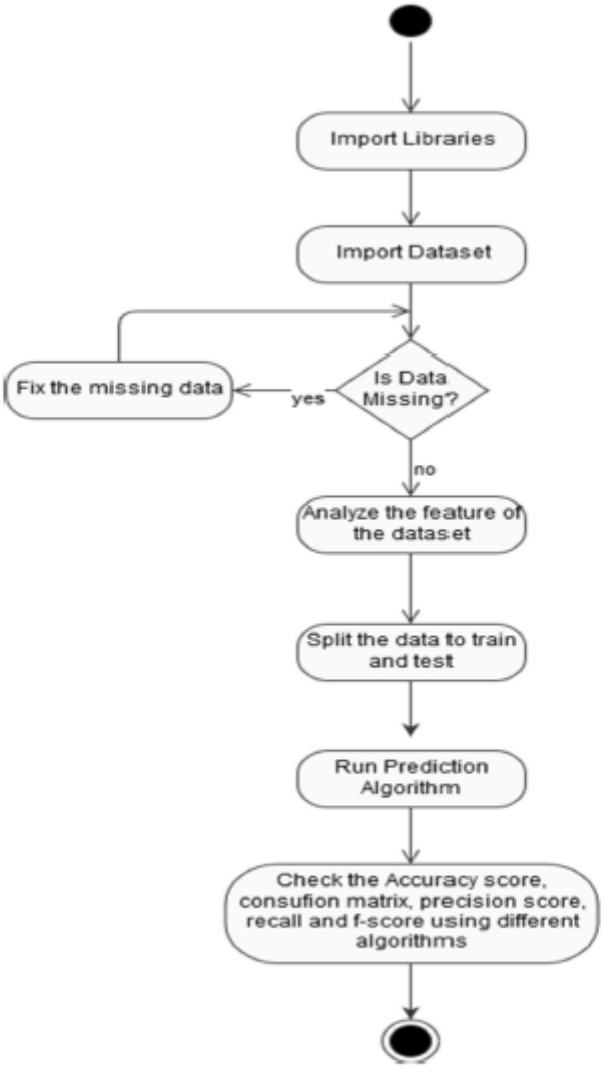

FIGURE 3.2: DFD 1

3.6.2 Data flow diagram:2

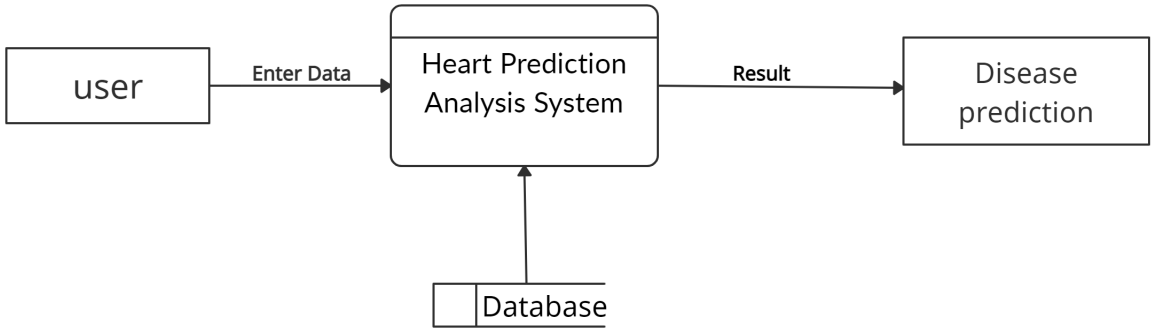

FIGURE 3.3: DFD 2

Chapter 4

SYSTEM DESIGN

4.1 Basic Modules

We had followed the divide and conquer theory, so we divided the overall problem into 3 parts and develop each part or module separately. When all modules are ready, we should integrate all the modules into one system.we briefly described all the modules and the functionality of these modules bellow.

- Data selection and EDA module
- Data preparation and modeling module
- Deployment module

4.2 Data Design

4.2.1 Schema Design

we are using data set from cleaveland database from UCI machine learning Repository..we are goingto use 14 important attributes from total 76 attributes from database .out of 14 attributes 1 is our target attribute. these attributes are as follows:

- age: age in years.
- sex: sex (1 = male; 0 = female).

- cp: chest pain type (Value 0: typical angina; Value 1: atypical angina; Value 2: non-anginal pain; Value 3: asymptomatic).
- trestbps: resting blood pressure in mm Hg
- chol: serum cholestoral in mg/dl.
- fbs: fasting blood sugar ¿ 120 mg/dl (1 = true; 0 = false).
- restecg: resting electrocardiographic results (Value 0: normal; Value 1: having ST-T wave abnormality; Value 2: probable or definite left ventricular hypertrophy).
- thalach: maximum heart rate achieved.
- exang: exercise induced angina (1 = yes; 0 = no).
- oldpeak: ST depression induced by exercise relative to rest.
- slope: the slope of the peak exercise ST segment (Value 0: upsloping; Value 1: flat; Value 2: downsloping).
- ca: number of major vessels (0-3) colored by flourosopy.
- thal: thalassemia (3 = normal; 6 = fixed defect; 7 = reversable defect).
- target: heart disease (1 = no, 2 = yes).

4.3 Procedural Design

4.3.1 Logic Diagrams

4.3.1.1 State diagram:

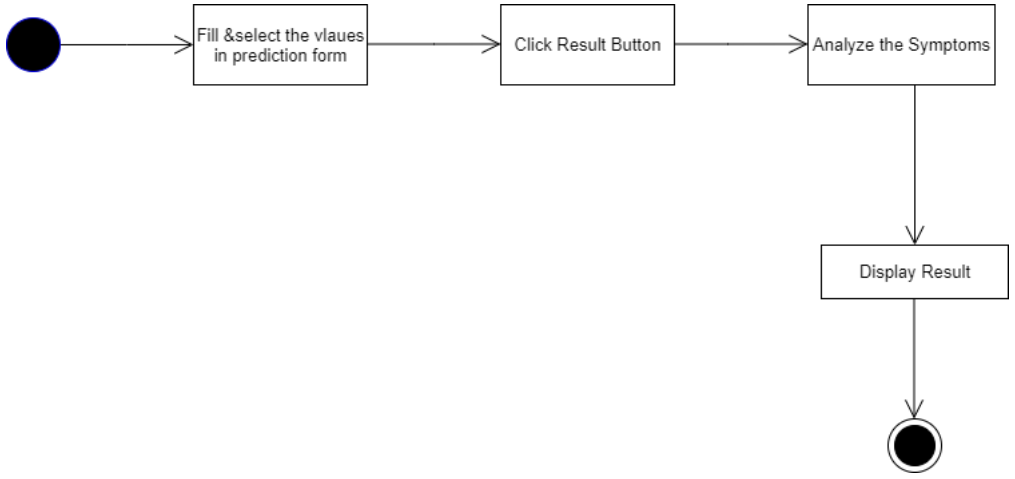

FIGURE 4.1: state diagram

4.3.1.2 Sequence diagram:

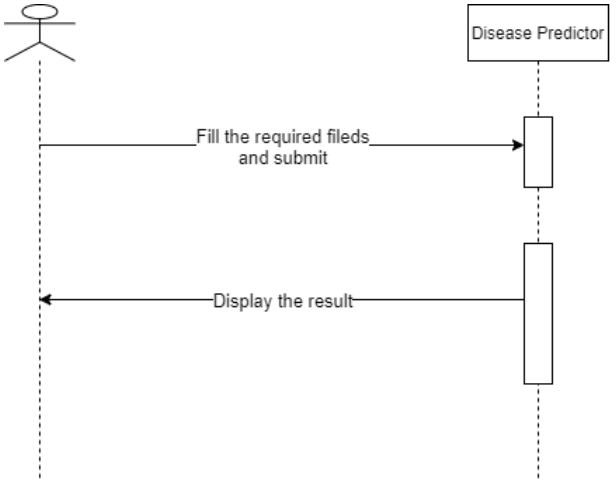

FIGURE 4.2: Sequence diagram

4.3.2 Data Structures

From the dataset that we have chosen for making our model of prediction of heart disease we have used 3 widely used algorithms Logical Regression, Random forest and K-Nearest Neighbors 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.

4.3.2.1 Importing libraries

```
In [1]:
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline

from sklearn.linear_model import LogisticRegression
from sklearn.neighbors import KNeighborsClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split, cross_val_score
from sklearn.model_selection import RandomizedSearchCV, GridSearchCV
from sklearn.metrics import confusion_matrix, classification_report
from sklearn.metrics import precision_score, recall_score, f1_score
from sklearn.metrics import plot_roc_curve
```

FIGURE 4.3: Importing libraries

4.3.2.2 load data

```
In [2]: df = pd.read_csv("heart-disease.csv")
```

FIGURE 4.4: load data

4.3.2.3 check the shape of the data

```
In [5]: df.shape
Out[5]: (303, 14)
```

FIGURE 4.5: shape of the data

4.3.2.4 dataset description

```
In [10]: df.describe()
Out[10]:
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	
count	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303
mean	54.366337	0.683168	0.966997	131.623762	246.264026	0.148515	0.528053	149.646865	0.326733	1.039604	1.399340	0.729373	2
std	9.082101	0.466011	1.032052	17.538143	51.830751	0.356198	0.525860	22.905161	0.469794	1.161075	0.616226	1.022606	0
min	29.000000	0.000000	0.000000	94.000000	126.000000	0.000000	0.000000	71.000000	0.000000	0.000000	0.000000	0.000000	0
25%	47.500000	0.000000	0.000000	120.000000	211.000000	0.000000	0.000000	133.500000	0.000000	0.000000	1.000000	0.000000	2
50%	55.000000	1.000000	1.000000	130.000000	240.000000	0.000000	1.000000	153.000000	0.000000	0.800000	1.000000	0.000000	2
75%	61.000000	1.000000	2.000000	140.000000	274.500000	0.000000	1.000000	166.000000	1.000000	1.600000	2.000000	1.000000	3
max	77.000000	1.000000	3.000000	200.000000	564.000000	1.000000	2.000000	202.000000	1.000000	6.200000	2.000000	4.000000	3

FIGURE 4.6: data description

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

FIGURE 4.7: data description

4.3.2.5 types of features

- **Categorical features** (Has two or more categories and each value in that feature can be categorized by them): **sex, chest pain**

- **Ordinal features** (Variable having relative ordering or sorting between the values): **fasting blood sugar, electrocardiographic, induced angina, slope, no of vessels, thalassemia,diagnosis**

- **Continuous features** (Variable taking values between any two points or between the minimum or maximum values in the feature column): **age, blood pressure, serum cholesterol, max heart rate, ST depression**

4.3.2.6 checking for missing Data

```
In [7]: df.isnull().sum()

Out[7]: age         0
        sex         0
        cp          0
        trestbps    0
        chol        0
        fbs         0
        restecg     0
        thalach     0
        exang       0
        oldpeak     0
        slope       0
        ca          0
        thal        0
        target      0
        dtype: int64
```

FIGURE 4.8: missing data

4.3.3 Algorithms Design

The main goal of the entire project is to predict heart disease occurrence with the highest accuracy. In order to achieve this, we have chosen three classification algorithms This section includes all brief information about these algorithms

4.3.4 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 project so we will only consider the classification part.

Random Forest pseudocode

1. Randomly select "k" features from total "m" features. Where k << m
2. Among the "k" features, calculate the node "d" using the best split point.
3. Split the node into daughter nodes using the best split.
4. Repeat 1 to 3 steps until "l" number of nodes has been reached.

Random forest prediction pseudocode

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

4.3.5 Logistic Regression

Logistic regression is a classification algorithm used to assign observations to a discrete set of classes. Unlike linear regression which outputs continuous number values, logistic regression transforms its output using the logistic sigmoid function to return a probability value which can then be mapped to two or more discrete classes.

Types of logical regression:

1. Binary (Pass/Fail)
2. Multi (Cats, Dogs, Sheep)

Sigmoid function

$S(z) = 1/1 + e^z$

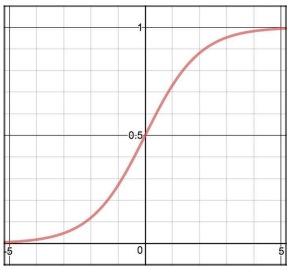

FIGURE 4.9: sigmoid function

Since the dichotomous experiments can only have one of two possible values for each experiment, the residuals will not be normally distributed about the predicted line. On the other hand, a logistic regression produces a logistic curve, which is limited to values between 0 and 1. Logistic regression is similar to a linear regression, but the curve is constructed using the natural logarithm of the "odds" of the target variable, rather than the probability. Moreover, the predictors do not have to be normally distributed or have equal variance in each group.

Procedure

1. Divide the problem into n+1 binary classification problem (+1 because the index starts at 0).
2. For each class...
3. Predict the probability the observations are in that single class.
4. prediction =< math>max (probability of the classes).

4.3.6 K-Nearest Neighbor

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

1. Load the data
2. Initialize the value of k
3. For getting the predicted class, iterate from 1 to total number of training data points

Procedure

- 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.

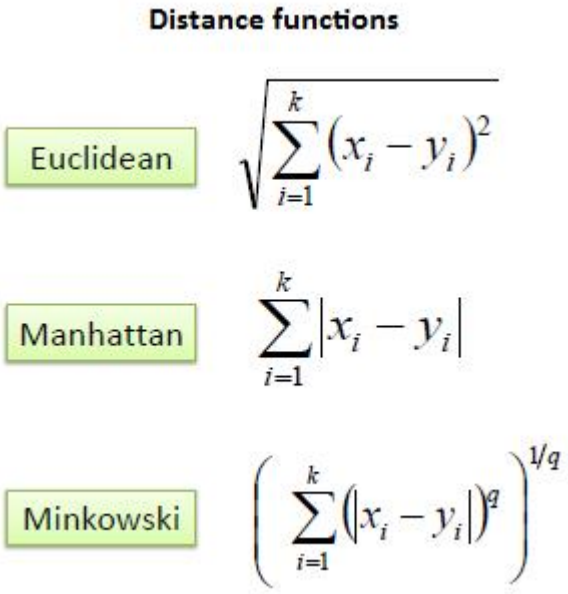

FIGURE 4.10: distance functions

4.4 User interface design

we have made a simple user interface design for users who want to know if they have chances of getting a heart disease or not for this user needs to perform some basic laboratory tests to fill the necessary input in the heart disease prediction form. The form has 13 inputs for the 13 features and a button. The button sends POST request to the/predict endpoint with the input data. In the form tag, the action attribute calls predict function when the form is submitted.Finally, the HTML page presents the stored result in the result parameter for users.

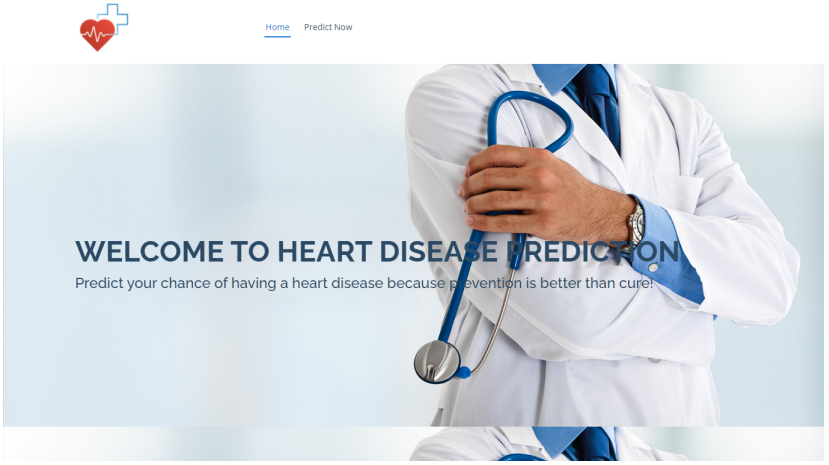

FIGURE 4.11: User interface-1

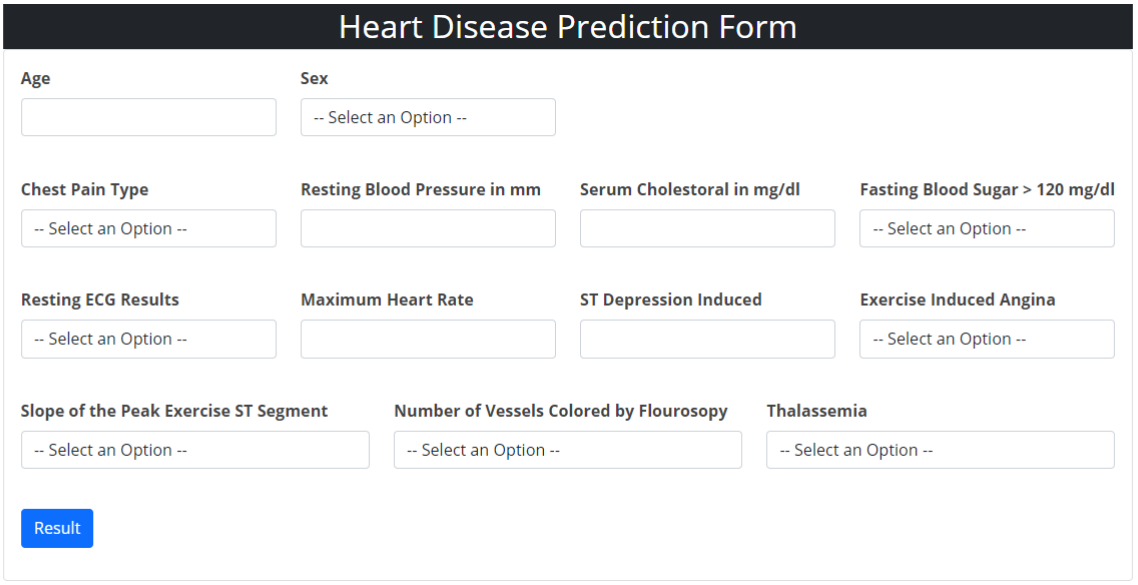

FIGURE 4.12: User interface-2

Chapter 5

IMPLEMENTATION AND TESTING

5.1 Implementation Approaches

In This Chapter ,we will study about how we had implemented this system using various methodology and technologies.

5.1.1 Programming Languages

5.1.1.1 Python:

we have used python as our main programming language for developing this project ,since it is widely used in machine learning rigorously. we have done EDA part,model making part ,web app code part using python and its various libraries and tools.the main libraries which we used in this project are:

- pandas for data analysis.
- NumPy for numerical operations.
- Matplotlib/seaborn for plotting or data visualization.
- Scikit-Learn for machine learning modelling and evaluation.
- pickle module to save our best prediction model and to load it in python web app eventually for hosting through FLASK API. ..etc.

5.1.1.2 HTML CSS :

We have used HTML(the Hypertext Markup Language) and CSS (Cascading Style Sheets) along with bootstrap framework for making user interface for entire system HTML and CSS are one of the core technologies for building this project where HTML provides the structure of the page CSS handles layout designs.we have also used bootstrap templates for making our application more attractive and user-friendly some templates we have used are : https://stackpath.bootstrapcdn.com" https://cdn.jsdelivr.net

5.1.2 Tools used:

Some of the main tools we used in our overall implementation process are as follows:

5.1.2.1 jupyter notebook:

Jupyter notebooks basically provides an interactive computational environment for developing Python based Data Science applications.A Jupyter Notebook provides a easy to use , interactive data science environment that dosent only work as an IDE ,but it helped us to present our code so as to explained it to anyone easily .and hence we used jupyter notebook to make our model

5.1.2.2 Anaconda:

In order to use jupyter notebook we installed Anaconda(more precisely miniconda environment because it takes less memory to run and works smoothly). Anaconda is a prepackaged distribution of Python which contains a number of Python modules and packages, including Jupyter.we enjoyed the flexibility provided by the andaconda nevigator because its ability to define a number of different "environments" with different frameworks.

5.1.2.3 Flask API:

We have used Flask framework in order to provide functionality for building our web application, including managing HTTP requests and rendering templates.to deploy our prediction model we created pickle file of model code so it can get loaded by flask environment when running python app.py.Flask runs on a server. This can be in the environment of the client or a different server depending on the client's requirements

5.1.2.4 Heroku:

Heroku is a container-based cloud Platform as a Service (PaaS) which allows us to deploy flask app freely over the internet. Heroku is fully managed, giving developers the freedom to focus on their core product without the distraction of maintaining servers, hardware, or infrastructure hence we used heroku .The Heroku platform uses Git as the primary means for deploying applications we have used Git CLI as well as GitHub for versioning our web application.

5.2 Coding Details and Code Efficiency

Our coding module consist of 3 parts

- EDA part
- Model building part
- Deployment part

5.2.1 EDA(Exploratory data analysis):

after we imported dataset ,the next step we have done is data exploration ,we tried to study our dataset as far as we can by comparing different columns to each other ,compare them to the target variable etc. some part of EDA is shown as follows:

5.2.1.1 Determining number of patients with or without heart problems in the given dataset

```
In [6]: df.target.value_counts()

Out[6]: 1    165
        0    138
        Name: target, dtype: int64
```

FIGURE 5.1: EDA1

5.2.1.2 Heart Disease Frequency according to Gender

```
In [10]: df.sex.value_counts()
Out[10]: 1    207
         0     96
         Name: sex, dtype: int64

         There are 207 males and 96 females

In [11]: pd.crosstab(df.target, df.sex)
Out[11]:
```

target	sex 0	1
0	24	114
1	72	93

FIGURE 5.2: EDA2

5.2.1.3 heart disease frequency per chest pain type

```
In [17]: pd.crosstab(df.cp, df.target)
Out[17]:
```

cp	target 0	1
0	104	39
1	9	41
2	18	69
3	7	16

FIGURE 5.3: EDA4

5.2.1.4 Age vs Max Heart rate for Heart Disease

Age vs Max Heart rate for Heart Disease

```
plt.figure(figsize=(10,6))

# We will start with positve examples
plt.scatter(df.age[df.target==1],
            df.thalach[df.target==1],
            c="salmon") # define it as a scatter figure

# Now for negative examples, we want them on the same plot, so we call plt again
plt.scatter(df.age[df.target==0],
            df.thalach[df.target==0],
            c="lightblue") # axis always come as (x, y)

# for plot
plt.title("Heart Disease in function of Age and Max Heart Rate")
plt.xlabel("Age")
plt.legend(["Disease", "No Disease"])
plt.ylabel("Max Heart Rate");
```

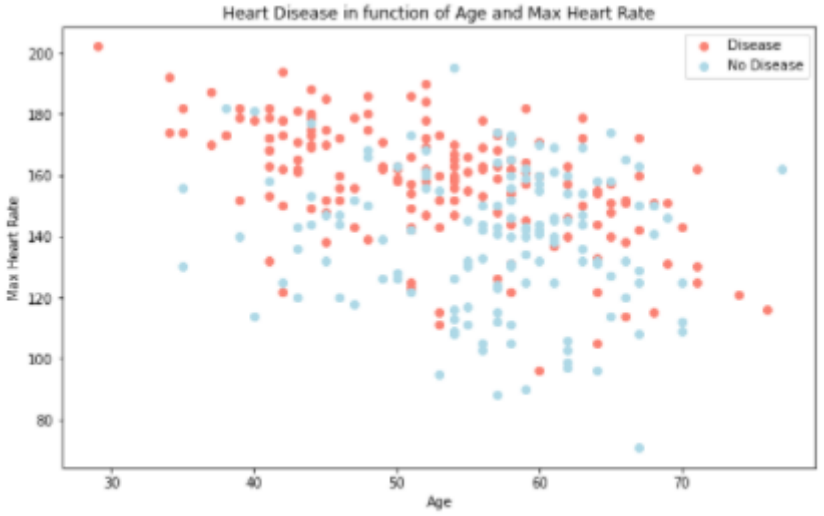

FIGURE 5.4: EDA5

5.2.1.5 Correlation between independent variables

```
In [20]:
corr_matrix = df.corr()
plt.figure(figsize=(15, 10))
sns.heatmap(corr_matrix,
            annot=True,
            linewidths=0.5,
            fmt= ".2f",
            cmap="YlGnBu");
```

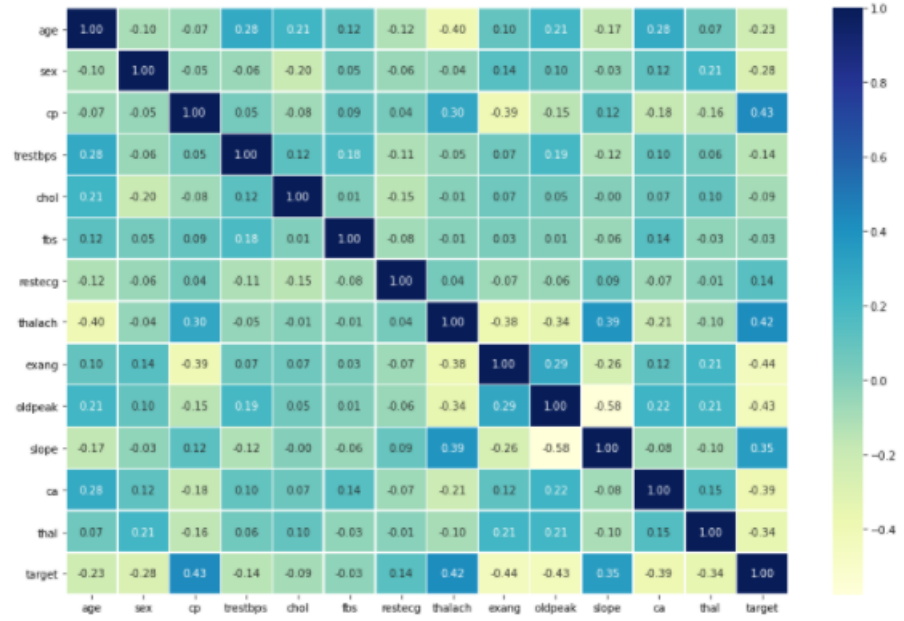

FIGURE 5.5: EDA6

5.2.2 model building

To split our data into a training and test set ,we have used Scikit-Learn's Train_test_split() and feed it out independent and dependent variables(X and y). we have kept 20 percentage of data for testing ,so we are using 80 percentage of it for training.

All of the algorithms in the Scikit-Learn library use the same functions, for training a model, model.fit(X_train, y_train) and for scoring a model model.score(X_test, y_test). score() returns the ratio of correct predictions (1.0 means 100 percent correct).

Since the algorithms we've chosen implement the same methods for fitting them to the data as well as evaluating them, we can optimize our coding by putting them in a dictionary and create a which fits and scores them. the code for this is as follows:

```
In [30]: # Putting models in a dictionary
models = {"KNN": KNeighborsClassifier(),
          "Logistic Regression": LogisticRegression(),
          "Random Forest": RandomForestClassifier()}

# Creating function to fit and score models
def fit_and_score(models, X_train, X_test, y_train, y_test):
    """
    Fits and evaluates given machine learning models.
    models : a dict of different Scikit-Learn machine learning models
    X_train : training data
    X_test : testing data
    y_train : labels assosciated with training data
    y_test : labels assosciated with test data
    """
    # Random seed for reproducible results
    np.random.seed(42)
    # Making a list to keep model scores
    model_scores = {}
    # Loop through models
    for name, model in models.items():
        # Fitting the model to the data
        model.fit(X_train, y_train)
        # Evaluating the model and append its score to model_scores
        model_scores[name] = model.score(X_test, y_test)
    return model_scores
```

FIGURE 5.6: modeling

```
In [31]: model_scores = fit_and_score(models=models,
                                      X_train=X_train,
                                      X_test=X_test,
                                      y_train=y_train,
                                      y_test=y_test)
         model_scores
```

```
Out[31]: {'KNN': 0.6885245901639344,
          'Logistic Regression': 0.8852459016393442,
          'Random Forest': 0.8360655737704918}
```

FIGURE 5.7: scoresoutput

5.2.3 Deployment:

For the deployment part we first save and loaded our mode into a pickle format as model.pkl file in the app to render it letter in our html page. the python app which we developed to load the model, get user input from the HTML template, make the prediction, and return the result.An HTML template made for the front end to allow the user to input heart disease symptoms of the patient and display if the patient may has heart disease or not. code of python app is as follows:

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

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

#to Create application
app = Flask(__name__)

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

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

    # to Put all form entries values in a list
    features = [float(i) for i in request.form.values()]



    # to Convert features to array
    array_features = [np.array(features)]
    # Predict features
    prediction = model.predict(array_features)

    output = prediction

    # to Check the output values and retrive the result with html tag based on the value
    if output == 1:
        return render_template('Heart Disease Classifier.html',
                               result = 'The patient is likely to have heart disease!')


    else:
        return render_template('Heart Disease Classifier.html',
                               result = 'The patient is not likely to have heart disease!')

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

FIGURE 5.8: app-code

5.3 Testing Approach

We had tested the system By using Following Approach which includes testing techniques listed bellow Testing strategies Unit Testing:

Unit Testing: Unit testing deals with testing a unit or module as a whole. This would test the interaction of many functions but, do confine the test within one module.

Integrated Testing: Brings all the modules together into a special testing environment, then checks for errors, bugs and interoperability. It deals with tests for the entire application. Application limits and features are tested here.

Test Case- I: Submit the symptoms from the list.
Precondition: The web application is open.
Assumptions: The symptoms or features are available

1. select the check boxes and fill all necessary fields from the form.
2. Select result

Expected Result: The symptoms selected should be submitted and further analyzed to calculate the probability of the disease.

5.4 Modifications and Improvements

we had found many small and large bugs in the systems from which we are mentioning some major bugs bellow:

1. we were not able to make pkl file properly out of our best accuracy model beacuase while modeling we putted all models for fitting into one dictionary .to solve this we made changes in the code by fitting best accuracy model sepreatly and then were able to save pkl file and then by loading the pkl file into our app code.
2. Complex User Interface Problem has been solved by using templates .

Chapter 6

RESULTS AND DISCUSSION

6.1 Test Reports

6.1.1 Test Results of Algorithms:

In this project as described earlier we used three algorithms for modeling i.e. Random Forest, logistic Regression and k-nearest neighbour . Out off this three algorithms we got highest prediction accuracy from model which is build using Logistic Regression which is 88.5245% ≅ 89%

further we tried to improve our model using hyperparameter tunning and cross validation. for tunning the hyperparameters for some of our models we used RandomizedSearchCV and GridSearchCV functions. RandomizedSearchCV tries _iter combinations of hyperparameters and saves the best,where GridSearchCV tries every single combination of hyperparameters and saves the best.then we generated confusion matrix to know where our best model made the right predictions and where it made the wrong predictions.we also created a classification report by using classification _ report() function of sklearn library which give us information of the precision and recall of our model for each class.classification report of logistic regression model is shown as below:

```
In [53]: # Show classification report
         print(classification_report(y_test, y_preds))
```

	precision	recall	f1-score	support
0	0.89	0.86	0.88	29
1	0.88	0.91	0.89	32
accuracy			0.89	61
macro avg	0.89	0.88	0.88	61
weighted avg	0.89	0.89	0.89	61

FIGURE 6.1: classification report

here , Precision Indicates the proportion of positive identifications (model predicted class 1) which were actually correct. A model which produces no false positives has a precision of 1.0.

Recall Indicates the proportion of actual positives which were correctly classified. A model which produces no false negatives has a recall of 1.0.

F1 score is a combination of precision and recall. a perfect model achieves an F1 score of 1.0.

Support indicates The number of samples each metric was calculated on.

The accuracy of the model in decimal form. Perfect accuracy is equal to 1.0.

Macro avg - Short for macro average, the average precision, recall and F1 score between classes.

Weighted avg is Short for weighted average, the weighted average precision, recall and F1 score between classes. Weighted means each metric is calculated with respect to how many samples there are in each class. This metric will favour the majority class (e.g. will give a high value when one class out performs another due to having more samples).

6.1.2 Test result of website:

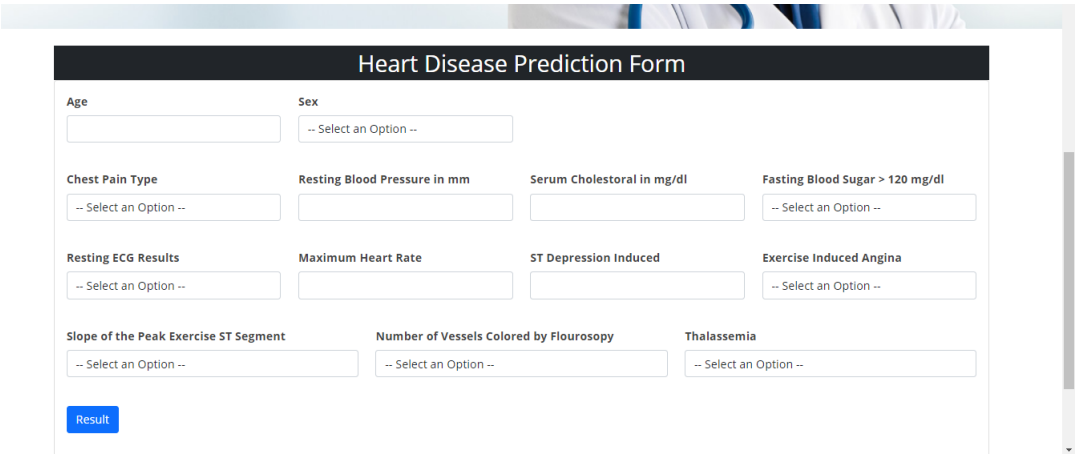

FIGURE 6.2: Initial state

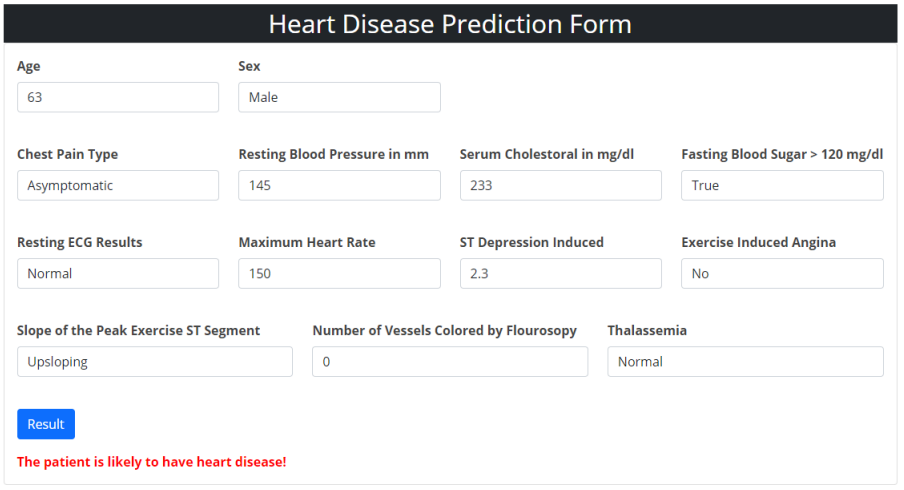

FIGURE 6.3: final state

6.2 User Documentation

User must have carried the the required tests in pathalogical lab(as mentioned in the features) before they use our heart disease predictor . we have kept our user interface of our website very simple and much user friendly .first user have to click on "predict now" button and then the form will open and user just need to fill the 13 Fields in the prediction form and then just need to click or tap on Result button at the end of the form .then users will get to know the probabilty of if they have heart disease or not .

Chapter 7

CONCLUSIONS

7.1 CONCLUSIONS

The overall objective of our project was to build a system that predicts accurately with less number of tests and attributes the presence of heart disease. In this 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. Three data mining classification techniques were applied namely K-Nearest Neighbor, Random Forest and Logistic Regression. It is shown that Logistic Regression has better accuracy of 89% which is better than the other techniques and then we successfully deployed our best model into web api .

7.2 Limitations of the System

Limitations of our existing system is as follows:

- The Algorithms used in our project does not give a 100% accuracy, so the prediction is not 100% feasible. Clinical diagnosis and diagnosis using our project may differ slightly because the prediction is not 100%
- the more accurate the system requires large no of data in training set to predict accurately, more the data the more accurate prediction is .

- since it require large no of data filed or parameters, the system has to be able to manage it.for managing that it require more time and much more complex data set.
- also our system doesnt gives medicine suggestion to user after showing results.

7.3 Future Scope of the Project

In future we have some plans to improve our existing system are as follows:

- Our project can be improved by implementing medicine suggestion to the patient along with the results
- We can implement a 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 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.
- further we are planning to build and mobile application as well. and also we will try to improve accuracy of our existing system by using new combination of algorithms and increasing and changing the data sets.

Bibliography

[1] A. Javeed and Nour] A. Javeed, S. Zhou, A. J. I. Q. A. N. and Nour, R. An intelligent learning system based on random search algorithm and optimized random forest model for improved heart disease detection.

[2] et al., L. A. (2019). An optimized stacked support vector machines based expert system for the effective prediction of heart failure. 7, pp. 54007–54014, 2019.

[3] L. Ali and Khan] L. Ali, A. Rahman, A. K. M. Z. A. J. and Khan, J. A. Automated diagnostic system for heart disease prediction based on 2 statistical model and optimally configured deep neural network.

[4] Purushottam, K. S. and Sharma, R. (2016). Efficient heart disease predict ion system," procedia comput. sci. 85, pp. 962–969,2016.