PROFESSIONAL TRAINING REPORT
at
Sathyabama Institute of Science and Technology
(Deemed to be University)

Submitted in partial fulfillment of the requirements for the award of

Bachelor of Engineering Degree in Computer Science and Engineering

By

ALLAM SRI SAI RAM

(Reg.No:40110069)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING
SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY
JEPPIAAR NAGAR, RAJIV GANDHI SALAI,
CHENNAI – 600119, TAMILNADU

APRIL 2022

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with Grade “A” by NAAC

(Established under Section 3 of UGC Act, 1956)

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI– 600119

www.sathyabamauniversity.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **ALLAM SRI SAI RAM (40110069)** who carried out the project entitled **“HEART DISEASE PREDECTION”,** under my supervision from February 2022 to April 2022.

Internal Guide

Mr. Murari Devakannan Kamalesh. M.E., Ph.D.,

Head of the Department

DR. L. Lakshmanan M.E., Ph.D.,

DR. S. Vigneshwari M.E., PH.D.,

Submitted for Viva voce Examination held on ____________________

Internal Examiner **External Examiner**

DECLARATION

I, ALLAM SRI SAI RAM hereby declare that the project report entitled **Heart disease prediction** done by me under the guidance of **Mr. Murari Devakannan Kamalesh M.E., Ph.D.,** is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Computer Science and Engineering.

DATE:

PLACE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management** of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala M.E., Ph.D.**, **Dean**, School of Computing, **Dr. S. Vigneshwari M.E., Ph.D. and Dr. L. Lakshmanan M.E., Ph.D.,** Heads of the Department of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Mr.Murari Devakannan Kamalesh. M.E., Ph.D.,** for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

TRAINING CERTIFICATE

ABSTRACT

The health care industries collect huge amounts of data that contain some hidden information, which is useful for making effective decisions. For providing appropriate results and making effective decisions on data, some advanced data mining techniques are used. In this study, a Heart Disease Prediction System (HDPS) is developed using Logistic Regression algorithms for predicting the risk level of heart disease. The system uses 15 medical parameters such as age, sex, blood pressure, cholesterol, and obesity for prediction. The HDPS predicts the likelihood of patients getting heart disease. It enables significant knowledge. E.g., Relationships between medical factors related to heart disease and patterns, to be established. We have employed the multilayer perceptron neural network with backpropagation as the training algorithm. The obtained results have illustrated that the designed diagnostic system can effectively predict the risk level of heart diseases.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1
INTRODUCTION

1.1 INTRODUCTION:

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

1.2 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.

1.3 MOTIVATION:

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

Hence, the early prognosis of cardiovascular diseases can aid in making decisions on lifestyle changes in high risk patients and in turn reduce the complications, which can be a great milestone in the field of medicine.

1.4 OBJECTIVES:

The main objective of developing this project are :

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

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

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

CHAPTER 2

AIM AND SCOPE OF THE PRESENT INVESTIGATION

2.1 AIM OF THE PROJECT:

The main aim of the heart disease prediction project is to determine if a patient should be diagnosed with heart disease or not. Which is a binary outcome, so: Positive result =1, the patient will be diagnosed with heart disease. Negative result =0, patient will not be diagnosed with heart disease.

2.2 SCOPE OF THE PROJECT:

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

2.3 LIMITATIONS:

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

CHAPTER 3
EXPERIMENTAL OR MATERIALS AND METHODS; ALGORITHMS USED

3.1 METHODS AND ALGORITHMS USED:

The main purpose of designing this system is to predict the ten-year risk of future heart disease. We have used Logistic regression as a machine-learning algorithm to train our system and various feature selection algorithms like Backward elimination and Recursive feature elimination. These algorithms are discussed below in detail

3.2 LOGISTIC REGRESSION:

Logistic Regression is a supervised classification algorithm. It is a predictive analysis algorithm based on the concept of probability. It measures the relationship between the dependent variable (Tenyear CHD) and the one or more independent variables (risk factors) by estimating probabilities using underlying logistic function (sigmoid function). Sigmoid function is used as a cost function to limit the hypothesis of logistic regression between 0 and 1 (squashing).

i.e., $0 \leq h\theta\ (x) \leq 1$.

In logistic regression cost function is defined as:

$$Cost(\mathrm{h\theta(x), y}) = \begin{cases} -\log\big(h\theta(x)\big) & if\ y = 1 \\ -\log\big(1-h\theta(x)\big) & if\ y = 0 \end{cases}$$

Logistic Regression relies highly on the proper presentation of data. So, to make the model more powerful, important features from the available data set are selected using Backward elimination and recursive elimination techniques. Logistic regression is one of the most popular Machine Learning algorithms, which comes under the Supervised Learning technique. It is used for predicting the categorical dependent variable using a given set of independent variables.

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

Logistic Regression is much similar to the Linear Regression except that how they are used. Linear Regression is used for solving Regression problems, whereas logistic regression is used for solving the classification problems. In Logistic regression, instead of fitting a regression line, we fit an "S” shaped logistic function, which predicts two maximum values (0 or 1).

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

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

3.2.1 ADVANTAGES:

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

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

This algorithm allows models to be updated easily to reflect new data, unlike Decision Tree or Support Vector Machine. The update can be done using stochastic gradient descent. Logistic Regression outputs well-calibrated probabilities along with classification results. This is an advantage over models that only give the final classification as results. If a training example has a 95% probability for a class, and another has a 55% probability for the same class, we get an inference about which training examples are more accurate for the formulated problem.

3.2.2 DISADVANTAGES:

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

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

Non-Linearly Separable Data:

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

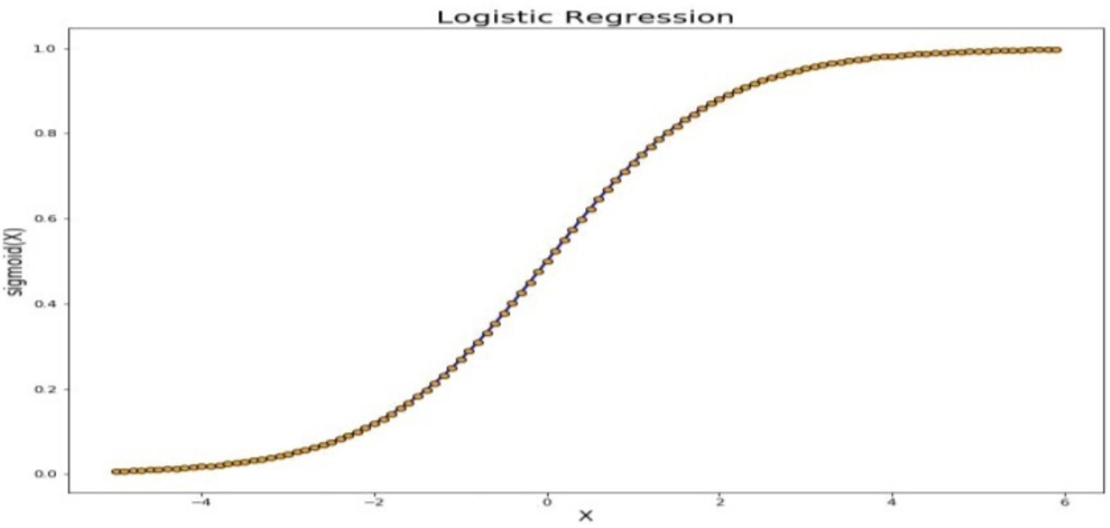
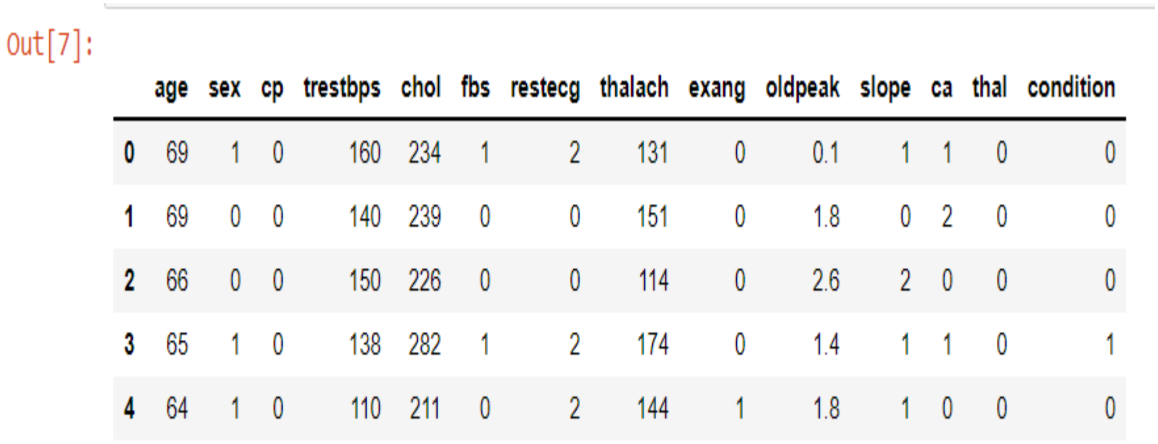


Fig 3.1 Logistic Regression

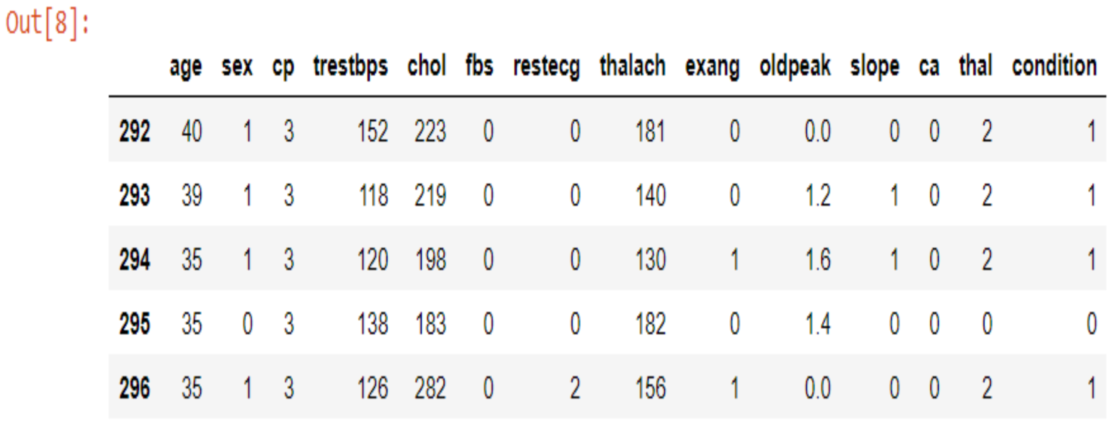
3.3 DATASETS:

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



Out[7]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	condition
0	69	1	0	160	234	1	2	131	0	0.1	1	1	0	0
1	69	0	0	140	239	0	0	151	0	1.8	0	2	0	0
2	66	0	0	150	226	0	0	114	0	2.6	2	0	0	0
3	65	1	0	138	282	1	2	174	0	1.4	1	1	0	1
4	64	1	0	110	211	0	2	144	1	1.8	1	0	0	0



Out[8]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	condition
292	40	1	3	152	223	0	0	181	0	0.0	0	0	2	1
293	39	1	3	118	219	0	0	140	0	1.2	1	0	2	1
294	35	1	3	120	198	0	0	130	1	1.6	1	0	2	1
295	35	0	3	138	183	0	0	182	0	1.4	0	0	0	0
296	35	1	3	126	282	0	2	156	1	0.0	0	0	2	1

Fig 3.2 Original dataset snapshot

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

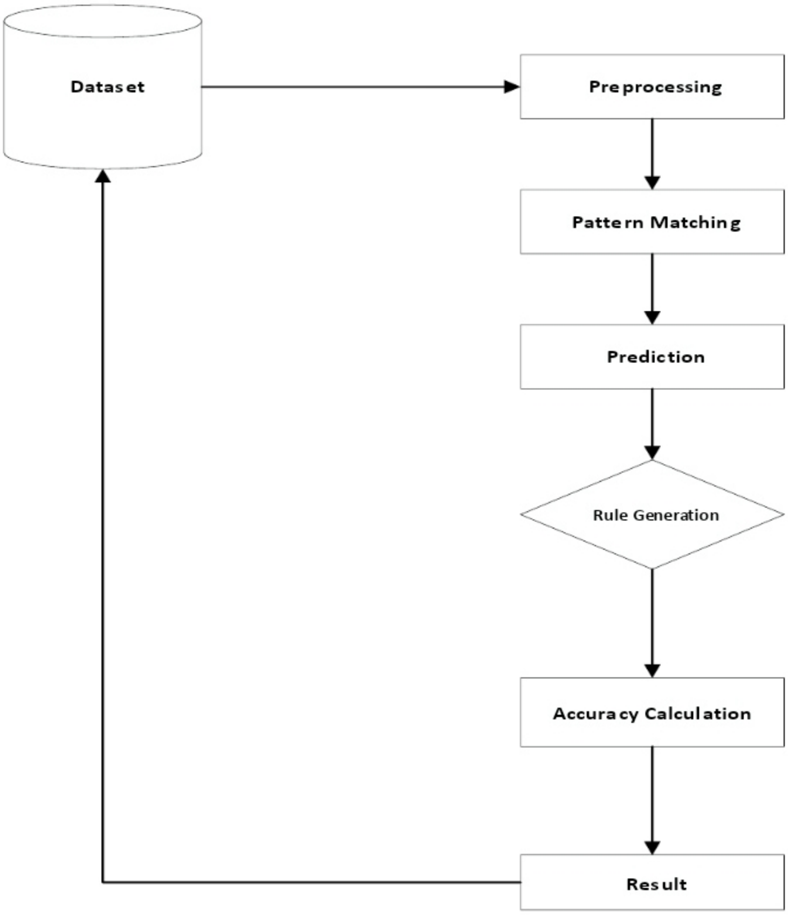
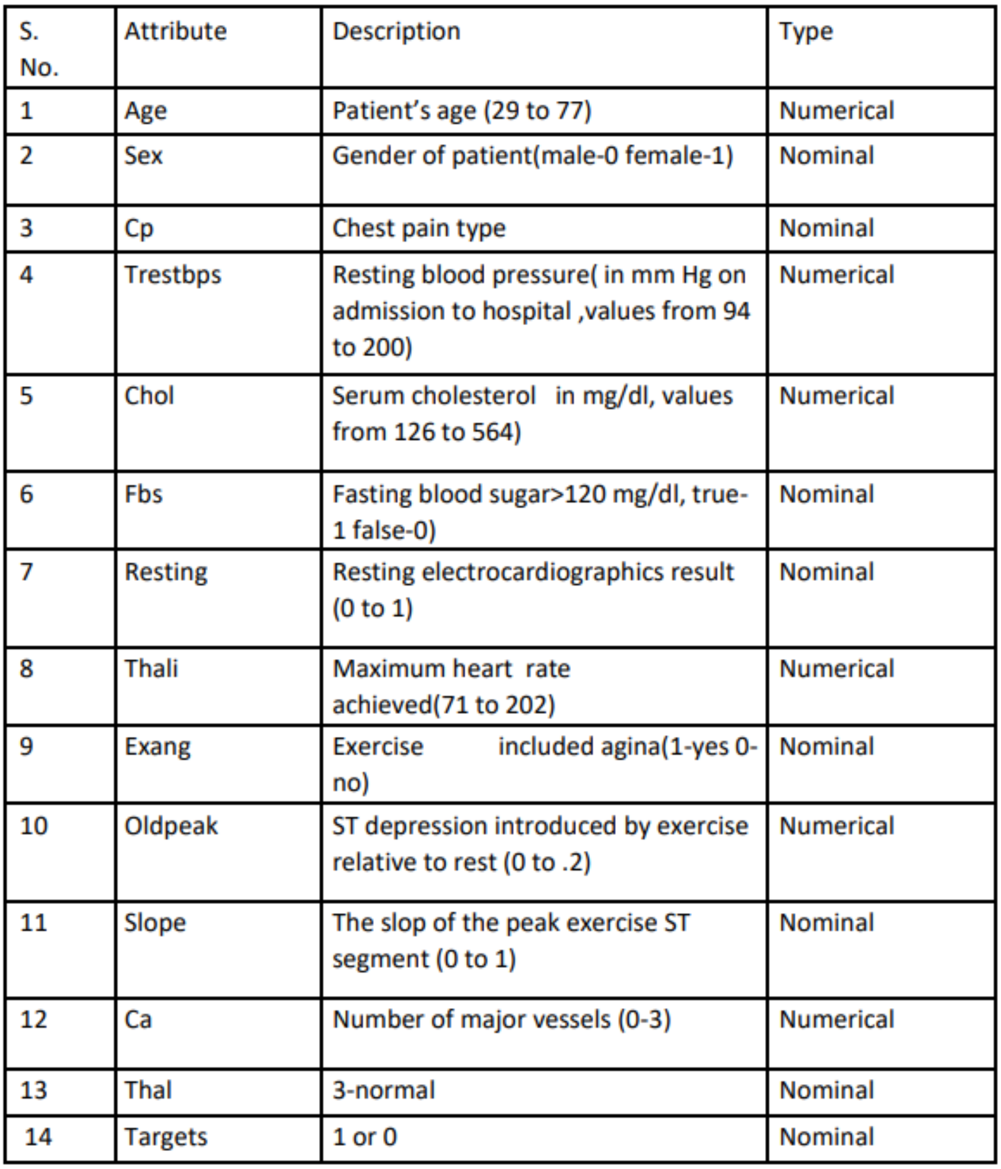


Fig 3.3 Data flow

3.3.1 INPUT DATASET ATTRIBUTES:

- Gender (value 1: Male; value 0 : Female)
- Chest Pain Type (value 1: typical type 1 angina, value 2: typical type angina, valueV3: non-angina pain; value 4: asymptomatic)
- Fasting Blood Sugar (value 1: > 120 mg/dl; value 0:< 120 mg/dl)
- Exang – exercise induced angina (value 1: yes; value 0: no)
- CA – number of major vessels colored by fluoroscopy (value 0 – 3)
- Thal (value 3: normal; value 6: fixed defect; value 7: reversible defect)
- Trest Blood Pressure (mm Hg on admission to the hospital)
- Serum Cholesterol (mg/dl)
- Thalach – maximum heart rate achieved
- Age in Year
- Height in cms
- Weight in Kgs.
- Cholestrol
- Restecg



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

Table 3.4 Attributes of dataset

3.4 TRAINING AND TESTING:

Finally, this resulting data split into 80% train and 20% test data, which was further passed to the Logistic Regression model to fit, predict and score the model.This means that training datasets are an essential part of any ML model. They are necessary to teach the algorithm how to make accurate predictions in accordance with the goals of an AI project.

Just like people learn better from examples, machines also require them to start seeing patterns in the data. Unlike human beings, however, computers need a lot more examples because they do not think in the same way as humans do. They do not see objects in the pictures or cannot recognize people in the photos as we can. They speak their own, programming languages that are structured in a different way. They require substantial work and a lot of data for training a machine learning model to identify emotions from videos.

When you teach a child what a cat is, it's sufficient to show a single picture. If you try teaching a computer to recognize a cat, you'll need to show thousands of images of different cats, in different sizes, colors, and forms, in order for a machine to accurately tell a cat from, say, a dog.

On the other hand, when an ML model is sufficiently sophisticated, it can deliver more accurate results than a human. This may feel counterintuitive but it also has to deal with the differences in how we and the machines process information.

But we'll talk about that a bit later. For now, let's take a dive into other important concepts like testing data, different types of data, and methods of machine learning.

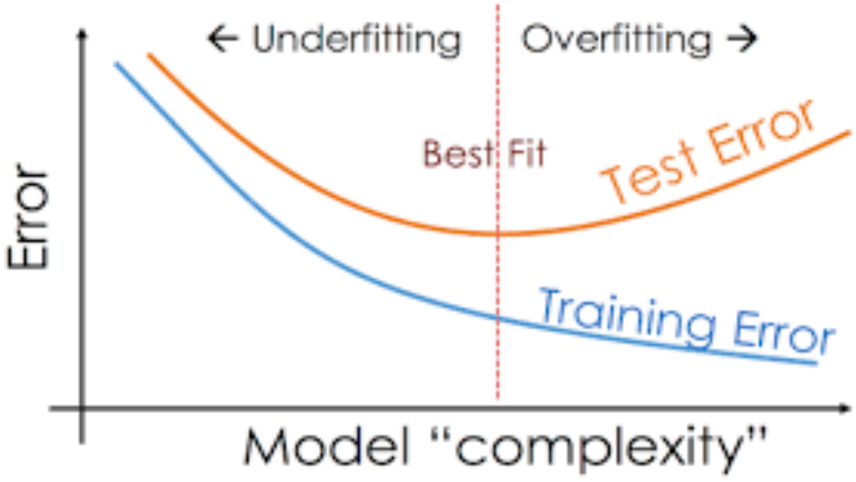


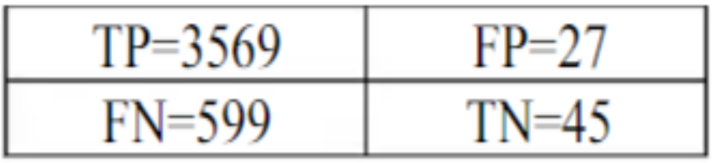
Fig 3.5 Train and Test data split

3.5 EVALUATION METRICS:

For the evaluation of our output from our training the data, the accuracy was analyzed “Confusion matrix”.

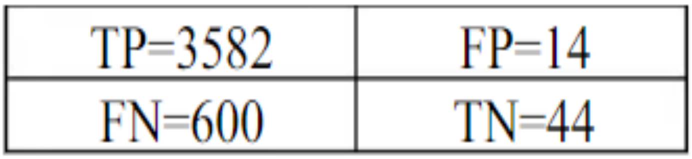
3.5.1 CONFUSION MATRIX:

A confusion matrix, also known as an error matrix, is a table that is often used to describe the performance of a classification model (or “classifier”) on a set of test data for which the true values are known. It allows the visualization of the performance of an algorithm. It allows easy identification of confusion between classes e.g.; one class is commonly mislabeled as the other. The key to the confusion matrix is the number of correct and incorrect predictions are summarized with count values and broken down by each class not just the number of errors made.



TP=3569	FP=27
FN=599	TN=45

Fig 3.6 Confusion matrix obtained after training the data (feature selection by backward elimination)



TP=3582	FP=14
FN=600	TN=44

Fig 3.7 Confusion matrix obtained after training the data (feature selection by RFECV method)

3.6 ACCURACY:

The accuracy is calculated as:

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

Where

- True Positive (TP) =Observation is positive, and is predicted to be positive.
- False Negative (FN) = Observation is positive, but is predicted negative.
- True Negative (TN) = Observation is negative, and is predicted to be negative.
- False Positive (FP) =Observation is negative, but is predicted positive

The obtained accuracy during training the data after feature selection using backward elimination was 86 % and during testing was 83%.

The obtained accuracy during training the data after feature selection using REFCV method was86 % and during testing was 85 %.

3.7 RECALL:

Recall can be defined as the ratio of the total number of correctly classified positive examples divide to the total number of positive examples. High Recall indicates the class is correctly recognized (a small number of FN).

Recall is calculated as:

$$\text{Recall} = \frac{TP}{TP+FN}$$

The obtained recall during training the data after feature selection using backward elimination was and during testing was 0.99.

The obtained recall during training the data after feature selection using REFCV method was 1.00and during testing was 0.99

3.8 PRECISION:

To get the value of precision we divide the total number of correctly classified positive examples by the total number of predicted positive examples. High Precision indicates an example labeled positive is indeed positive (a small number of FP).

Precision is calculated as:

$$\text{Precision} = \frac{TP}{TP+FP}$$

The obtained precision during training the data after feature selection using backward elimination was 0.86 and during testing was 0.84.

The obtained precision during training the data after feature selection using REFCV method and during testing was 0.86.

3.9 SOFTWARE REQUIREMENTS:

Software requirements deal with defining software resource requirements and prerequisites that need to be installed on a computer to provide optimal functioning of an application. These requirements or prerequisites are generally not included in the software installation package and need to be installed separately before the software is installed.

Operating System	:	Windows family
Technology	:	Python3.7, flask, Html, Css
IDE	:	Jupyer notebook
Network	:	Wi-Fi internet or cellular Network

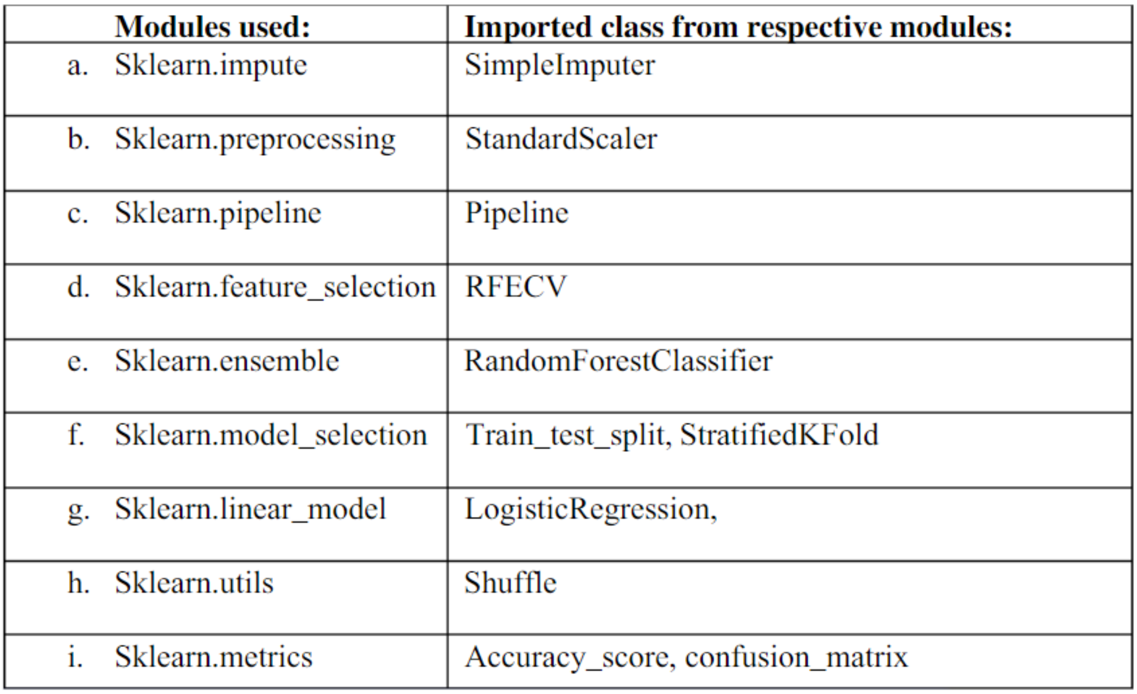
3.10 HARDWARE REQUIREMENTS:

The most common set of requirements defined by an operating system or software application is the physical computer resources, also known as hardware. A hardware requirements list is often accompanied by a hardware compatibility list (HCL), especially in the case of operating systems. An HCL list is tested, compatibility and sometimes incompatible hardware devices for a particular operating system or application. The following lists discuss the various aspects of hardware requirements.

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

3.11 LIBRARIES USED:

- NumPy
- SciPy
- Matplotlib (pyplot, rcparams, matshow)
- Statsmodels
- Pandas
- Tkinter
- Sklearn



Modules used:	Imported class from respective modules:
a. Sklearn.impute	SimpleImputer
b. Sklearn.preprocessing	StandardScaler
c. Sklearn.pipeline	Pipeline
d. Sklearn.feature_selection	RFECV
e. Sklearn.ensemble	RandomForestClassifier
f. Sklearn.model_selection	Train_test_split, StratifiedKFold
g. Sklearn.linear_model	LogisticRegression,
h. Sklearn.utils	Shuffle
i. Sklearn.metrics	Accuracy_score, confusion_matrix

Table 3.8 Modules used

3.12 CODE

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

The code is available in the Git repository given in following link:

https://github.com/Srisairam9881/heart-disease-prediction-using-machine-learning-PT-2

3.12.1 SAMPLE CODE

```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score

df = pd.read_csv("D:\pythonprojects\data.csv")

df.describe()
df.head()
df.tail()
df.shape()


df.info()

df.isnull().sum()

df [‘condition’].value_counts()

x= df.drop(cloumns=’condition’, axis=1)
```

```
y= df[‘condition’]

print(x)

print(y)

X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2, stratify=Y, random_state=2

print(X.shape, X_train.shape, X_test.shape)

model = LogisticRegression()

model.fit(X_train.values, Y_train.values)

X_train_prediction = model.predict(X_train.values)
training_data_accuracy = accuracy_score(X_train_prediction, Y_train.values)
X_test_prediction = model.predict(X_test.values)
test_data_accuracy = accuracy_score(X_test_prediction, Y_test.values)
input_data = (44,1,1,120,160,0,2,160,0,0.8,0,2,0)
input_data_as_numpy_array= np.asarray(input_data)
input_data_reshaped = input_data_as_numpy_array.reshape(1,-1)
prediction = model.predict(input_data_reshaped)
print(prediction)


if (prediction[0]== 0):
  print('The Person does not have a Heart Disease')
else:
  print('The Person has Heart Disease')


from joblib import dump, load
dump(model,'hearthealth.joblib')
```

GUI INDEX HTML CODE:

```
<!DOCTYPE html>
<html lang="en">
<head>
```

```
<meta charset="UTF-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Heart Health ML model</title>
<style>
  .heading{
    background: #C4C4C4;
    text-align: center;
    padding: 0.1rem;
    border-radius: 15px;
  }
  .heading h1{
    text-transform: uppercase;
  }
  .heading{
    text-transform: capitalize;
  }
  .info{
    text-align: center;
    padding: 1rem;
  }
  form{
    background: #C4C4C4;
    width: max-content;
    margin: 0 auto;
    height: max-content;
    padding: 2rem;
    border-radius:15px;
  }
  form section{
    display: grid;
    grid-template-columns: 1fr 3fr;
  }
  .invisible{
```

```
            display: none;
        }
        .visible{
            display: grid;
        }
        form section label{
            justify-self: flex-end;

        }
        form section label::after{
            content: ": "
        }
        form section button, .submit{
            grid-column: span 2;
            width: 50%;
            justify-self: center;
            background: #A93F3F;
            border-radius:5px;
            color:white;
        }
        form section *{
            margin: 10px;
        }
    </style>
</head>
<body>
    <div class="heading">
        <h1><b>Heart Disease Predition</b></h1>
        <p><b>using machine Learning</b></p>
    </div>
    <div class="info">Enter the Details below</div>
    <form action="/result" method="post">
        <section>
            <label for="name">Name</label><input type="text" name="name"
```

```
placeholder="Enter your Name"/>
        <label for="sex">Gender</label><div><input type="radio" name="sex" value="0"/>Male<input type="radio" name="sex" value="1"/>Female</div>
        <label for="age">Age</label><input type="number" name="age" placeholder="Enter your Age"/>
        <button class="next">Next ></button>
    </section>
    <section>
        <label for="cp">CP Type</label><input type="text" name="cp" placeholder="Enter Chest Pain Type 1/2/3/4"/>
        <label for="trestbps">B.P</label><input type="number" name="trestbps" placeholder="Enter your B.P"/>
        <label for="chol">Cholestrol</label><input type="number" name="chol" placeholder="Enter your Cholestrol Level"/>
        <button class="next">Next ></button>
    </section>
    <section>
        <label for="fbs">Blood Sugar</label><input type="text" name="fbs" placeholder="0 for <=120, 1 for >120"/>
        <label for="restecg">E.C.G</label><input type="number" name="restecg" placeholder="Enter the ECG parameters 0/1/2"/>
        <label for="thalach">Max. Heart Rate</label><input type="number" name="thalach" placeholder="Enter your maximum heart rate"/>
        <label for="ca">Flouroscopy</label><input type="number" name="ca" placeholder="No. of major vessels (0-3) colored by flourosopy"/>
        <button class="next">Next ></button>
    </section>
    <section>
        <label for="exang">E.I.A.</label><div><input type="radio" name="exang" value="0"/>Yes<input type="radio" name="exang" value="1"/>No</div>
        <label for="oldpeak">Old Peak</label><input type="number" step="0.1" name="oldpeak" placeholder="ST depression induced by exercise relative to rest"/>
        <label for="thal">Thal.</label><div><input type="radio" name="thal" value="0"/>Normal<input type="radio" name="thal" value="1"/>Fixed<input type="radio" name="thal" value="2"/>Reversible</div>
        <label for="slope">Slope</label><div><input type="radio" name="slope" value="0"/>0<input type="radio" name="slope" value="1"/>1<input type="radio" name="slope" value="2">2</div>
        <input type="submit" class="submit" value="Next >">
      </section>
    </form>
    <script>
      let sections = document.querySelectorAll('section');
      for(let section of [...sections].slice(1)){
        section.classList.add('invisible')
      }
      let nexts = document.querySelectorAll('.next');
      for(let i=0;i<nexts.length;i++){
        console.log(nexts[i]);
        nexts[i].addEventListener('click',(e)=>{
          e.preventDefault()
          sections[i].classList.remove('visible');
          sections[i].classList.add('invisible');
          sections[i+1].classList.remove('invisible');
          sections[i+1].classList.add('visible')
        })
      }
    </script>
</body>
</html>
```

SUCCESS HTML CODE

```
<!DOCTYPE html>
<html lang="en">
```

```
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Congrats!!!</title>
  <style>
    .heading{
      background: #C4C4C4;
      text-align: center;
      padding: 0.1rem;
      height: 50vh;
      border-radius:15px;
    }
    .heading h1{
      text-transform: uppercase;
    }
    .heading{
      text-transform: capitalize;
    }
    .heading a{
      position: relative;
      background-color: #A93F3F;
      border-radius:10px;
      color:white;
      padding:1rem;
      text-decoration: none;
      top:20vh;
    }
  </style>
</head>
<body>
  <div class="heading">
    <h1 style="font-family:Arial, Helvetica, sans-
serif"><b>Congragulations</b></h1>
```

```
        <p style="font-family:Arial, Helvetica, sans-serif">you have no risk of developing a heart disease</p>
        <a href="/">Go To Home</a>
    </div>
</body>
</html>
```

FAILURE HTML CODE:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Sorry!!!</title>
    <style>
       .heading{
            background: #C4C4C4;
            text-align: center;
            padding: 0.1rem;
            height: 50vh;
            border-radius:10px;
        }
        .heading h1{
            text-transform: uppercase;
        }
        .heading{
            text-transform: capitalize;
        }
        .heading a{
            position: relative;
            background-color: #A93F3F;
            border-radius:10px;
            color:white;
            padding:1rem;
            text-decoration: none;
            top:20vh;
        }
    </style>
</head>
<body>

    <div class="heading">
        <h1 style="font-family:Arial, Helvetica, sans-serif"><b>Sorry!!!</b></h1>
        <p style="font-family:Arial, Helvetica, sans-serif">You have a risk of Developing a Heart Disease</p>
        <p style="font-family:Arial, Helvetica, sans-serif">please consult your doctor</p>
        <a href="/">Go to home</a>
    </div>
</body>
</html>
```

CHAPTER 4

RESULTS AND DISCUSSION, PERFORMANCE ANALYSIS

4.1 RESULTS:

When performing various methods of feature selection, testing it was found that backward elimination gave us the best results among others. The various methods tried were Backward Elimination with and without KFold, Recursive Feature Elimination with Cross Validation. The accuracy that was seen in them ranged around 85% with 85.5% being maximum. Though both methods gave similar accuracy but it was seen that in Backward Elimination we found that the number of misclassifications of True Negative was more and it was observed that the accuracy had more variance compared to RFEV. The precision of Backward Elimination and RFEV are 84%and 86% respectively. And the recalls are 0.99 and 1 respectively. The precision and recall also shows that the number of misclassifications is less in RFECV than in Backward Elimination.

4.2 FEASIBILITY STUDY:

The main goal of Machine Learning (ML) feasibility studies is to assess whether it is feasible to solve the problem satisfactorily using ML with the available data. We want to avoid investing too much in the solution before we have.

- Sufficient evidence that an ML solution would be the best technical solution given the business case
- Sufficient evidence that an ML solution is possible
- Some vetted direction on what an ML solution should look like

This effort ensures quality solutions backed by the appropriate, thorough amount of consideration and evidence.

Every engagement with an ML component, potentially excluding pure ML Ops engagements, can benefit from an ML feasibility study early in the project.

Architectural discussions can still occur in parallel as the team works towards a gaining solid understanding and definition of what will be built.

Feasibility studies can last between 3-12 weeks, depending on specific problem details, volume of data, state of the data etc. Starting with a 3-week milestone might be useful, during which it can be determined how much more time, if any, is required for completion.

4.3 IMPLEMENTATION:

- Collect data
- Prepare the data
- Choose the model
- Train your machine model
- Evaluation
- Parameter Tuning
- Prediction or inference

CHAPTER 5

SUMMARY AND CONCLUSIONS

SUMMARY:

Machine learning:

Machine learning is a branch of AI. Other tools for reaching AI include rule-based engines, evolutionary algorithms, and Bayesian statistics. While many early AI programs, like IBM's Deep Blue, which defeated Garry Kasparov in chess in 1997, were rule-based and dependent on human programming, machine learning is a tool through which computers have the ability to teach themselves, and set their own rules. In 2016, Google's DeepMind beat the world champion in Go by using machine learning–training itself on a large data set of expert moves.

There are several kinds of machine learning:

- In supervised learning, the "trainer" will present the computer with certain rules that connect an input (an object's feature, like "smooth," for example) with an output (the object itself, like a marble).
- In unsupervised learning, the computer is given inputs and is left alone to discover patterns.
- In reinforcement learning, a computer system receives input continuously (in the case of a driverless car receiving input about the road, for example) and constantly is improving.

CONCLUSION:

The early prognosis of cardiovascular diseases can aid in making decisions on lifestyle changes in high-risk patients and in turn reduce the complications, which can be a great milestone in the field of medicine. This project resolved the feature selection i.e., backward elimination and RFECV behind the models and successfully predict the heart disease, with 86% accuracy. The model used was Logistic Regression. Further for its enhancement, we can train on models and predict the types of cardiovascular diseases providing recommendations to the users, and also use more enhanced models.

REFERENCES

BOOKS:

- Artificial intelligence: A modern_approach.
- Deep Learning.
- The hundred-page machine learning book.
- The element of statical learning: data miming, inference, and prediction.
- Applied predictive modeling
- Python machine learning

WEBSITES:

- https://www.learndatasci.com/best-machine-learning-courses/
- https://machinelearningmastery.com/start-here/
- https://www.openml.org/
- https://www.quora.com/What-are-the-best-websites-to-learn-machine-learning-and-deep-learning-on-my-computer-for-free
- https://medium.com/javarevisited/6-best-websites-to-learn-data-science-and-machine-learning-36f014af63fb

APPENDIX

PYTHON

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

SKLEARN

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

NUMPY

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

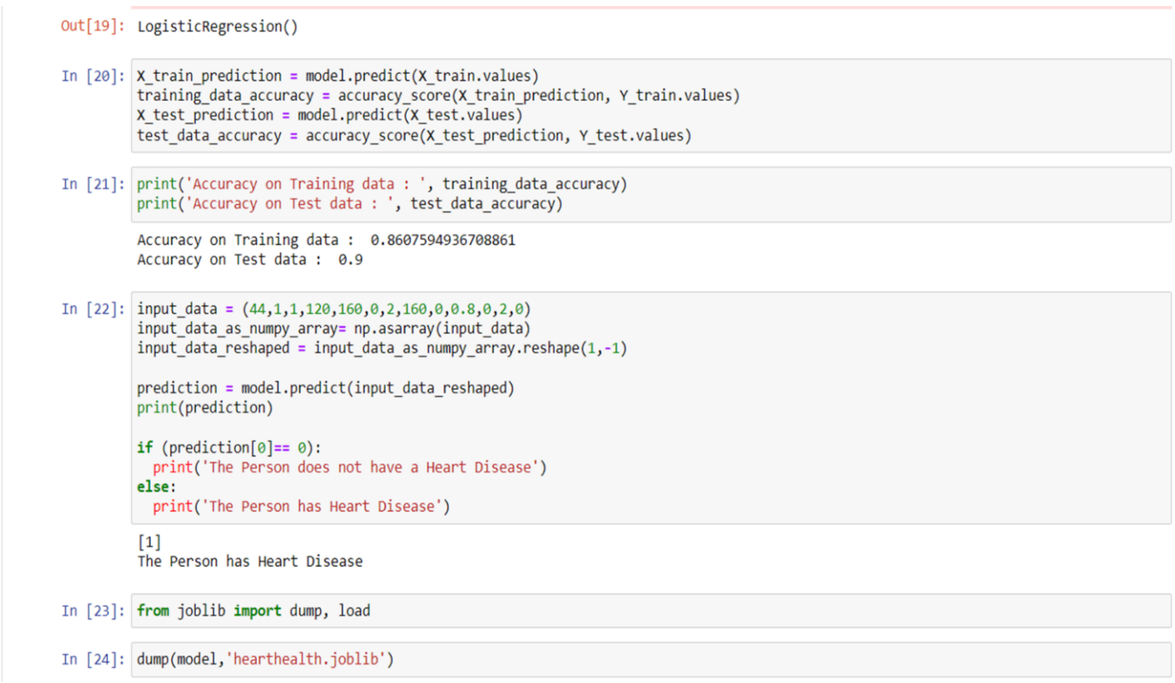
MATPLOTLIB

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

SCIPY

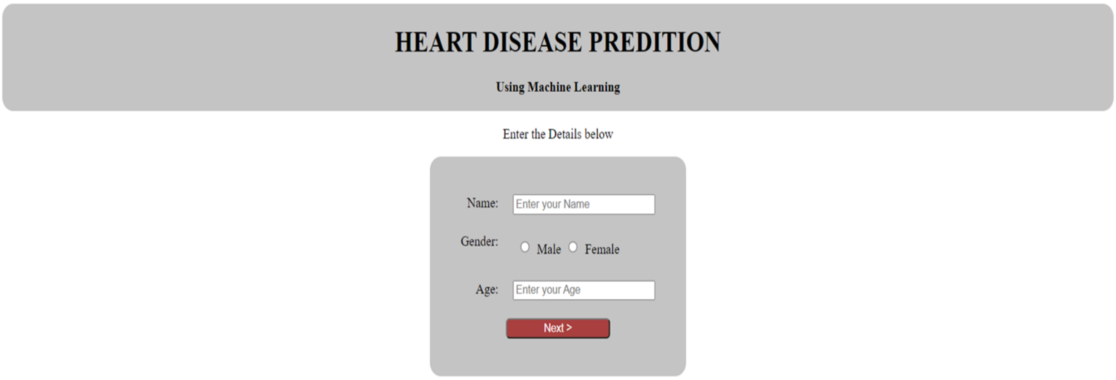
SciPy contains modules for optimization, linear algebra, integration, interpolation, special functions, FFT, signal and image processing, ODE solvers and other tasks common in science and engineering. SciPy is also a family of conferences for users and developers of these tools: SciPy (in the United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

A. SCREEN SHOTS



```
Out[19]: LogisticRegression()

In [20]: X_train_prediction = model.predict(X_train.values)
         training_data_accuracy = accuracy_score(X_train_prediction, Y_train.values)
         X_test_prediction = model.predict(X_test.values)
         test_data_accuracy = accuracy_score(X_test_prediction, Y_test.values)

In [21]: print('Accuracy on Training data : ', training_data_accuracy)
         print('Accuracy on Test data : ', test_data_accuracy)

         Accuracy on Training data :  0.8607594936708861
         Accuracy on Test data :  0.9

In [22]: input_data = (44,1,1,120,160,0,2,160,0,0.8,0,2,0)
         input_data_as_numpy_array= np.asarray(input_data)
         input_data_reshaped = input_data_as_numpy_array.reshape(1,-1)

         prediction = model.predict(input_data_reshaped)
         print(prediction)

         if (prediction[0]== 0):
           print('The Person does not have a Heart Disease')
         else:
           print('The Person has Heart Disease')

         [1]
         The Person has Heart Disease

In [23]: from joblib import dump, load

In [24]: dump(model,'hearthealth.joblib')
```

Fig 5.1 Accuracy



HEART DISEASE PREDITION

Using Machine Learning

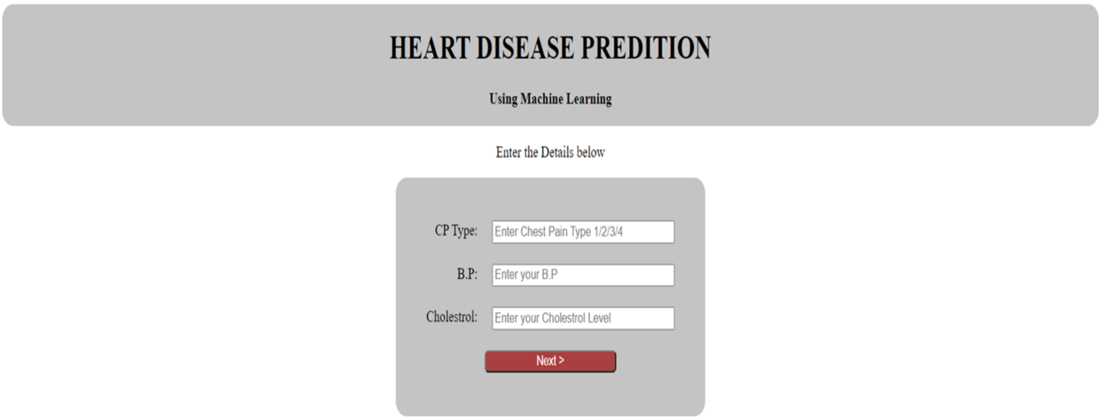
Enter the Details below

Name: Enter your Name

Gender: ○ Male ○ Female

Age: Enter your Age

Fig 5.2 Index 1

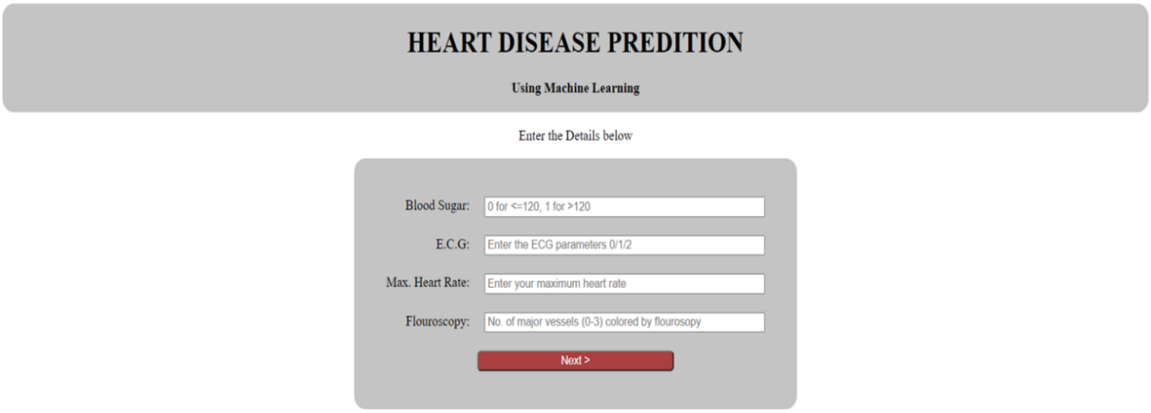


HEART DISEASE PREDITION

Using Machine Learning

Enter the Details below

Fig 5.3 Index 2



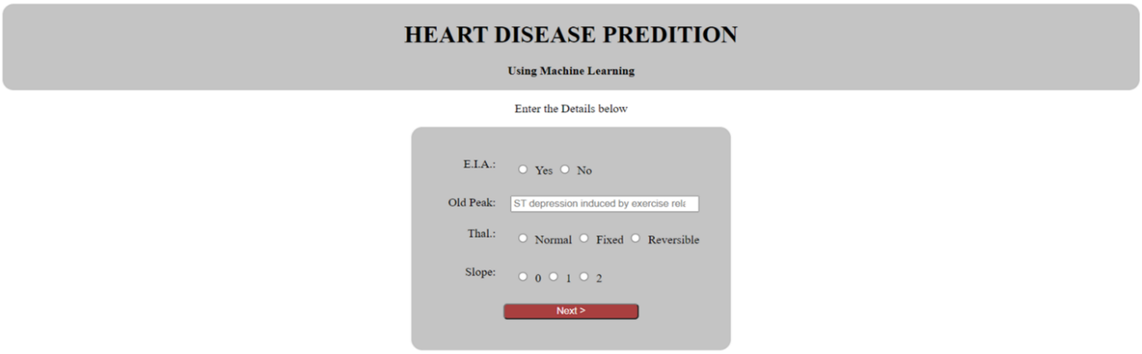
HEART DISEASE PREDITION

Using Machine Learning

Enter the Details below

Blood Sugar:
0 for <=120, 1 for >120
E.C.G:
Enter the ECG parameters 0/1/2
Max. Heart Rate:
Enter your maximum heart rate
Flouroscopy:
No. of major vessels (0-3) colored by flourosopy
Next >

Fig 5.4 Index 3



HEART DISEASE PREDITION

Using Machine Learning

Enter the Details below

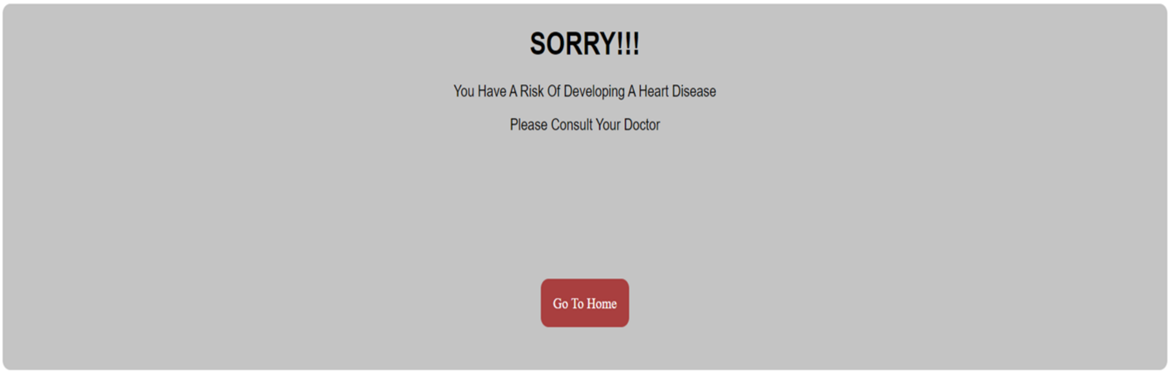
E.I.A.: ○ Yes ○ No

Old Peak: ST depression induced by exercise rel

Thal.: ○ Normal ○ Fixed ○ Reversible

Slope: ○ 0 ○ 1 ○ 2

Fig 5.5 Index 4

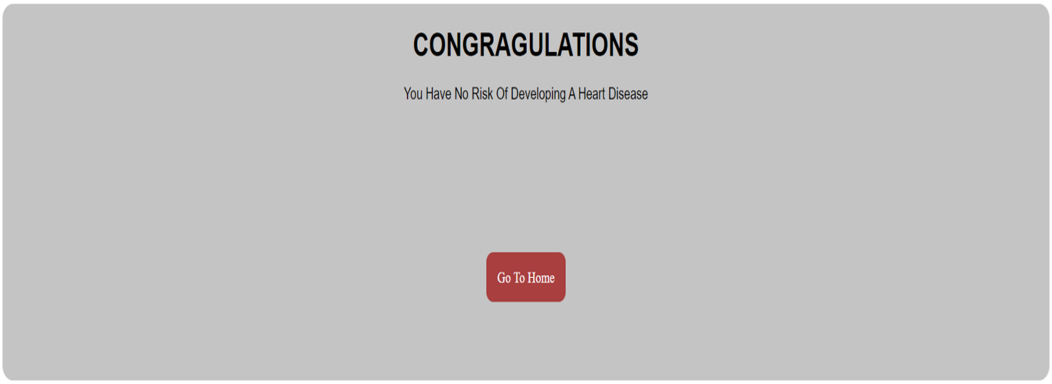


SORRY!!!

You Have A Risk Of Developing A Heart Disease

Please Consult Your Doctor

Fig 5.6 Failure



CONGRAGULATIONS

You Have No Risk Of Developing A Heart Disease

Fig 5.7 Success