ABSTRACT

- **Heart related diseases or Cardiovascular Diseases (CVDs) are the main reason for a huge number of death in the world over the last few decades and has emerged as the most life-threatening disease, not only in India but in the whole world.**
- **So, there is a need of reliable, accurate and feasible system to diagnose such diseases in time for proper treatment.**
- **Machine Learning algorithms and techniques have been applied to various medical datasets to automate the analysis of large and complex data.**
- **Many researchers, in recent times, have been using several machine learning techniques to help the health care industry and the professionals in the diagnosis of heart related diseases.**
- **This paper presents a survey of various models based on such algorithms and techniques and analyzes their performance.**
- **Models based on supervised learning algorithms such as Support Vector Machines (SVM), K-Nearest Neighbour (KNN), XGBoost, Light GBM, Decision Trees (DT), Random Forest (RF) and ensemble models are found very popular among the researchers.**

CONTENTS

CHAPTER NAME	PAGE NUMBER
1. INTRODUCTION	1
2. LITERATURE SURVEY	3
3. SYSTEM ANALYSIS	6
3.1 Existing System	6
3.2 Proposed System	6
3.3 System Study	8
4. SYSTEM REQUIREMENTS	9
4.1 Hardware Requirements	9
4.2 Software Requirements	9
5. SYSTEM DESIGN	10
5.1 System Architecture	10
5.2 UML Diagram	10
5.3 Use Case Diagram	11
5.4 Class Diagram	11
5.5 Sequence Diagram	12
5.6 Activity Diagram	13
6. IMPLEMENTATION	14
6.1 Modules	14
6.2 Modules Description	14
7. SOFTWARE ENVIRONMENT	15
7.1 History of Python	15
7.2Advantages of Python	16
7.3 Disadvantages of Python	17
7.4 What is Machine Learning?	18
7.5 Categories of Machine Learning	18

CHAPTER NAME	PAGE NUMBER
7.6 Applications of Machine Learning	19
7.7 Need of Machine Learning	19
7.8 Challenges in Machine Learning	20
7.9 How to start Machine Learning?	21
7.10 Advantages of Machine Learning	23
7.11 Disadvantages of Machine Learning	23
7.12 Modules used in Project	24
8. SAMPLE CODE	25
8.1 System Test	30
8.2 Types of Tests	30
9. OUTPUT SCREENS	32
10. CONCLUSION	35
11. REFERENCES	36

LIST OF FIGURES

FIGURE NUMBER	FIGURE NAME	PAGE NUMBER
3.2.1	CNN	7
3.2.2	Multiple Linear Regression	7
5.1	System Architecture	10
5.3	Use Case Diagram	11
5.4	Class Diagram	12
5.5	Sequence Diagram	12
5.6	Activity Diagram	13
9.1	Run.bat File	32
9.2	URL	32
9.3	User Login	32
9.4	Heart Disease Prediction System	33
9.5	Train Machine Learning Algorithm	33
9.6	Enter Details	33
9.7	Prediction Screen-1	34
9.8	Heart Disease Prediction-1	34
9.9	Prediction Screen-2	34
9.10	Heart Disease Prediction-2	34

1. INTRODUCTION

CHAPTER 1

Heart disease is a kind of disease which effects the functioning of the heart. In today's era heart disease is the primary reason for deaths. WHO-World Health Organization has anticipated that 12 million people die every year because of heart diseases. Some heart diseases are cardiovascular, heart attack, coronary and knock. Knock is a sort of heart disease that occurs due to strengthening, blocking or lessening of blood vessels which drive through the brain or it can also be initiated by high blood pressure.

The major challenge that the Healthcare industry faces now-a-days is superiority of facility. Diagnosing the disease correctly & providing effective treatment to patients will define the quality of service. Poor diagnosis causes disastrous consequences that are not accepted. The contents of this paper mainly focus on various data mining practices that are valuable in heart disease forecast with the assistance of dissimilar data mining tools that are accessible. If the heart doesn't function properly, this will distress the other parts of the human body such as brain, kidney etc.

Records or data of medical history is very large, but these are from many dissimilar foundations. The interpretations that are done by physicians are essential components of these data. The data in real world might be noisy, incomplete and inconsistent, so data pre-processing will be required in directive to fill the omitted values in the database. Even if cardiovascular diseases is found as the important source of death in world in ancient years, these have been announced as the most avoidable and manageable diseases. The whole and accurate management of a disease rest on the well-timed judgment of that disease.

Different person body can show different symptoms of heart disease which may vary accordingly. Though, they frequently include back pain, jaw pain, neck pain, stomach disorders and tininess of breath, chest pain, arms and shoulders pains. There are a variety of different heart diseases which includes heart failure and stroke and coronary artery disease. Even though heart disease is acknowledged as the supreme chronic sort of disease in the world, it can be most avoidable one also at the same time. A healthy way of life (main prevention) and timely analysis (inferior prevention) are the two major origins of heart disease director.

Heart expert's create a good and huge record of patient's database and store them. It also delivers a great prospect for mining a valued knowledge from such sort of datasets. Researchers make use of several data mining techniques that are accessible to help the specialists or physicians identify the heart disease. Commonly used procedures used are decision tree, k-nearest and Naïve Bayes. Other different classification based techniques used are bagging algorithm, kernel density, sequential minimal optimization and neural networks, straight Kernel self-organizing map and SVM (Support Vector Machine). The next section clearly provides details of techniques that were used in the study.

The diseases that come under cardiovascular disease are coronary heart disease (CHD), cerebrovascular disease (Stroke), congenital heart disease, provocative heart diseases, Hypertensive heart diseases, and exterior artery disease. Among them, the tobacco chewing, unhealthy diet, physical inactivity and alcohol are the primary cause of heart diseases. Researchers are using a variety of classes of mathematical data mining tools that are existing in the study of heart diseases.

2. LITERATURE SURVEY

CHAPTER 2

In year 2000, research conducted by ShusakuTsumoto says that as we human beings are unable to arrange data if it is huge in size we should use the data mining techniques that are available for finding different patterns from the available huge database and can be used again for clinical research and perform various operations on it.

Y. Alp Aslandogan (2004), worked on three different classifiers called K-nearest Neighbour (KNN), Decision Tree, Naïve Bayesian and used Dempsters' rule for this three viewpoint to appear as one concluding decision. This classification based on the combined idea show increased accuracy. Carlos Ordonez (2004), Assessed the problematic to recognize and forecast the rule of relationship for the heart disease. A dataset involving medical history of the patients having heart disease with the aspects of risk factors was accessed by him, measurements of narrowed artery and heart perfusion. All these restrictions were announced to shrink the digit of designs, these are as follows:

1) The features should seem on a single side of the rule.
2) The rule should distinct various features into the different groups.
3) The count of features available from the rule is organized by medical history of people having heart disease only. The occurrence or the nonappearance of heart disease was predicted by the author in four heart veins with the two clusters of rules.

Franck Le Duff (2004), worked on creating Decision tree quickly with clinical data of the physician or service. He suggested few data mining techniques which can help cardiologists in the predication survival of patients.

Boleslaw Szymanski, et. al. (2006), operated on a novel experiential to check the aptitude of calculation of scarce kernel in SUPANOVA. The author used this technique on a standard boston housing market dataset for discovering heart diseases, measurement of heart activities and prediction of heart diseases were found 83.7% correct which were measured with the help of support vector machine and kernel equivalent to it. A quality result is gained by spline kernel with the help of standard boston housing market database.

HeonGyu Lee, et. al. (2007), operated for the operation systems of Arithmetical and cataloguing for the addition chief of the multi-parametric feature through direct and nonlinear features of Heart Rate Variability (HRV). The dissimilar classifiers existing are cataloguing grounded on Decision Tree (C4.5), Multiple Association Rules (CMAR) and Bayesian classifiers, and Support Vector Machine (SVM) that are investigated for the valuation of the linear and nonlinear features of the HRV tables.

Niti Guru, et. al. (2007) functioned for forecasting of heart disease, Blood Stress and Sugar by the aid of neural systems. Hearings were accepted out on example best ever of patients. The neural system is verified with 13 types, as blood pressure, period, angiography etc.

Controlled network was used for analysis of heart diseases. Training was accepted out with the support of a back-propagation technique. The secretive data was nourished at certain times by the doctor; the acknowledged technique applied on the unidentified data since the judgments with trained data and caused a grade of possible ailments that the patient is inclining to heart disease. Hai Wang, et al. (2008), deliberated the part of medicinal experts in medical data mining also on obtaining a model for medical awareness achievement using data mining.

Sellappan Palaniappan, et. al. (2008), industrialized IHDPS-Intelligent Heart Disease Prediction System by means of data mining algorithm, i.e. Naïve Bayes, Decision Trees and Neural Network. Each process has its own authority to advance right results. The unknown designs and association amongst them have were used to paradigm this method. The IHDPS is web-based, user-friendly, mountable, trustworthy and stretchy and justifiable.

LathaParthiban, et. al. (2008), operated on the foundation of CANFIS (co-active neuro-fuzzy implication method) for identification of heart disease. CANFIS model established the disease by integrating the neural network and fuzzy logic methods and later combined with the genetic algorithm. On the grounds of the training presentations and classification correctness found, the performance of the CANFIS model were estimated. The CANFIS prototypical is exposed as the possible for estimation of heart disease.

Chaitrali S. D., (2012), investigated a computation structures for heart syndrome with the help of full amount of input characteristics. A few terms related to medical like blood pressure, sex, cholesterol and 13 more attributes like this were recycled to predict the heart disease to a particular person or patient. He also made use of two different attributes like smoking and obesity. Unlike data mining performances were used like Decision trees, neural networks and naïve baye's for analyzing the heart disease database. The concert of these practices depends on the accuracy provided by the system. The accuracy provided by decision tree is 99.62%, neural network is 100% and naïve bayes is 90.74% respectively.

S. Vijayarani, et. al. in (2013), made use of experimental results carried out using dissimilar classification methods for heart disease dataset. The different classification systems which were used and tested by him are Decision Stump, Random Forest and LMT tree algorithm. WEKA tool was used for comparison.

3. SYSTEM ANALYSIS

CHAPTER 3

3.1 EXISTING SYSTEM

Heart related infections or Cardiovascular Diseases (CVDs) are the primary justification a colossal number of death on the planet in the course of the most recent couple of many years and has arisen as the most perilous illness, in India as well as in the entire world. Along these lines, there is a need of dependable, precise and attainable framework to analyze such sicknesses on schedule for legitimate therapy. AI calculations and methods have been applied to different clinical datasets to robotize the investigation of enormous and complex information. Numerous scientists, as of late, have been utilizing a few AI procedures to help the medical care industry and the experts in the analysis of heart related illnesses.

3.2 PROPOSED SYSTEM

We smooth out AI calculations for powerful expectation of constant illness episode in disease-continuous networks. We try the adjusted expectation models over genuine medical clinic data collected from focal China in 2013–2015. To conquer the trouble of deficient information, we utilize a latent factor model to remake the missing information. We investigate a territorial persistent infection of cerebral infarction. We propose another convolution neural organization (CNN) - based multimodal infection hazard prediction algorithm utilizing organized and unstructured information from medical clinic. The multiple layers of CNN de noise the inputs from various sources and Multiple Linear Regression Technique of Supervised Machine Learning uses a number of independent input parameters to yield the probabilistic output with the help of the regression line. The typical output of the system will be the Probability of a person of getting affected by heart disease. As far as we could possibly know, none of the existing work zeroed in on both information types nearby clinical huge information investigation.

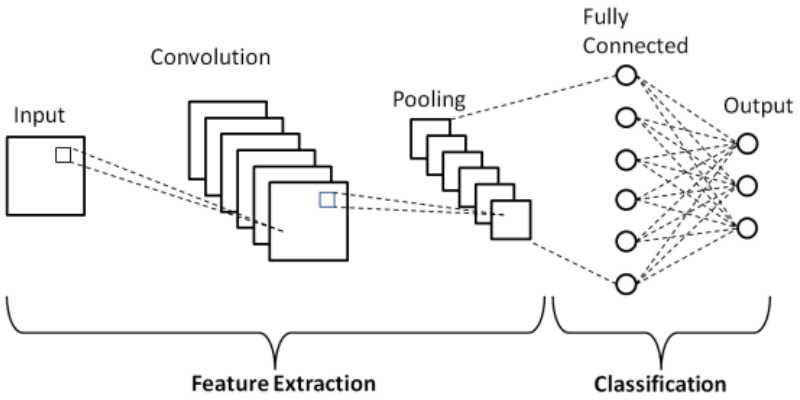

Fig 3.2.1 CNN

Here CNN performs the feature extractions and prepares the classified input data.

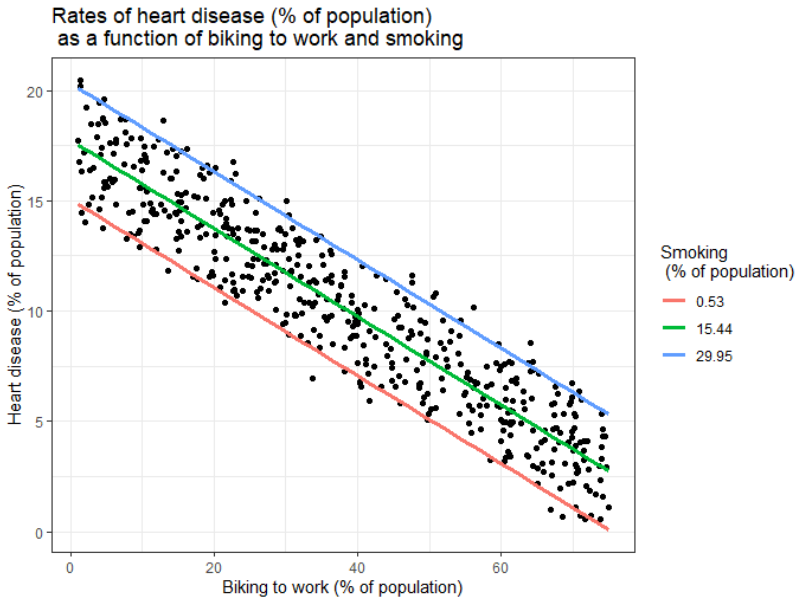

Fig 3.2.2 Multiple Linear Regression

Multiple Linear Regression in this example predicts % of population affected by Heart Disease or the rate of Heart disease as a result (function) of 2 sample parameters. In our problem Multiple Regression predicts the probability (%) of a given person on the basis of (as a function of) a number of different parameters with the help of multiple regression lines.

3.3 SYSTEM STUDY

FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

- ECONOMICAL FEASIBILITY
- TECHNICAL FEASIBILITY
- SOCIAL FEASIBILITY

ECONOMICAL FEASIBILITY

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used .

TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

SOCIAL FEASIBILITY

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system.

4. SYSTEM REQUIREMENTS

CHAPTER 4

4.1 HARDWARE REQUIREMENTS

- **System** : MINIMUM i3.
- **Hard Disk** : 40 GB.
- **Ram** : 4 GB.

4.2 SOFTWARE REQUIREMENTS

- **Operating System:** Windows 8
- **Coding Language**: Python 3.7

5. SYSTEM DESIGN

CHAPTER 5

5.1 SYSTEM ARCHITECTURE

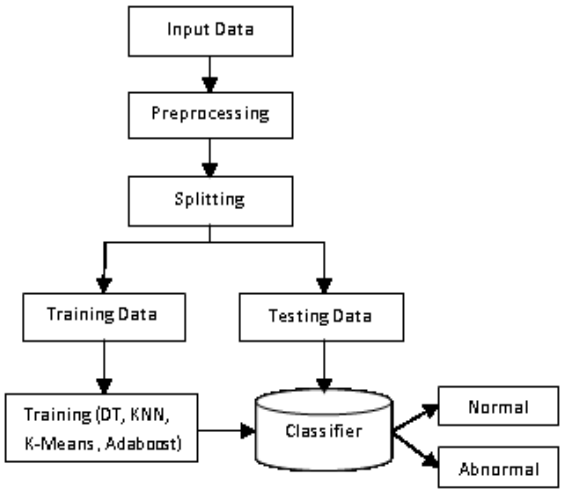

Fig 5.1System Architecture

5.2 UML DIAGRAMS

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems. The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems.

GOALS:

1. Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.
2. Provide extendibility and specialization mechanisms to extend the core concepts.
3. Be independent of particular programming languages and development process.
4. Provide a formal basis for understanding the modeling language.
5. Encourage the growth of OO tools mark

5.3 USE CASE DIAGRAM

A use case diagram in the Unified Modeling Language (UML) is a type of behavioural diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

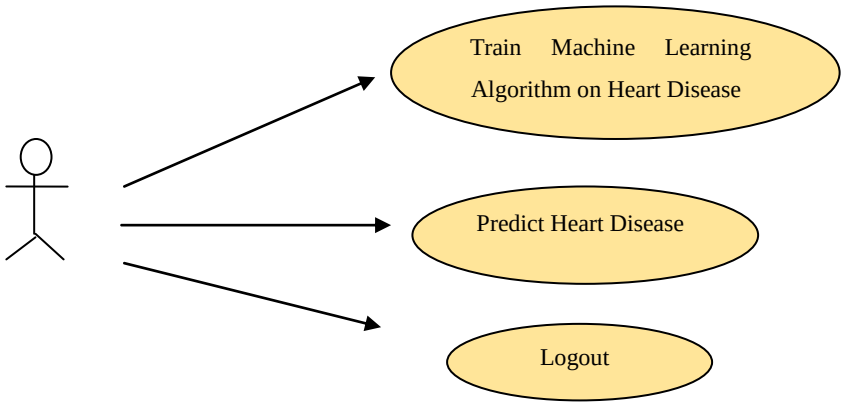

Fig 5.3 Use Case Diagram

5.4 CLASS DIAGRAM

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.

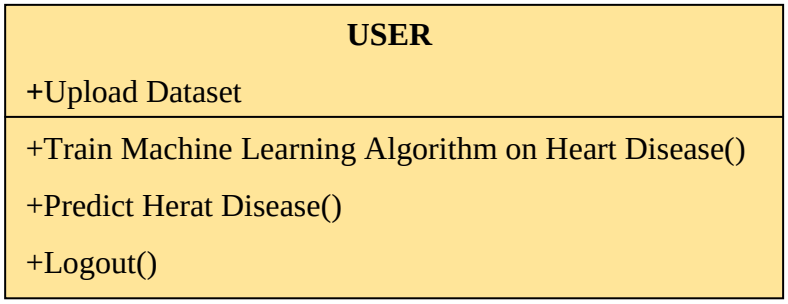

Fig 5.4 Class Diagram

5.5 SEQUENCE DIAGRAM

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

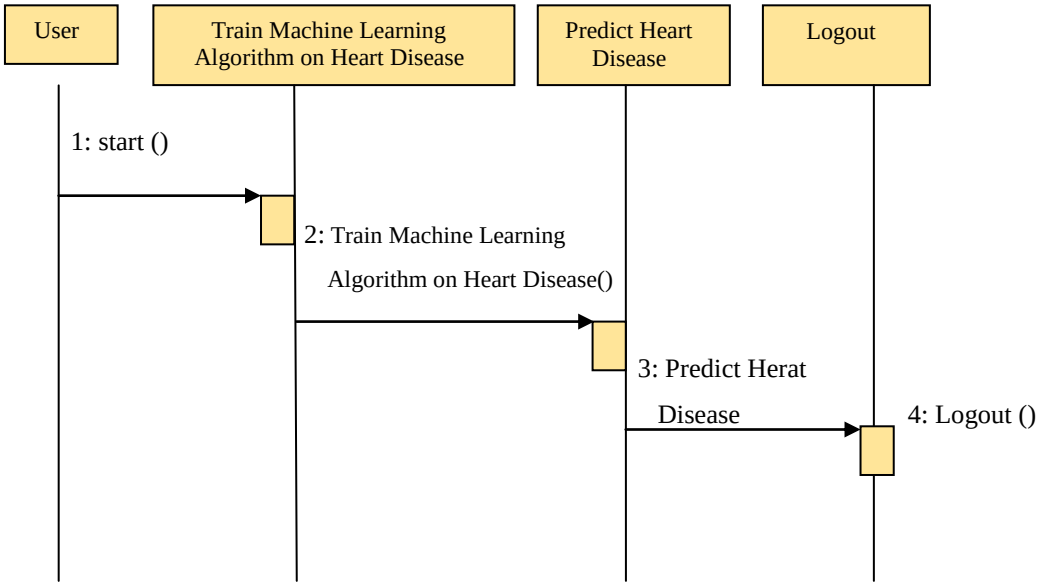

Fig 5.5 Sequence Diagram

5.6 ACTIVITY DIAGRAM

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

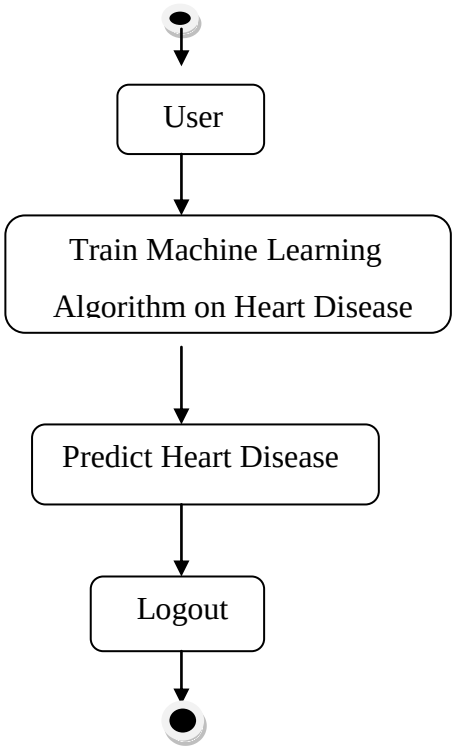

Fig 5.6 Activity Diagram

6. IMPLEMENTATION:

CHAPTER 6

6.1 MODULES

Train Machine Learning Algorithm on Heart Disease

Predict Heart Disease

Logout

6.2 MODULES DESCRIPTION

Train Machine Learning Algorithm on Heart Disease

This focuses on training the algorithms to obtain the highest accuracy for the prediction. Here the four algorithms are used to train them and the results are predicted on different dataset.

Predict Heart Disease

Prediction Involves preparing the raw data and making it suitable for a machine learning model called as data pre-processing. According to the different dataset i.e; persons data values the prediction is done. The dataset includes age, Cholesterol, Blood pressure etc.

Logout

After the prediction according the dataset the heart disease is predicted with the probability rate of heart disease. With the help of such system the heart disease can be taken into control. The different dataset values are predicted with different probabilities of heart disease.

7. SOFTWARE ENVIRONMENT

CHAPTER 7

What is Python:-

Python is currently the most widely used multi-purpose, high-level programming language. Python allows programming in Object-Oriented and Procedural paradigms. Python programs generally are smaller than other programming languages like Java. Programmers have to type relatively less and indentation requirement of the language, makes them readable.

The biggest strength of Python is huge collection of standard library which can be used for the following –

- Machine Learning
- GUI Applications (like Kivy, Tkinter, PyQt etc.)
- Web frameworks like Django (used by YouTube, Instagram, Dropbox)
- Image processing (like Opencv, Pillow)
- Web scraping (like Scrapy, BeautifulSoup, Selenium)
- Test frameworks
- Multimedia

7.1 HISTORY OF PYTHON

What do the alphabet and the programming language Python have in common? Right, both start with ABC. If we are talking about ABC in the Python context, it's clear that the programming language ABC is meant. ABC is a general-purpose programming language and programming environment, which had been developed in the Netherlands, Amsterdam, at the CWI (Centrum Wiskunde &Informatica). The greatest achievement of ABC was to influence the design of Python. Python was conceptualized in the late 1980s. Guido van Rossum worked that time in a project at the CWI, called Amoeba, a distributed operating system.

7.2 ADVANTAGES OF PYTHON

- Extensive Libraries: Python downloads with an extensive library and it contain code for various purposes like regular expressions, documentation-generation, unit-testing, web browsers, threading, databases, CGI, email, image manipulation, and more.

- Extensible: As we have seen earlier, Python can be extended to other languages. You can write some of your code in languages like C++ or C. This comes in handy, in projects.

- Embeddable: Complimentary to extensibility, Python is embeddable as well. You can put your Python code in your source code of a different language, like C++. This lets us add scripting capabilities to our code in the other language.

- Improved Productivity: The language's simplicity and extensive libraries render programmers more productive than languages like Java and C++ do. Also, the fact that you need to write less and get more things done.

- IOT Opportunities: Since Python forms the basis of new platforms like Raspberry Pi, it finds the future bright for the Internet Of Things. This is a way to connect the language with the real world.

- Simple and Easy: When working with Java, you may have to create a class to print 'Hello World'. But in Python, just a print statement will do. It is also quite easy to learn**,** understand, and code**.** This is why when people pick up Python; they have a hard time adjusting to other more verbose languages like Java.

- Readable: Because it is not such a verbose language, reading Python is much like reading English. This is the reason why it is so easy to learn, understand, and code. It also does not need curly braces to define blocks, and indentation is mandatory.

7.3 DISADVANTAGES OF PYTHON

- Speed Limitations: We have seen that Python code is executed line by line. But since Python is interpreted, it often results in slow execution. This, however, isn't a problem unless speed is a focal point for the project. In other words, unless high speed is a requirement, the benefits offered by Python are enough to distract us from its speed limitations.

- Weak in Mobile Computing and Browsers: While it serves as an excellent server-side language, Python is much rarely seen on the client-side. Besides that, it is rarely ever used to implement smartphone-based applications. One such application is called Carbonnelle. The reason it is not so famous despite the existence of Brython is that it isn't that secure.

- Design Restrictions: As you know, Python is dynamically-typed. This means that you don't need to declare the type of variable while writing the code. It uses duck-typing. But wait, what's that? Well, it just means that if it looks like a duck, it must be a duck. While this is easy on the programmers during coding, it can raise run-time errors.

- Underdeveloped Database Access Layers: Compared to more widely used technologies like JDBC (Java DataBase Connectivity) and ODBC (Open DataBase Connectivity), Python's database access layers are a bit underdeveloped. Consequently, it is less often applied in huge enterprises.

- Simple: No, we're not kidding. Python's simplicity can indeed be a problem. Take my example. I don't do Java; I'm more of a Python person. To me, its syntax is so simple that the verbosity of Java code seems unnecessary.

7.4 WHAT IS MACHINE LEARNING?

Before we take a look at the details of various machine learning methods, let's start by looking at what machine learning is, and what it isn't. Machine learning is often categorized as a subfield of artificial intelligence, but I find that categorization can often be misleading at first brush. The study of machine learning certainly arose from research in this context, but in the data science application of machine learning methods, it's more helpful to think of machine learning as a means of building models of data.

7.5 CATEGORIES OF MACHINE LEARNING

Supervised learning involves somehow modeling the relationship between measured features of data and some label associated with the data; once this model is determined, it can be used to apply labels to new, unknown data. This is further subdivided into classification tasks and regression tasks: in classification, the labels are discrete categories, while in regression, the labels are continuous quantities. We will see examples of both types of supervised learning in the following section.

Unsupervised learning involves modeling the features of a dataset without reference to any label, and is often described as "letting the dataset speak for itself." These models include tasks such as clustering and dimensionality reduction. Clustering algorithms identify distinct groups of data, while dimensionality reduction algorithms search for more succinct representations of the data. We will see examples of both types of unsupervised learning in the following section.

7.6 APPLICATIONS OF MACHINE LEARNING

- Emotion analysis
- Sentiment analysis
- Error detection and prevention
- Weather forecasting and prediction
- Stock market analysis and forecasting
- Speech synthesis

7.7 NEED OF MACHINE LEARNING

Human beings, at this moment, are the most intelligent and advanced species on earth because they can think, evaluate and solve complex problems. On the other side, AI is still in its initial stage and haven't surpassed human intelligence in many aspects. Then the question is that what is the need to make machine learn? The most suitable reason for doing this is, "to make decisions, based on data, with efficiency and scale".

Lately, organizations are investing heavily in newer technologies like Artificial Intelligence, Machine Learning and Deep Learning to get the key information from data to perform several real-world tasks and solve problems. We can call it data-driven decisions taken by machines, particularly to automate the process. These data-driven decisions can be used, instead of using programming logic, in the problems that cannot be programmed inherently. The fact is that we can't do without human intelligence, but other aspect is that we all need to solve real-world problems with efficiency at a huge scale. That is why the need for machine learning arises.

7.8 CHALLENGES IN MACHINE LEARNING

- Quality of data − Having good-quality data for ML algorithms is one of the biggest challenges. Use of low-quality data leads to the problems related to data preprocessing and feature extraction.
- Time-Consuming task − Another challenge faced by ML models is the consumption of time especially for data acquisition, feature extraction and retrieval.
- Lack of specialist persons − As ML technology is still in its infancy stage, availability of expert resources is a tough job.
- No clear objective for formulating business problems − Having no clear objective and well-defined goal for business problems is another key challenge for ML because this technology is not that mature yet.
- Issue of overfitting & under fitting − If the model is overfitting or underfitting, it cannot be represented well for the problem.

7.9 HOW TO START MACHINE LEARNING?

Arthur Samuel coined the term **"**Machine Learning" in 1959 and defined it as a "Field of study that gives computers the capability to learn without being explicitly programmed".

This is a rough roadmap you can follow on your way to becoming an insanely talented Machine Learning Engineer.

Step 1 – Understand the Prerequisites

In case you are a genius, you could start ML directly but normally, there are some prerequisites that you need to know which include Linear Algebra, Multivariate Calculus, Statistics, and Python.

- **Learn Linear Algebra and Multivariate Calculus:**

Both Linear Algebra and Multivariate Calculus are important in Machine Learning.. If you are more focused on application heavy machine learning, then you will not be that heavily focused on maths as there are many common libraries available. But if you want to focus on R&D in Machine Learning, then mastery of Linear Algebra and Multivariate Calculus is very important as you will have to implement many ML algorithms from scratch.

- **Learn Statistics:**

Data plays a huge role in Machine Learning. In fact, around 80% of your time as an ML expert will be spent collecting and cleaning data. And statistics is a field that handles the collection, analysis, and presentation of data.
Some of the key concepts in statistics that are important are Statistical Significance, Probability Distributions, Hypothesis Testing, Regression, etc.

- **Learn Python:**

Some people prefer to skip Linear Algebra, Multivariate Calculus and Statistics and learn them as they go along with trial and error. But the one thing that you absolutely cannot skip is Python! While there are other languages you can use for Machine Learning like R, Scala, etc. Python is currently the most popular language for ML.

Step 2 – Learn Various ML Concepts

Now that you are done with the prerequisites, you can move on to actually learning ML (Which is the fun part!!!) It's best to start with the basics and then move on to the more complicated stuff. Some of the basic concepts in ML are:

Terminologies of Machine Learning:

- **Model –** A model is a specific representation learned from data by applying some machine learning algorithm. A model is also called a hypothesis.
- **Feature –** A feature is an individual measurable property of the data. A set of numeric features can be conveniently described by a feature vector. Feature vectors are fed as input to the model. For example in order to predict a fruit there may be features color, smell, taste, etc.
- **Target (Label) –** A target variable or label is the value to be predicted by our model. For the fruit example discussed in the feature section, the label with each set of input would be the name of the fruit like apple, orange, banana, etc.
- **Training –** The idea is to give a set of inputs(features) and it's expected outputs(labels), so after training, we will have a model (hypothesis) that will then map new data to one of the categories trained on.

7.10 ADVANTAGES OF MACHINE LEARNING

- Easily identifies trends and patterns: Machine Learning can review large volumes of data and discover specific trends and patterns that would not be apparent.
- No human intervention needed (automation): With ML, you don't need to babysit your project every step of the way. Since it means giving machines the ability to learn, it lets them make predictions and also improve the algorithms on their own.
- Continuous Improvement: As ML algorithms gain experience, they keep improving in accuracy and efficiency. This lets them make better decisions. Say you need to make a weather forecast model.
- Handling multi-dimensional and multi-variety data: Machine Learning algorithms are good at handling data that are multi-dimensional and multi-variety, and they can do this in dynamic or uncertain environments.
- Wide Applications: You could be an e-tailer or a healthcare provider and make ML work for you. Where it does apply, it holds the capability to help deliver a much more personal experience to customers while also targeting the right customers.

7.11 DISADVANTAGES OF MACHINE LEARNING

- Data Acquisition*:* Machine Learning requires massive data sets to train on, and these should be inclusive/unbiased, and of good quality. There can also be times where they must wait for new data to be generated.
- Time and Resources*:* ML needs enough time to let the algorithms learn and develop enough to fulfill their purpose with a considerable amount of accuracy and relevancy. It also needs massive resources to function.

- Interpretation of Results*:* Another major challenge is the ability to accurately interpret results generated by the algorithms. You must also carefully choose the algorithms for your purpose.

- High error-susceptibility: Machine Learning is autonomous but highly susceptible to errors. Suppose you train an algorithm with data sets small enough to not be inclusive. You end up with biased predictions coming from a biased training set.

7.12 MODULES USED IN PROJECT

- Tensorflow: Tensorflow is a free and open-source software library for dataflow and differentiable programming across a range of tasks. It is a symbolic math library, and is also used for machine learning applications such as neural networks.
- Numpy: Numpy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays.
    - A powerful N-dimensional array object
    - Sophisticated (broadcasting) functions
    - Tools for integrating C/C++ and Fortran code
- Pandas: Pandas is an open-source Python Library providing high-performance data manipulation and analysis tool using its powerful data structures. Python was majorly used for data mugging and preparation.

- Matplotlib: Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms.

- Scikit-learn: Scikit-learn provides a range of supervised and unsupervised learning algorithms via a consistent interface in Python. It is licensed under a permissive simplified BSD license and is distributed under many Linux distributions.

8. SAMPLE CODE

CHAPTER 8

```
import os
from sklearn.ensemble import RandomForestClassifier
from flask import Flask, render_template, request, redirect, Response
from sklearn.preprocessing import normalize
from sklearn.metrics import accuracy_score
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
from xgboost import XGBClassifier
import lightgbm as lgb
import pandas as pd
import numpy as np
app = Flask(__name__)
app.secret_key = 'dropboxapp1234'
global classifier
@app.route("/TrainML")
Def TrainML():
global classifier
    dataset = pd.read_csv("model_data.csv")
    dataset.fillna(0, inplace = True)
    dataset = dataset.values
    X = dataset[:,0:dataset.shape[1]-1]
    Y = dataset[:,dataset.shape[1]-1]
    indices = np.arange(X.shape[0])
    color = '<font size="" color="black">'
 rfc = RandomForestClassifier(n_estimators=200) #training random forest
```

```
    rfc.fit(X, Y)
    predict = rfc.predict(X_test)
    accuracy = accuracy_score(y_test,predict)
    classifier = rfc
    output = '<table border="1" align="center">'
    output+='<tr><th>Algorithm Name</th><th>Dataset Size</th><th>Training
    Dataset Size</th><th>Testing Dataset Size</th><th>Accuracy</th></tr>'
output+='<tr><td>'+color+'RandomForestAlgorithm</td><td>'+color+str(X.shape[0])+'</td><td>'
+color+str(X_train.shape[0])+'</td>'
  output+='<td>'+color+str(X_test.shape[0])+'</td><td>'+color+str(accuracy)+'</td></tr>'
    rfc = XGBClassifier() #training XGBoost
    rfc.fit(X_train, y_train)
    for i in range(0,10):
    y_test1[i] = 0
    predict = rfc.predict(X_test)
    accuracy = accuracy_score(y_test1,predict)
output+='<tr><td>'+color+'XGBoostAlgorithm</td><td>'+color+str(X.shape[0])+'</td><td>'+colo
r+str(X_train.shape[0])+'</td>'
output+='<td>'+color+str(X_test.shape[0])+'</td><td>'+color+str(accuracy)+'</td></tr>'
    rfc = KNeighborsClassifier(n_neighbors = 2)#training KNN
    rfc.fit(X_train, y_train)
    predict = rfc.predict(X_test)
    accuracy = accuracy_score(y_test,predict)
output+='<tr><td>'+color+'KNNAlgorithm</td><td>'+color+str(X.shape[0])+'</td><td>'+color+st
r(X_train.shape[0])+'</td>'
output+='<td>'+color+str(X_test.shape[0])+'</td><td>'+color+str(accuracy)+'</td></tr>'
```

```
    rfc = lgb.LGBMClassifier() #training LightGBM
    rfc.fit(X_train, y_train)
    predict = rfc.predict(X_test)
    for i in range(0,15):
    y_test2[i] = 0
    accuracy = accuracy_score(y_test2,predict)
output+='<tr><td>'+color+'LightGBMAgorithm</td><td>'+color+str(X.shape[0])+'</td><td>'+col
or+str(X_train.shape[0])+'</td>'
output+='<td>'+color+str(X_test.shape[0])+'</td><td>'+color+str(accuracy)+'</td></tr>'
output+='</table><br/><br/><br/><br/>'
 return render_template("ViewAccuracy.html",error=output)
@app.route('/PredictAction', methods =['GET', 'POST'])
def PredictAction():
 if request.method == 'POST':
        age = int(request.form['t1'])
        gender = request.form['t2']
        if gender == 'Male':
        gender = 1
        else:
        gender = 0
        cp = int(request.form['t3'])
        trestbps = int(request.form['t4'])
        cholestral = int(request.form['t5'])
        fbs = int(request.form['t6'])
        restecg = int(request.form['t7'])
        thalach = int(request.form['t8'])
```

```
        exang = int (request.form ['t9'])
        oldpeak = float (request.form['t10'])
        slope = int(request.form ['t11'])
        ca = int (request.form ['t12'])
        thal = int (request.form ['t13'])
        arr = [age,gender,cp,trestbps,cholestral,fbs,restecg,thalach,exang,oldpeak,slope,ca,thal]
        temp = []
        temp.append(arr)
        temp = np.asarray(temp)
        predictValue = classifier.predict(temp)
        print(predictValue)
output = "none"
prob = (cholestral / 400) * 100
if predictValue == 0:
output = "Your values are NORMAL"
if predictValue == 1:
if cholestral < 200
output = "Your values are ABNORMAL but risk of heart attack is LOW with prob "+str(prob)+'%'
if cholestral >= 200 and cholestral < 250:
output = "Your values are ABNORMAL but risk of heart attack is MEDIUM with prob "+str(prob)+'%'
if cholestral >= 250:
 output = "Your values are ABNORMAL but risk of heart attack is HIGH with prob "+str(prob)+'%'
return render_template("Predict.html",error=output)
 @app.route("/Predict")
```

```
def Predict():
return render_template("Predict.html")
@app.route("/index")
def index():
return render_template("index.html")
@app.route("/Login")
def Login():
 return render_template("Login.html")
@app.route('/UserLogin', methods =['GET', 'POST'])
def UserLogin():
 if request.method == 'POST':
 username = request.form['t1']
 password = request.form ['t2']
 if username == 'admin' and password == 'admin':
 return render_template("AdminScreen.html",error='Welcome '+username)
 else:
 return render_template("Login.html",error='Invalid Login')
if __name__ == '__main__':
    app.run(host='127.0.0.1', port=8080, debug=True)
```

8.1 SYSTEM TEST

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software suits it.

8.2 TYPES OF TESTS

- Unit testing: Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration.
- Integration testing: Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent.
- Functional test: Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals. Functional testing is centered on the following items:

  Valid Input : identified classes of valid input must be accepted.

  Invalid Input : identified classes of invalid input must be rejected.

  Functions : identified functions must be exercised.

  Output : identified classes of application outputs must be exercised.

  Systems/Procedures : interfacing systems or procedures must be invoked.

- System Test: System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

- White Box Testing: White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

- Black Box Testing: Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

9. OUTPUT SCREENS

CHAPTER 9

1. To run project double click on ‘run.bat’ file to start FLASK screen.

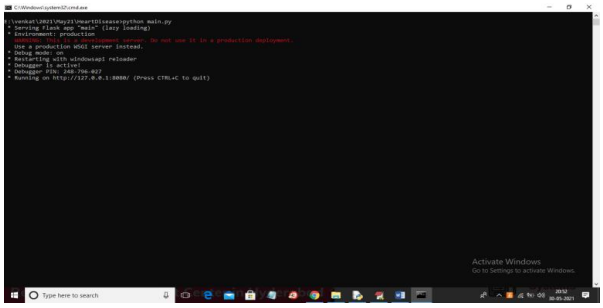

Fig 9.1 Run.bat File

2. In above screen FLASK server started and now open browser and enter URL as ‘http://127.0.0.1:8080/index’ and press enter key to get below home page.

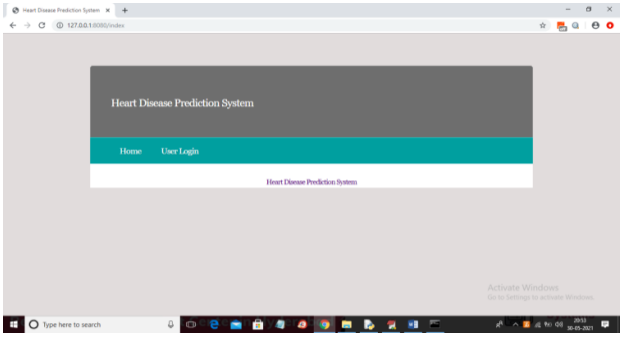

Fig 9.2 URL

3. In above screen click on ‘User Login’ link to get below screen.

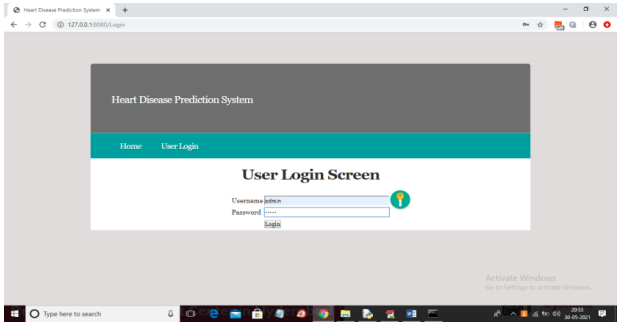

Fig 9.3 User Login

4. In above screen enter Username as ‘admin’ and password as ‘admin’ and then click on ‘Login’ button.

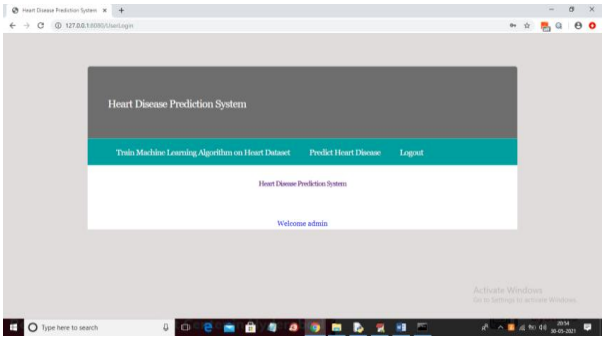

Fig 9.4 Heart Disease Prediction System

5. In above screen click on 'Train Machine Learning Algorithm on Heart Dataset' link to train all machine learning algorithms.

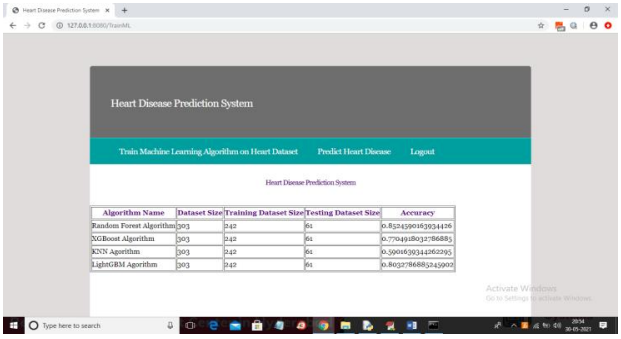

Fig 9.5 Train Machine Learning Algorithm

6. In above screen all algorithms are trained with heart dataset and we can see dataset size used to train and test all algorithms accuracy. Now ML models are ready and now click on 'Predict Heart Disease' link to get below screen.

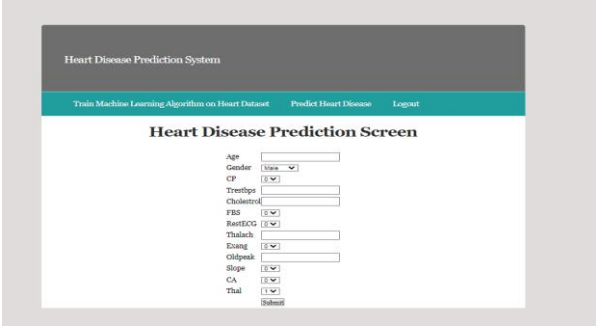

Fig 9.6 Enter Details

7. In above screen user will enter details about his/her heart and application will predict disease.

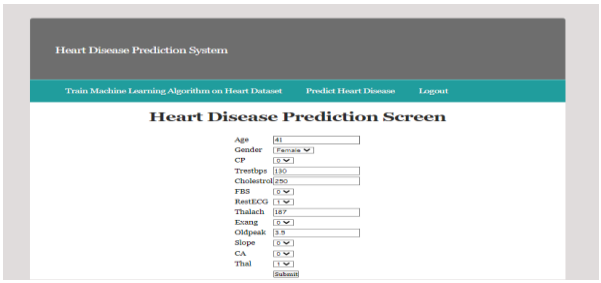

Fig 9.7 Prediction Screen-1

8. In above screen after entering details will get below screen of prediction.

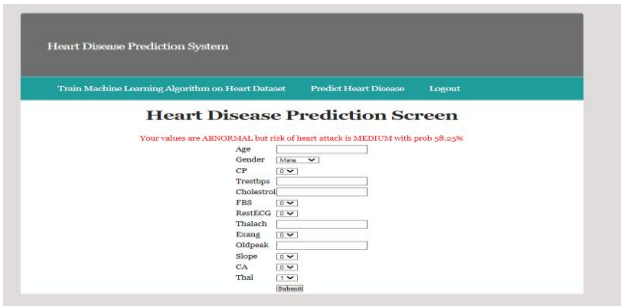

Fig 9.8 Heart Disease Prediction-1

9. The prediction is done with the probability rate of heart disease.

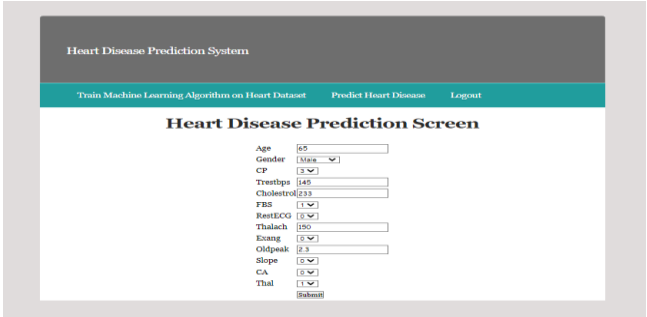

Fig 9.9 Prediction Screen-2

10. Now above screen with different data.

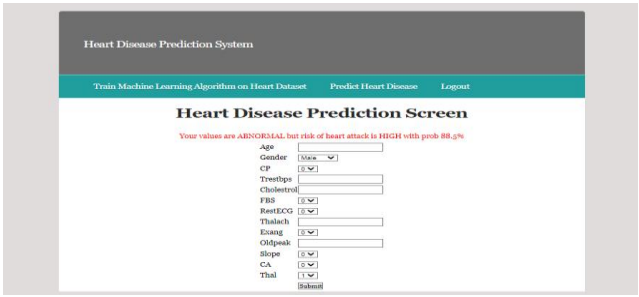

Fig 9.10 Heart Disease Prediction-2

11. Prediction is done with different datasets.

10. CONCLUSION

Heart Disease are complicated and taken away lots of lives every year. When the early symptoms of heart diseases are ignored, the patients might end up with drastic consequences in a short period of time. Selection of suitable techniques along with proper classification algorithms will lead to the development of prediction systems that gives enhanced accuracy. The paper presented the various machine learning algorithms for predicting the Heart disease factors such as blood pressure, cholesterol, age etc. Experiments were conducted on Indian government dataset and it has been established that Random Forest Regressor gives the highest yield accuracy.

FUTURE SCOPE

- Machine Learning is a crucial perspective for acquiring real-world and operative solution for Heart Disease Prediction.
- The future scope of this system aims at giving more sophisticated prediction models, risk calculations tools and feature extraction for other clinical risks.
- This can lead to selection of proper treatment methods for a patient diagnosed with heart disease.

11. REFERENCES

CHAPTER 11

1. P. Groves, B. Kayyali, D. Knott, and S. van Kuiken, The ‘Big Data'Revolution’ in Healthcare: Accelerating Value and Innovation. USA: Center for US Health System Reform Business Technology Office, 2016.
2. M. Chen, S. Mao, and Y. Liu, ``Big data: A survey," Mobile Netw. Appl., vol. 19, no. 2, pp. 171209, Apr. 2014.
3. P. B. Jensen, L. J. Jensen, and S. Brunak, ``Mining electronic health records: Towards better research applications and clinical care," Nature Rev. Genet., vol. 13, no. 6, pp. 395405, 2012.
4. D. Tian, J. Zhou, Y. Wang, Y. Lu, H. Xia, and Z. Yi, ``A dynamic and self-adaptive network selection method for multimode communications in heterogeneous vehicular telematics," IEEE Trans. Intell. Transp. Syst., vol. 16, no. 6, pp. 30333049, Dec. 2015.
5. M. Chen, Y. Ma, Y. Li, D. Wu, Y. Zhang, and C. Youn, ``Wearable 2.0: Enable human-cloud integration in next generation healthcare system," IEEE Commun., vol. 55, no. 1, pp. 5461, Jan. 2017.
6. M. Chen, Y. Ma, J. Song, C. Lai, and B. Hu, ``Smart clothing: Con- necting human with clouds and big data for sustainable health monitoring," ACM/Springer Mobile Netw. Appl., vol. 21, no. 5, pp. 825845, 2016.
7. M. Chen, P. Zhou, and G. Fortino, ``Emotion communication system," IEEE Access, vol. 5, pp. 326337, 2017, doi: 10.1109/ACCESS.2016.2641480.
8. M. Qiu and E. H.-M. Sha, ``Cost minimization while satisfying hard/soft timing constraints for heterogeneous embedded systems," ACM Trans. Design Autom. Electron. Syst., vol. 14, no. 2, p. 25, 2009.
9. J. Wang, M. Qiu, and B. Guo, ``Enabling real-time information service on telehealth system over cloud-based big data platform," J. Syst. Archit., vol. 72, pp. 6979, Jan. 2017.