Heart Disease Prediction

using

Machine learning with Supervised Learning

**A PROJECT REPORT**

###### ***Submitted by***

Group - 226

##### Ansh Bansal (22BCE10330)

**Aryan Sharma** (22BCE11100)

**Shivam Pant** (22BCE10068)

**Prakhar Mishra** (22BCE10639)

**Ayush Vishnoi** (22BCE10577)

*in partial fulfillment for the award of the degree*

*of*

**BACHELOR OF TECHNOLOGY**

*in*

# COMPUTER SCIENCE AND ENGINEERING



**SCHOOL OF COMPUTING SCIENCE AND ENGINEERING**

**VIT BHOPAL UNIVERSITY**

**KOTHRIKALAN, SEHORE**

**MADHYA PRADESH – 466114**

##### May 2024

**VIT BHOPAL UNIVERSITY**

**KOTHRI KALAN, SEHORE**

**MADHYA PRADESH – 466114**

**BONAFIDE CERTIFICATE**

Certified that this project report titled **“Heart Disease Prediction using Machine Learning with Supervised Learning”** is the bonafide work of “**Ansh Bansal (22BCE10330), Aryan Sharma (22BCE11100), Ayush Vishnoi (22BCE10577), Prakhar Mishra (22BCE10639), Shivam Pant (22BCE10068)”** who carried out the project work under the supervision of **Dr.** **Ajeet Singh**. Certified further that to the best of my knowledge the work reported here does not form part of any other project on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

**PROJECT SUPERVISOR**

Dr. Ajeet Singh

School of Computer Science and Engineering

VIT BHOPAL UNIVERSITY

The Project Exhibition II is held on April 2024.

**ACKNOWLEDGEMENT**

First and foremost, we would like to thank the Lord Almighty for His presence and immense blessings throughout the project work.

We wish to express my heartfelt gratitude to **Dr. Poonkuntran S**, Head of Department, **Dr. Sandip Mal**, Programme chair, **Dr. Preetam Suman**, Programme chair, School of Computer Science and Engineering for much of his valuable support and encouragement in carrying out this work.

We would like to thank our internal guide **Dr. Ajeet Singh**, for continually guiding and actively participating in our project, giving valuable suggestions to complete the project work.

We would like to thank all the technical and teaching staff of the School of Computer Science and Engineering, who extended directly or indirectly all support.

Last, but not least, we are deeply indebted to our parents who have been the greatest support while we worked day and night for the project to make it a success.

**LIST OF ABBREVIATIONS**

|  |  |  |
| --- | --- | --- |
| **SR. NO.** | **ABBREVIATIONS** | **FULL-FORM** |
| 1 | BSD | Berkeley Source Distribution |
| 2 | Chol | Cholesterol |
| 3 | Cp | Chest Pain |
| 4 | DFD | Data Flow Diagram |
| 5 | FBS | Fasting Blood Sugar |
| 6 | HTML | Hypertext Markup Language |
| 7 | LogR | Logistic Regression |
| 8 | MIME | Multipurpose Internet Mail Extension |
| 9 | MOE | Measures of Effectiveness |
| 10 | MOS | Measures of Suitability |
| 11 | NumPy | Numerical Python |
| 12 | PLR | Penalized Logistic Regression |
| 13 | RBF | Radial Basis Functions |
| 14 | Restecg | Resting Electrocardiographic results |
| 15 | SciPy | Scientific Python |
| 16 | SRS | Software Requirements Specifications |
| 17 | SVC | Support Vector Classifier |
| 18 | SVM | Support Vector Machine |
| 19 | UI | User Interface |
| 20 | UML | Unified Modeling Language |

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **FIGURE NO.** | **TITLE** | **PAGE NO.** |
| **1** | Fig 3.1: System Architecture | 14 |
| **2** | Fig 3.2: Use case Diagram | 16 |
| **3** | Fig 3.3: Sequence Diagram to illustrate the dynamic view of a system | 17 |
| **4** | Fig 3.4: Class Diagram | 18 |
| **5** | Fig 5.1: logistic regression | 26 |
| **6** | Fig 5.2: Random Forest | 27 |
| **7** | Fig 5.3: Support Vector Machine (SVM) | 28 |
| **8** | Fig 5.4: Decision Tree Classifier | 29 |
| **9** | Fig 6.1: Dataset Analysis | 31 |
| **10** | Fig 6.2: Heart Disease Frequency for Ages | 31 |
| **11** | Fig 6.3: Training and Testing Data Accuracy using SVM | 32 |
| **12** | Fig 6.4: Training and Testing Data Accuracy using Logistic Regression | 32 |
| **13** | Fig 6.5: Training and Testing Data Accuracy using Decision Tree Classifier | 33 |
| **14** | Fig 6.6: Training and Testing Data Accuracy using Random Forest Classifier | 33 |
| **15** | Fig 6.7: User Interface | 34 |
| **16** | Fig 6.8: Results for heart disease of a particular patient | 34 |

**LIST OF TABLES**

|  |  |  |
| --- | --- | --- |
| **TABLE NO.** | **TITLE** | **PAGE NO.** |
| 1 | Table 2.1: List of attributes | 14 |

**ABSTRACT**

In the medical field, the diagnosis of heart disease is the most difficult task. The diagnosis of heart disease is difficult as a decision relied on grouping of large clinical and pathological data. Due to this complication, the interest increased in a significant amount between the researchers and clinical professionals about the efficient and accurate heart disease prediction. In case of heart disease, the correct diagnosis in early stage is important as time is the very important factor.

Heart disease is the principal source of deaths widespread, and the prediction of heart disease is significant at an untimely phase. Machine learning in recent years has been the evolving, reliable and supporting tool in medical domain and has provided the greatest support for predicting disease with correct case of training and testing.

The main idea behind this work is to study diverse prediction models for heart disease and select important heart disease features using Random Forests algorithm, SVM, Decision Tree Classifier, Logistic Regression.

Random Forests is the Supervised Machine Learning algorithm which has the high accuracy compared to other Supervised Machine Learning algorithms such as logistic regression etc. By using Random Forests Algorithm, we are going to predict if a person has heart disease or not.

Decision Tree Classifier is a supervised learning algorithm that uses a tree-like structure to make decisions, offering simplicity and interpretability but can be prone to overfitting.

Support Vector Machine (SVM) is a supervised learning algorithm that finds the optimal hyperplane to separate classes, providing robust classification with different kernels but requiring careful parameter tuning.

Logistic Regression is a supervised learning algorithm used for binary classification, modeling the probability of a target class based on input features and offering simplicity and interpretability.

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **CHAPTER NO.** | **TITLE** | **PAGE NO.** |
|  | List of Abbreviations  List of Figures  List of Tables  Abstract | iv  v  vi  vii |
| 1 | **INTRODUCTION** IntroductionProblem definition 1.3 Motivation  1.4 Objective | 10  10  10  11 |
| 2 | **LITERATURE SURVEY** | 12-13 |
| 3 | **SYSTEM DESIGN**  3.1System Architecture  3.2 Modules  3.3 Data Flow Diagram  3.4 UML Diagram  3.4.1 Use-case Diagram  3.4.2 Sequence Diagram  3.4.3 Class Diagram | 14  14-15  16  16  16  17  17-18 |
| 4 | **SOFTWARE REQUIREMENT SPECIFICATION**  4.1 Requirement Analysis  4.1.1 Domain Requirement  4.1.2 Requirement Efficiency  4.1.3 Usability  4.1.4 Operational Requirements  4.2 System Requirements  4.2.1 Hardware Requirements  4.2.2 Software Requirements  4.3 Software Description  4.3.1 Python  4.3.2 Pandas  4.3.3 NumPy  4.3.4 Sckit Learn  4.3.5 Matplot Lib  4.3.6 Jupyter Notebook | 19  19  20  20  20  21  21  21  21  21  21  22  23  23  23 |
| 5 | **SYSTEM ANALYSIS**  5.1 Existing system  5.2 Proposed System  5.3 Algorithm  5.3.1 Logistic Regression  5.3.2 Random Forest  5.3.3 Support Vector Machine (SVM)  5.3.4 Decision Tree classifier  5.4 Feasibility Study  5.4.1 Economical Feasibility  5.4.2 Technical Feasibility  5.4.3 Operational Feasibility | 24  24  25  25-26  26-27  27-28  28-29  29  30  30  30 |
| 6 | **OBTAINED EXPERIMENTAL RESULT**  6.1 Dataset Analysis  6.2 Accuracy  6.3 User Experience | 31  32-33  33-34 |
| 7 | **FUTURE SCOPE AND CONCLUSION**  7.1 Future Scope  7.2 Conclusion | 35  35 |
|  | References | 36 |

**CHAPTER 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. Much research has been conducted to pinpoint the most influential factors of heart disease and 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 deciding lifestyle changes in high-risk patients and reduces 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.

**1.1 Problem Definition**

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

**1.2 Motivation**

Machine learning techniques have been around us and have 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 machine learning. With firsthand 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 done 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.

**1.3 Objectives**

The main objective of developing this project is:

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.
4. Implement the system to help prevent biases that may arise from human decision-making.

**CHAPTER 2**

**LITERATURE SURVEY**

Machine Learning techniques are used to analyze and predict medical data information resources. Diagnosis of heart disease is a significant and tedious task in medicine. The term heart disease encompasses the various diseases that affect the heart. Exposing heart disease from various factors or symptoms is an issue not complimentary from false presumptions often accompanied by unpredictable effects. The data classification is based on Supervised Machine Learning algorithm which results in better accuracy. Here we are using the Random Forest as the training algorithm to train the heart disease dataset and to predict heart disease. The results showed that the medicinal prescription and designed prediction system can prophesy the heart attack successfully. Machine Learning techniques are used to indicate early mortality by analyzing heart disease patients and their clinical records (Richards, G. et al., 2001). (Sung, S.F. et al., 2015) have brought about the two Machine Learning techniques, k-nearest neighbor model and existing multi linear regression to predict the stroke severity index (SSI) of the patients. Their study shows that k nearest neighbor performed better than Multi Linear Regression model. (Arslan, A. K. et al., 2016) have suggested various Machine Learning techniques such as Support Vector Machine (SVM), Penalized Logistic Regression (PLR) to predict heart stroke. Their results show that SVM produced the best performance in prediction when compared to other models. Boshra Brahmi et al, [20] developed different Machine Learning techniques to evaluate the prediction and diagnosis of heart disease. The main objective is to evaluate the different classification techniques such as J48, Decision Tree, KNN and Naïve Bayes. After this, evaluating some performance in measures of accuracy, precision, sensitivity, specificity are evaluated.

**Data source**

Clinical databases have collected a significant amount of information about patients and their medical conditions. Records set with medical attributes were obtained from the Cleveland Heart Disease database. With the help of the dataset, the patterns significant to the heart attack diagnosis are extracted. The records were split equally into two datasets: training dataset and testing dataset. 303 records with 76 medical attributes were obtained.

All the attributes are numeric-valued. We are working on a reduced set of attributes, i.e. only 14 attributes.

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 distinguish 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 following table shows the list of attributes on which we are working.

|  |  |  |
| --- | --- | --- |
| S no | Attribute Name | Description |
| 1 | Age | age in years |
| 2 | Sex | (1 = male; 0 = female) |
| 3 | Cp | Chest Pain |
| 4 | Trestbps | resting blood pressure (in mm Hg on admission to the hospital) |
| 5 | Chol | serum cholesterol in mg/dl |
| 6 | Fbs | (fasting blood sugar > 120 mg/dl) (1 = true; 0 = false) |
| 7 | Restecg | resting electrocardiographic results |
| 8 | Thalach | maximum heart rate achieved |
| 9 | Exang | exercise induced angina (1 = yes; 0 = no) |
| 10 | Oldpeak | ST depression induced by exercise relative to rest |
| 11 | Slope | the slope of the peak exercise ST segment |
| 12 | Ca | number of major vessels (0-3) colored by fluoroscopy |
| 13 | Thal | 3 = normal; 6 = fixed defect; 7 = reversible defect |
| 14 | Target | 1 or 0 |

Table 2.1: List of attributes

**CHAPTER 3**

**SYSTEM DESIGN**

**3.1 SYSTEM ARCHITECTURE**

The below figure shows the process flow diagram or proposed work.

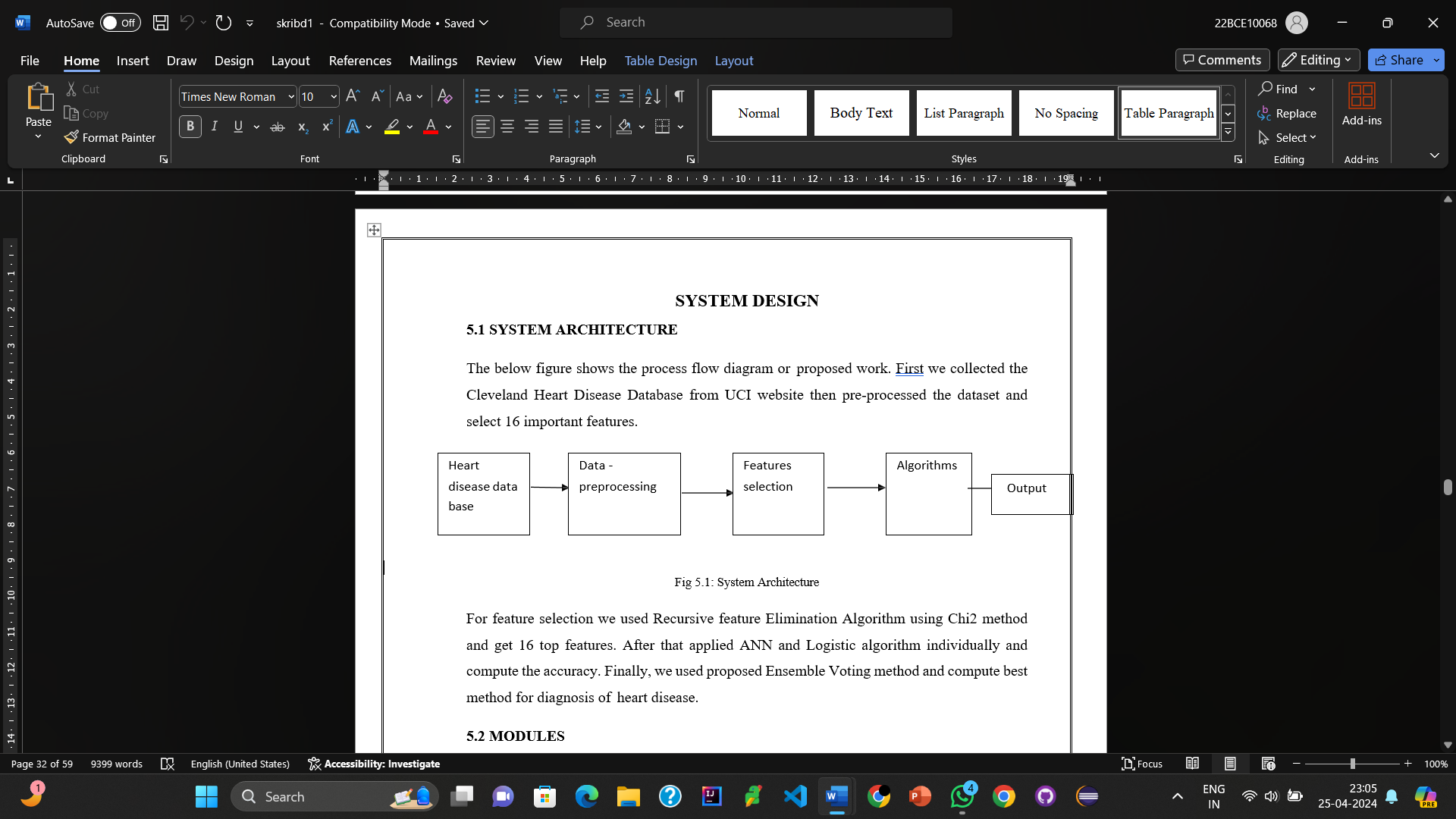


Fig 3.1: System Architecture

We import the Dataset from Kaggle like website and search for real time data to further train our model. We have to adapt our current system architecture with different machine learning models (Random Forest, Decision Tree, and Support Vector Machine), you can follow a similar structure and replace the Logistic Regression model with the desired model while keeping the data processing and evaluation steps largely unchanged.

**3.2 Modules**

1. **Data Collection and Processing**

Data Loading: Load the data from a CSV file into a Pandas Data Frame.

Data Exploration: Use. head(), .tail(), and .info() to understand the data and check for missing values.

Data Cleaning and Preparation: If necessary, fill in missing values or remove rows with missing data. Standardize or normalize numerical data if required.

1. **Splitting Features and Target**

Feature Selection: Define the features (X) by dropping the target variable (target column) from the DataFrame.

Target Variable: Define the target variable (Y) as the target column from the DataFrame.

1. **Data Splitting**

Train-Test Split: Split the data into training and testing sets using train\_test\_split with stratification on the target variable (test\_size=0.2 and stratify=Y).

1. **Model Selection and Training**

Model Initialization: Initialize the model of your choice (Random Forest, Decision Tree, or SVM).

Random Forest: from sklearn.ensemble import RandomForest

Decision Tree: from sklearn.tree import DecisionTreeClassifier

SVM: from sklearn.svm import SVC

Model Training: Fit the model to the training data (X\_train and Y\_train).

1. **Model Evaluation**

Training Accuracy: Use the trained model to predict the target variable for the training data and compute the accuracy score.

Test Accuracy: Use the trained model to predict the target variable for the test data and compute the accuracy score.

1. **Building a Predictive System**

Input Data Processing: Convert input data to a NumPy array and reshape it.

Prediction: Use the trained model to make a prediction based on the input data.

Output: Provide an output message based on the prediction.

Key Points for Different Models:

**Random Forest:** This ensemble learning method uses multiple decision trees to make predictions. The final prediction is based on the majority vote of the individual trees.

**Decision Tree:** This model works by creating a tree-like structure for decision-making. It splits the data based on features to predict the target variable.

**Support Vector Machine (SVM):** This algorithm aims to find the optimal separating hyperplane between different classes. It can work well for both classification and regression problems.

The overall architecture remains the same across these models, with the key difference being the choice of the machine learning algorithm used for training and prediction. Adjust the code to initialize the specific model and tune its hyperparameters if necessary.

**3.3 DATA FLOW DIAGRAM**

The data flow diagram (DFD) is one of the most important tools used by system analysis. Data flow diagrams are made up of number of symbols, which represents system components. Most data flow modeling methods use four kinds of symbols: Processes, Data stores, Data flows and external entities. These symbols are used to represent four kinds of system components. Circles in DFD represent processes. Data Flow represented by a thin line in the DFD and each data store has a unique name and square or rectangle represents external entities

**3.4 UML DIAGRAMS**

**3.4.1 Use-Case Diagram**

A use case diagram is a diagram that shows a set of use cases and actors and their relationships. A use case diagram is just a special kind of diagram and shares the same common properties as do all other diagrams, i.e. a name and graphical contents that are a projection into a model. What distinguishes a use case diagram from all other kinds of diagrams is its particular content.

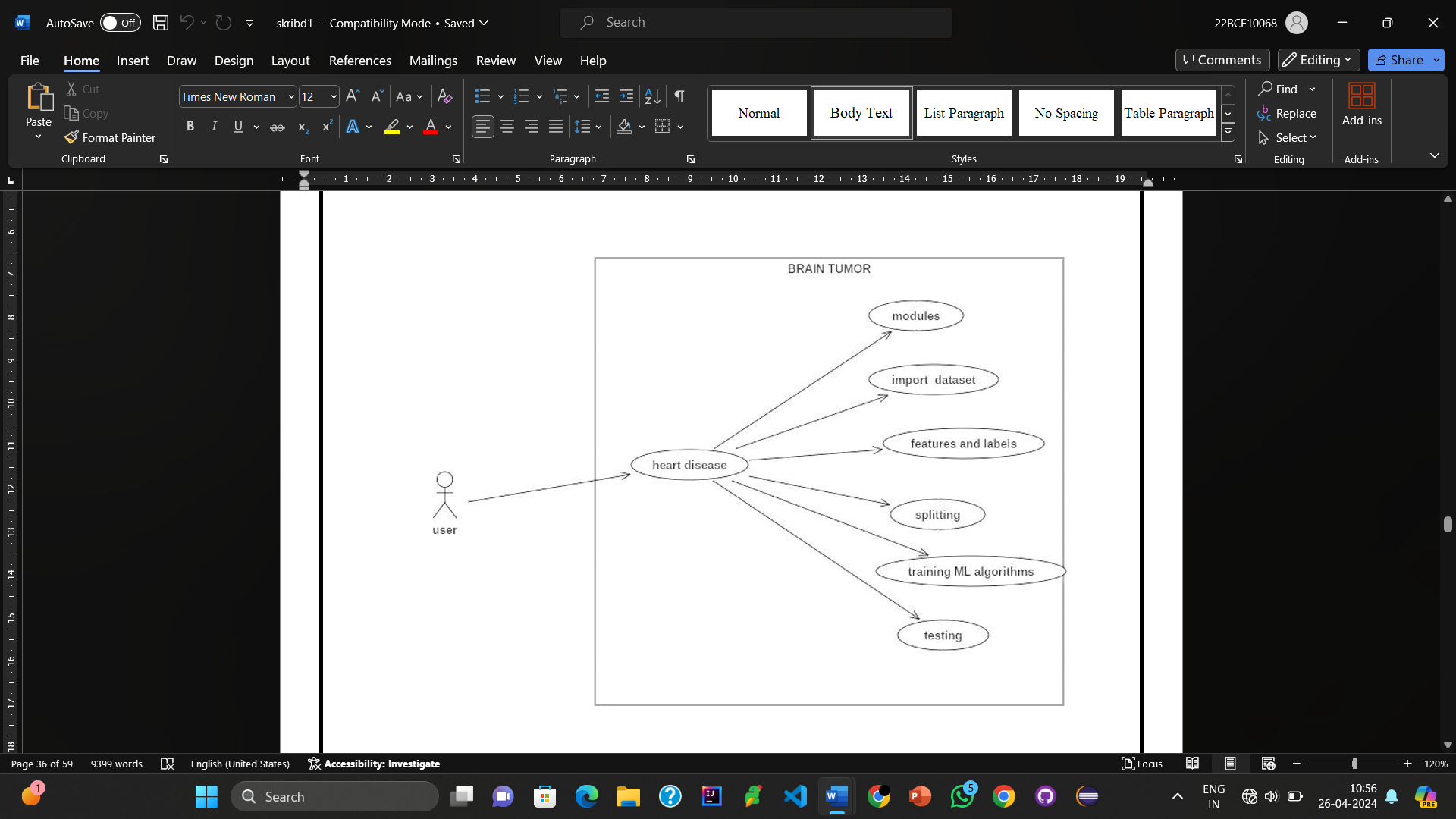


Fig 3.2: Use case Diagram

**3.4.2 Sequence Diagram**

A sequence diagram is an interaction diagram that emphasizes the time ordering of messages. A sequence diagram shows a set of objects and the messages sent and received by those objects. The objects are typically named or anonymous instances of classes, but may also represent instances of other things, such as collaborations, components, and nodes. We use sequence diagrams to illustrate the dynamic view of a system.

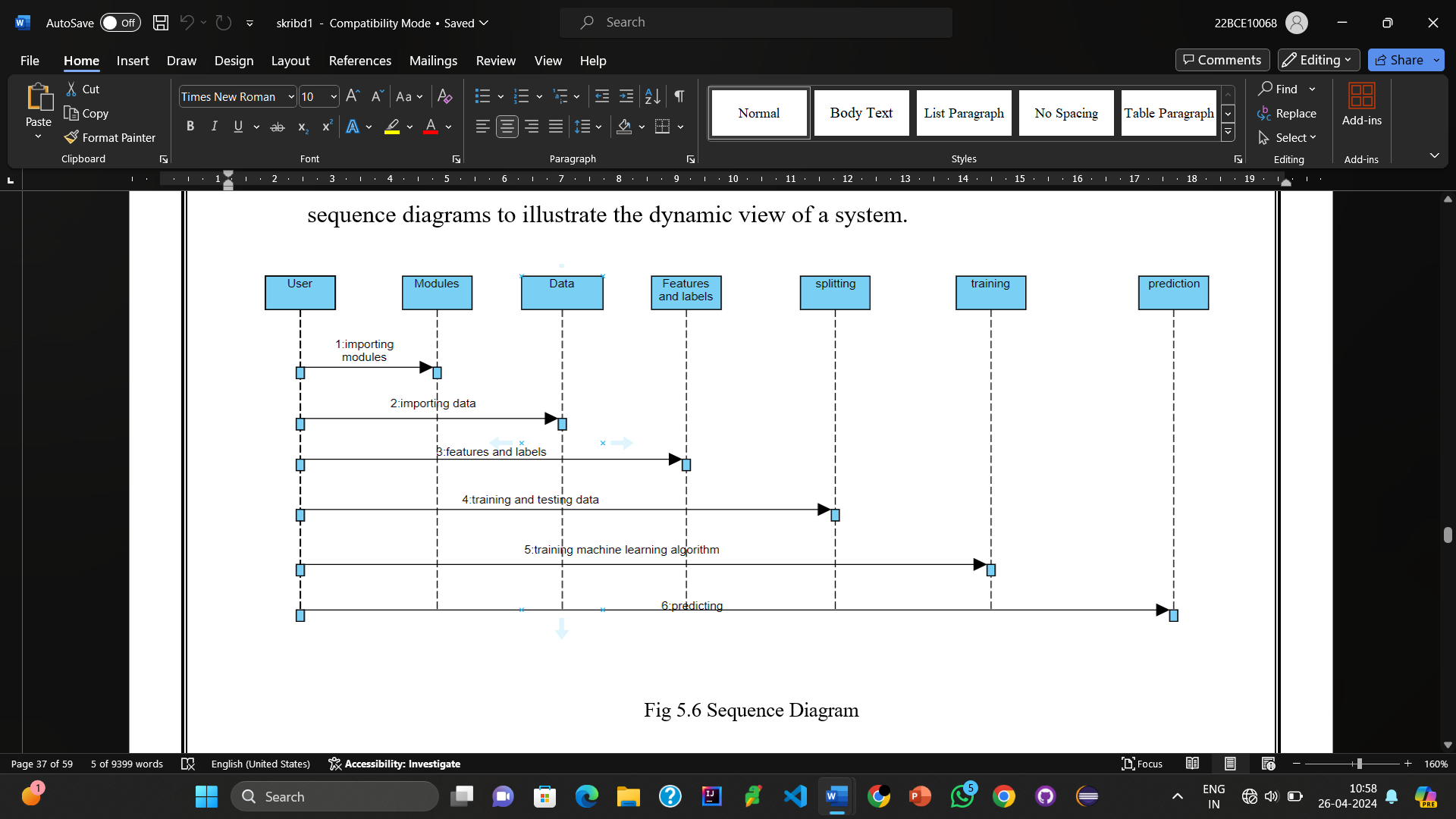


Fig 3.3: Sequence Diagram to illustrate the dynamic view of a system

**3.4.3 Class diagram**

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 objects. It provides a basic notation for other structure diagrams prescribed by UML. It is helpful for developers and other team members too.

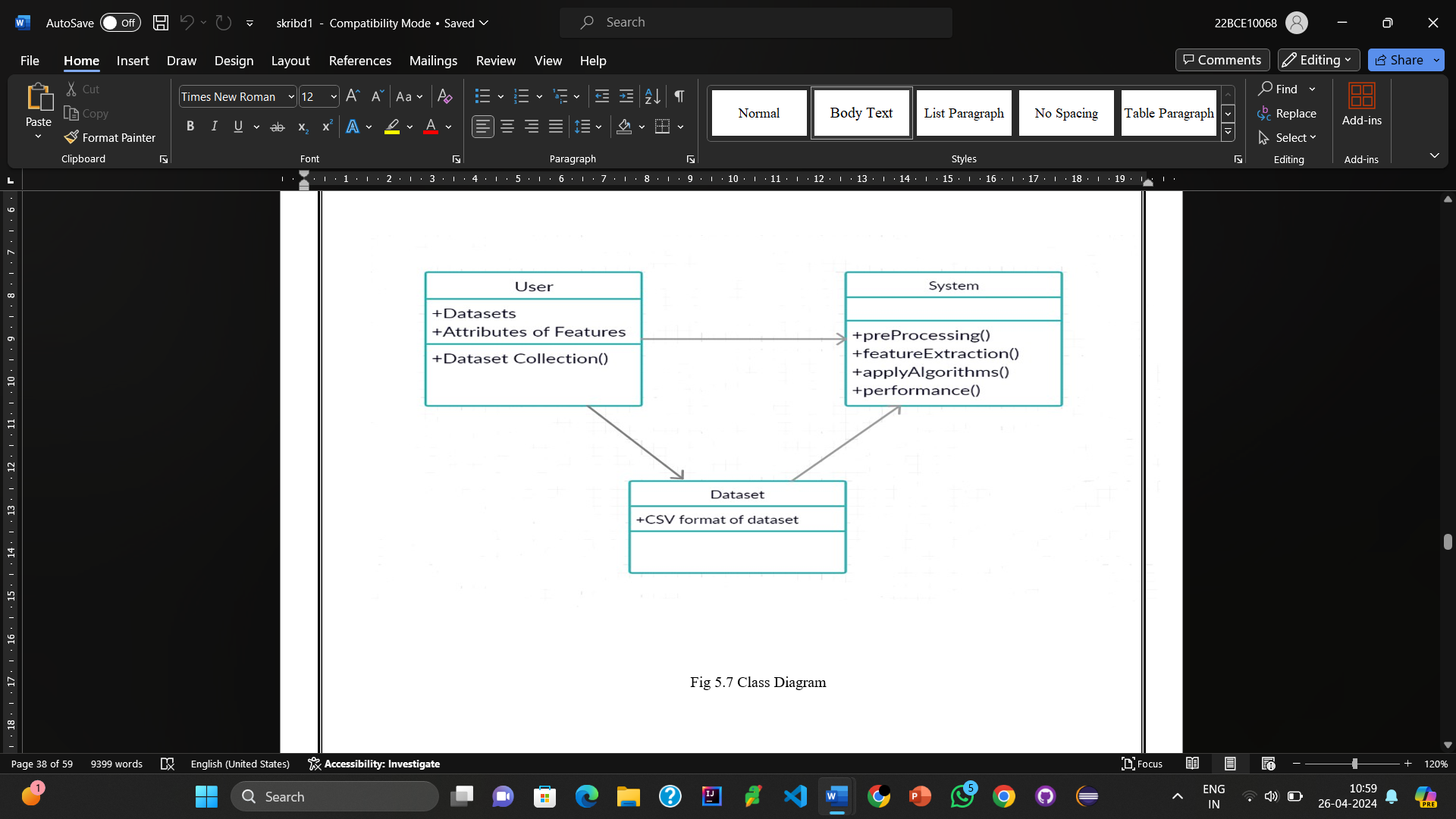


Fig 3.4: Class Diagram

**CHAPTER 4**

**SOFTWARE REQUIREMENT SPECIFICATIONS**

**4.1 REQUIREMENT ANALYSIS**

Software Requirement Specification (SRS) is the starting point of the software developing activity. As system grew more complex it became evident that the goal of the entire system cannot be easily comprehended. Hence the need for the requirement phase arose. The SRS is the means of translating the ideas of the minds of clients (the input) into a formal document. Under requirement specification, the focus is on specifying what has been found giving analysis such as representation, specification languages and tools, and checking the specifications are addressed during this activity. The Requirement phase terminates with the production of the validate SRS document. Producing the SRS document is the basic goal of this phase. The purpose of the Software Requirement Specification is to reduce the communication gap between the clients and the developers. Software Requirement Specification is the medium though which the client and user needs are accurately specified. It forms the basis of software development. A good SRS should satisfy all the parties involved in the system.

**4.1.1 Domain Requirements:**

This document is the only one that describes the requirements of the system. It is meant for the use by the developers, and will also be the bases for validating the final heart disease system. Any changes made to the requirements in the future will have to go through a formal change approval process. User Requirements User can decide on the prediction accuracy to decide on which algorithm can be used in real-time predictions.

**4.1.2 Requirements Efficiency:**

Less time for predicting the Heart Disease Reliability: Maturity, fault tolerance and recoverability. Portability: can the software easily be transferred to another environment, including install ability.

**4.1.3 Usability:**

How easy it is to understand, learn and operate the software system Organizational Requirements: Do not block the same available ports through the windows firewall. Internet connection should be available Implementation Requirements The dataset collection, internet connection to install related libraries. Engineering Standard Requirements User Interfaces User interface is developed in python, which gets input such stock symbol.

**4.1.4 Operational Requirements:**

1. Economic: The developed product is economic as it is not required any hardware interface etc. Environmental Statements of fact and assumptions that define the expectations of the system in terms of mission objectives, environment, constraints, and measures of effectiveness and suitability (MOE/MOS). The customers are those that perform the eight primary functions of systems engineering, with special emphasis on the operator as the key customer.
2. Health and Safety: The software may be safety-critical. If so, there are issues associated with its integrity level. The software may not be safety-critical although it forms part of a safety-critical system.

* For example, software may simply log transactions. If a system must be of a high integrity level and if the software is shown to be of that integrity level, then the hardware must be at least of the same integrity level.
* There is little point in producing 'perfect' code in some language if hardware and system software (in widest sense) are not reliable. If a computer system is to run software of a high integrity level, then that system should not at the same time accommodate software of a lower integrity level.
* Systems with different requirements for safety levels must be separated. Otherwise, the highest level of integrity required must be applied to all systems in the same environment.

**4.2 SYSTEM REQUIREMENTS**

**4.2.1 Hardware Requirements**

Processor: above 500 MHz

Ram: 4 GB

Hard Disk: 4 GB

Input device: Standard Keyboard and Mouse.

Output device: VGA and High-Resolution Monitor.

**4.2.2 Software Requirements**

Operating System: Windows 7 or higher

Programming: Python 3.6 and related libraries

Software: Jupyter Notebook.

**4.3 SOFTWARE DESCRIPTION**

**4.3.1 Python**

Python is an interpreted high-level programming language for general-purpose programming. Python has a design philosophy that emphasizes code readability, notably using significant whitespace. It provides constructs that enable clear programming on both small and large scales. Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library. Python interpreters are available for many operating systems.

**4.3.2 Pandas**

Pandas is an open-source Python Library providing high-performance data manipulation and analysis tool using its powerful data structures. The name Pandas is derived from the word Panel Data – an Econometrics from Multidimensional data. Prior to Pandas, Python was majorly used for data mining and preparation. It had very little contribution towards data analysis. Pandas solved this problem. Using Pandas, we can accomplish five typical steps in the processing and analysis of data, regardless of the origin of data — load, prepare, manipulate, model, and analyze. Python with Pandas is used in a wide range of fields including academic and commercial domains including finance, economics, Statistics, analytics, etc.Key Features of Pandas:

* Fast and efficient Data Frame object with default and customized indexing.
* Tools for loading data into in-memory data objects from different file formats.
* Data alignment and integrated handling of missing data.
* Reshaping and pivoting of date sets.
* Label-based slicing, indexing and sub setting of large data sets.
* Columns from a data structure can be deleted or inserted.
* Group by data for aggregation and transformations.
* High performance merging and joining of data.
* Time Series functionality.

**4.3.3 NumPy**

NumPy is a general-purpose array-processing package. It provides a high- performance multidimensional array object, and tools for working with these arrays. It is the fundamental package for scientific computing with Python. It contains various features including these important ones:

* A powerful N-dimensional array object
* Sophisticated (broadcasting) functions
* Tools for integrating C/C++ and Fortran code
* Useful linear algebra, Fourier transform, and random number capabilities 24
* Besides its obvious scientific uses, NumPy can also be used as an efficient multi-dimensional container of generic data. Arbitrary data-types can be defined using Numpy which allows NumPy to seamlessly and speedily integrate with a wide variety of databases.

**4.3.4 Sckit-Learn**

• Simple and efficient tools for data mining and data analysis\ • Accessible to everybody, and reusable in various contexts • Built on NumPy, SciPy, and matplotlib • Open source, commercially usable - BSD license

**4.3.5 Matplot lib**

• Matplotlib is a python library used to create 2D graphs and plots by using python scripts.• It has a module named pyplot which makes things easy for plotting by providing feature to control line styles, font properties, formatting axes etc. • It supports a very wide variety of graphs and plots namely - histogram, bar charts, power spectra, error charts etc.

**4.3.6 Jupyter Notebook**

• The Jupyter Notebook is an incredibly powerful tool for interactively developing and presenting data science projects. • A notebook integrates code and its output into a single document that combines visualizations, narrative text, mathematical equations, and other rich media. • The Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations and narrative text.• Uses include: data cleaning and transformation, numerical simulation, statistical modeling, data visualization, machine learning, and much more. • The Notebook has support for over 40 programming languages, including Python, R, Julia, and Scala. • Notebooks can be shared with others using email, Drop box, Git Hub and the Jupyter Notebook. • Your code can produce rich, interactive output: HTML, images, videos, LATEX, and custom MIME types. • Leverage big data tools, such as Apache Spark. Explore that same data with pandas, scikit-learn, ggplot2, Tensor Flow.

**CHAPTER 5**

**SYSTEM ANALYSIS**

### **5.1 EXISTING SYSTEM**

Clinical decisions are often made based on doctors’ 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 affect the quality of service provided to patients. There are many ways that a medical misdiagnosis can present itself. Whether a doctor is at fault, or hospital staff, a misdiagnosis of a serious illness can have very extreme and harmful effects. The National Patient Safety Foundation cites that 42% of medical patients feel they have had experienced a medical error or missed diagnosis. Patient safety is sometimes negligently given the back seat for other concerns, such as the cost of medical tests, drugs, and operations. Medical Misdiagnoses are a serious risk to our healthcare profession. If they continue, then people will fear going to the hospital for treatment. We can put an end to medical misdiagnosis by informing the public and filing claims and suits against the medical practitioners at fault.

#### **Disadvantages:**

* Prediction is not possible at early stages.
* In the Existing system, practical use of collected data is time consuming.
* Any faults occurred by the doctor or hospital staff in predicting would lead to fatal incidents.
* A highly expensive and laborious process needs to be performed before treating the patient to find out if he/she has any chance of getting heart disease in future.

### **5.2 PROPOSED SYSTEM**

This section depicts the overview of the proposed system and illustrates all the components, techniques and tools used for developing the entire system. To develop an intelligent and user-friendly heart disease prediction system, an efficient software tool is needed to train huge datasets and compare multiple machine learning algorithms. After choosing the robust algorithm with best accuracy and performance measures, it will be implemented on the development of the smart phone-based application for detecting and predicting heart disease risk level. Hardware components like Arduino/Raspberry Pi, different biomedical sensors, display monitor, buzzer etc. are needed to build the continuous patient monitoring system.

**5.3 ALGORITHMS**

#### **5.3.1 Logistic Regression**

A popular statistical technique to predict binomial outcomes (y = 0 or 1) is Logistic Regression. Logistic regression predicts categorical outcomes (binomial / multinomial values of y). The predictions of Logistic Regression (henceforth, LogR in this article) are in the form of probabilities of an event occurring, i.e. the probability of y=1, given certain values of input variables x. Thus, the results of LogR range between 0-1.

LogR models the data points using the standard logistic function, which is an S- shaped curve also called as sigmoid curve and is given by the equation:



#### **Logistic Regression Assumptions:**

* Logistic regression requires the dependent variable to be binary.
* For a binary regression, the factor level 1 of the dependent variable should represent the desired outcome.
* Only the meaningful variables should be included.
* The independent variables should be independent of each other·
* Logistic regression requires quite large sample sizes.
* Even though logistic (logit) regression is frequently used for binary variables (2 classes), it can be used for categorical dependent variables with more than 2 classes.
* In this case it’s called Multinomial Logistic Regression.

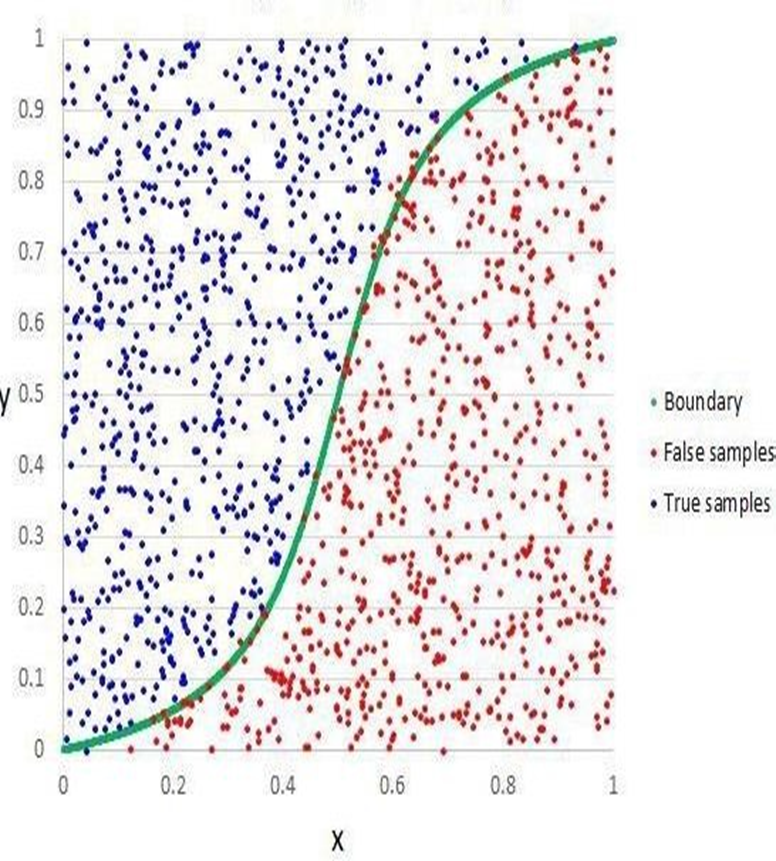


Fig 5.1: logistic regression

#### **5.3.2 Random Forest**

Random forest is a supervised learning algorithm used for classification and regression. But however, it is mainly used for classification problems. As we know, a forest is made up of trees and more trees means a more robust forest.

Similarly, random forests create decision trees on data samples and then get the prediction from each of them and finally select the best solution by means of voting. It is ensemble method which is better than a single decision tree because it reduces the over-fitting by averaging the result.

Working of Random Forest with the help of following steps:

* First, start with the selection of random samples from a given dataset.
* Next, this algorithm will construct a decision tree for every sample. Then it will get the prediction result from every decision tree.
* In this step, voting will be performed for every predicted result.
* Finally, select the most voted prediction results as the final prediction result. The following diagram will illustrate its working-

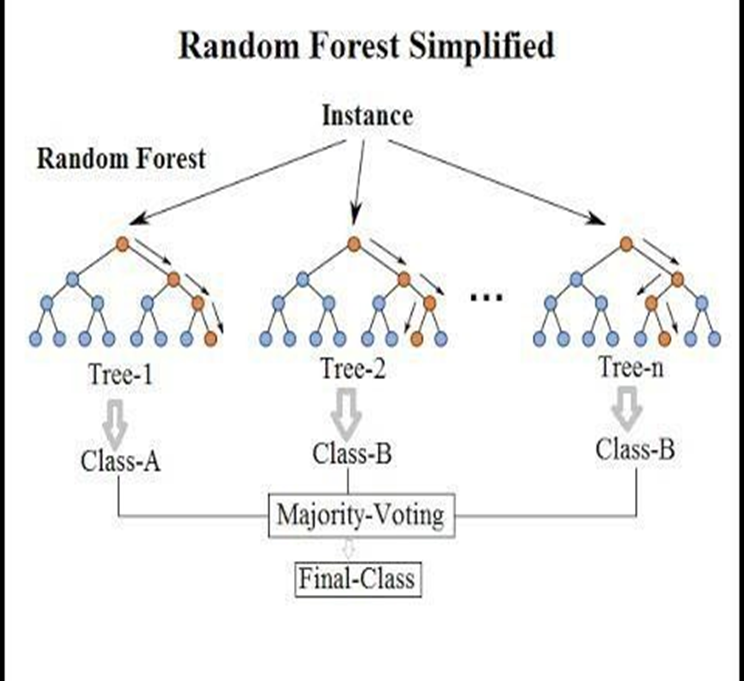


Fig 5.2: Random Forest

#### **5.3.3 Support Vector Machine (SVM)**

Support Vector Machine (SVM) is a supervised learning algorithm used for classification and regression tasks. It operates by finding the optimal hyperplane that best separates data points into different classes while maximizing the margin, which is the distance between the hyperplane and the nearest data points (support vectors).

Working of SVM with the help of following steps:

* Data Collection and Preprocessing: Gather labeled data points and preprocess them by scaling features for uniformity.
* Kernel Selection: Choose a kernel function (e.g., linear, polynomial, RBF) to transform data into a higher-dimensional space.
* Training: Optimize a cost function to find the parameters of the hyperplane that minimize errors while maximizing the margin. This involves identifying support vectors.
* Testing and Predictions: Use the trained model to classify new data points based on their distance from the hyperplane.
* Hyperparameter Tuning: Fine-tune parameters like the choice of kernel and regularization parameter using techniques like cross-validation.
* Evaluation: Assess the model's performance using metrics like accuracy or F1-score.



Fig 5.3: Support Vector Machine (SVM)

#### **5.3.4 Decision Tree Classifier**

A Decision Tree Classifier is a popular supervised learning algorithm used in machine learning for both classification and regression tasks. It operates by recursively partitioning the feature space into smaller regions, making decisions at each node based on the features' values.

Working of Decision Tree classifier with the help of following steps:

* Initialization: The algorithm begins with the entire dataset at the root node. It selects the best feature to split the dataset based on a criterion such as Gini impurity or information gain.
* Splitting: The selected feature is used to partition the data into subsets at each node. Each subset corresponds to a unique value of the selected feature. The process continues recursively for each subset until a stopping criterion is met, such as reaching a maximum depth or minimum number of samples per leaf.
* Decision Making: At each node, the algorithm makes decisions based on the feature value that optimally splits the data. This decision process continues until the tree reaches its maximum depth or another stopping criterion.
* Leaf Nodes: Once the tree reaches a stopping criterion, the final nodes are called leaf nodes or terminal nodes. Each leaf node represents a class label or a continuous value.
* Prediction: To make predictions for new data points, they are passed down the tree, and at each node, decisions are made based on the feature values until a leaf node is reached. The output of the leaf node is then used as the prediction.
* Model Evaluation: The performance of the decision tree model is evaluated using metrics such as accuracy, precision, recall, or F1-score, depending on the specific problem being addressed.



Fig 5.4: Decision Tree Classifier

### **5.4 FEASIBILITY STUDY**

A Feasibility Study is a preliminary study undertaken before the real work of a project starts to ascertain the likely hood of the project's success. It is an analysis of possible alternative solutions to a problem and a recommendation on the best alternative.

#### **5.4.1 Economic Feasibility:**

It is defined as the process of assessing the benefits and costs associated with a project's development. A proposed system, which is both operationally and technically feasible, must be a good investment for the organization. With the proposed system the users are greatly benefited as the users can be able to detect the fake news from the real news and are aware of most real and most fake news published in the recent years. This proposed system does not need any additional software and high system configuration. Hence the proposed system is economically feasible.

#### **5.4.2 Technical Feasibility:**

The technical feasibility infers whether the proposed system can be developed considering the technical issues like availability of the necessary technology, technical capacity, adequate response and extensibility. The project was decided to build using Python. Jupyter Notebook is designed for use in distributed environment of the internet and for the professional programmer it is easy to learn and use effectively. As the developing organization has all the resources available to build the system therefore the proposed system is technically feasible.

#### **5.4.3 Operational Feasibility:**

Operational feasibility is defined as the process of assessing the degree to which a proposed system solves business problems or takes advantage of business opportunities. The system is self-explanatory and doesn’t need any extra sophisticated training. The system has built-in methods and classes which are required to produce the result. The application can be handled very easily with a novice user. The overall time that a user needs to get trained is 14 less than one hour. As the software used for developing this application is economical and available in the market. Therefore, the proposed system is operationally feasible.

**CHAPTER 6**

**OBTAINED EXPERIMENTAL RESULT**

**6.1 DATASET ANALYSIS**

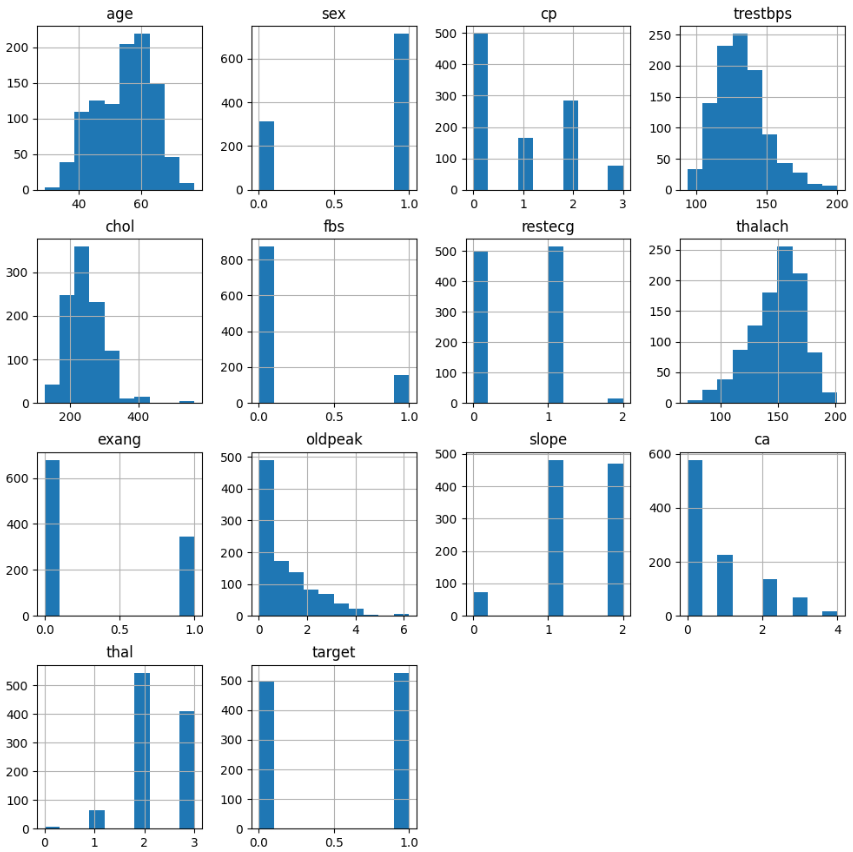
****

Fig 6.1: Dataset Analysis

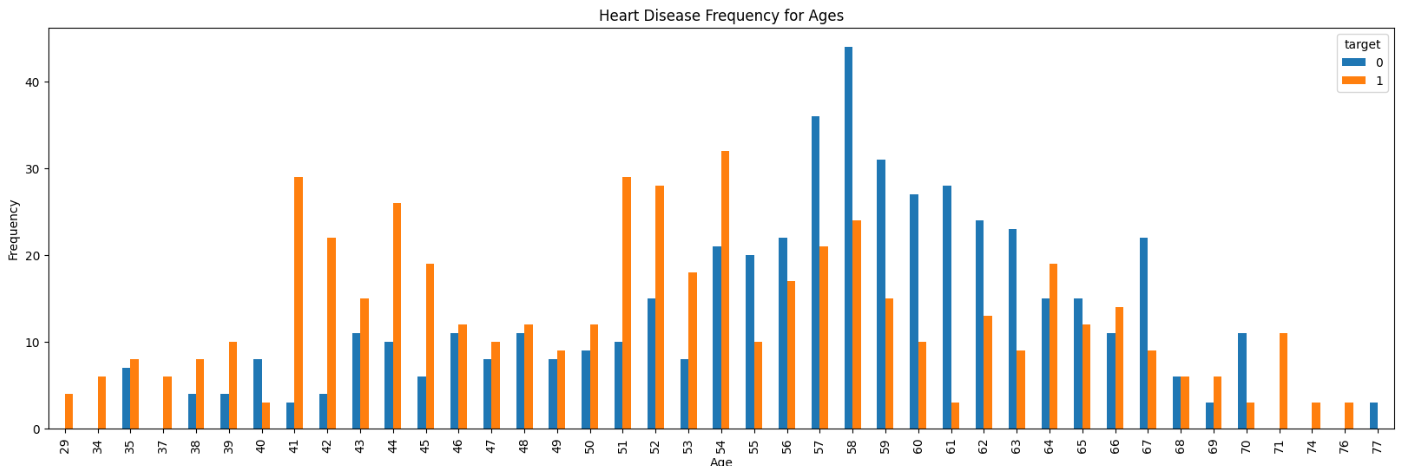


Fig 6.2: Heart Disease Frequency for Ages

**6.2 ACCURACY**

The figures display the accuracy scores of various Machine Learning Models, showcasing their performance in predicting heart disease. Understanding these scores aids in selecting the most effective model for early detection and treatment planning.

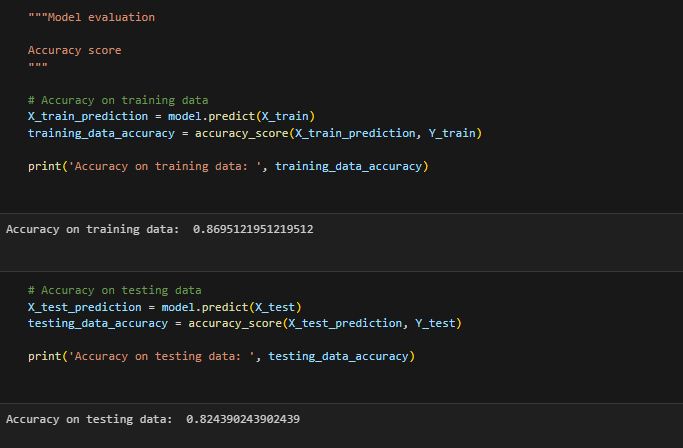


Fig 6.3: Training and Testing Data Accuracy using SVM

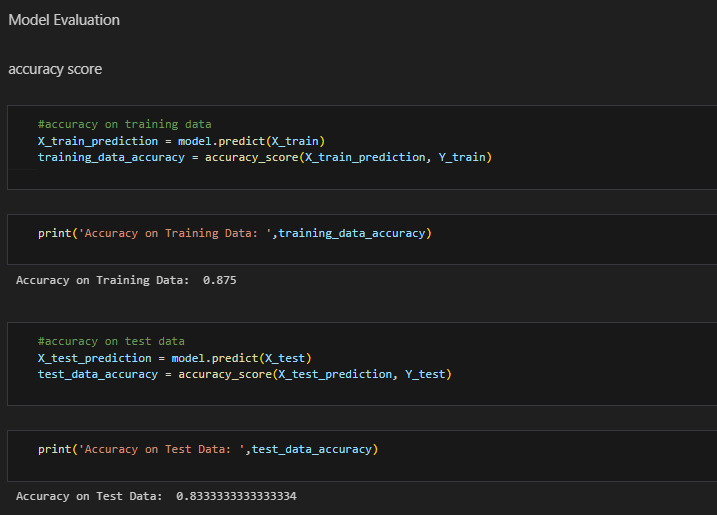


Fig 6.4: Training and Testing Data Accuracy using Logistic Regression

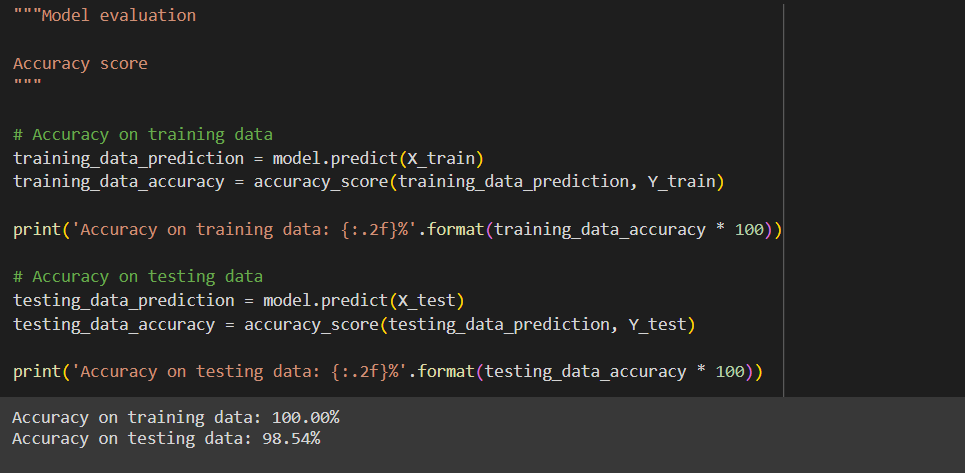


Fig 6.5: Training and Testing Data Accuracy using Decision Tree Clasifier

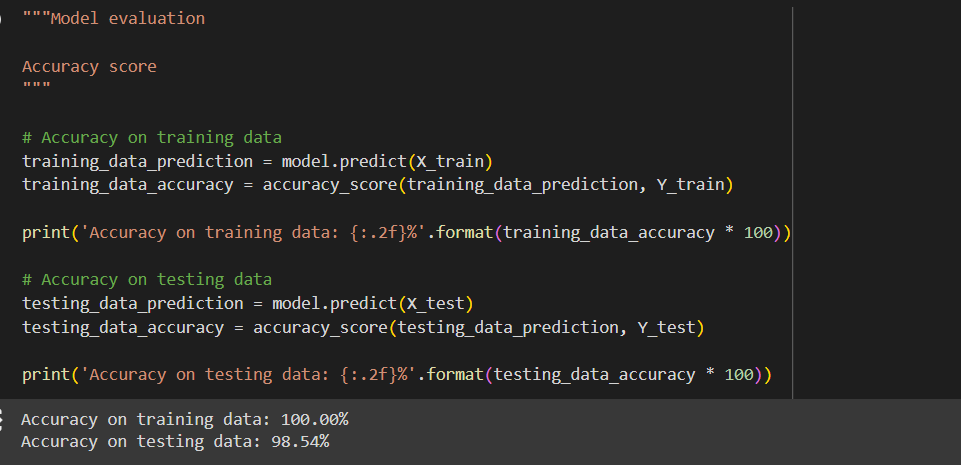


Fig 6.6: Training and Testing Data Accuracy using Random Forest Classifier

**6.3 USER INTERFACE**

The website presents the end results of heart disease prediction, providing users with valuable insights into their cardiovascular health. This facilitates informed decision-making and empowers proactive healthcare management for improved well-being.

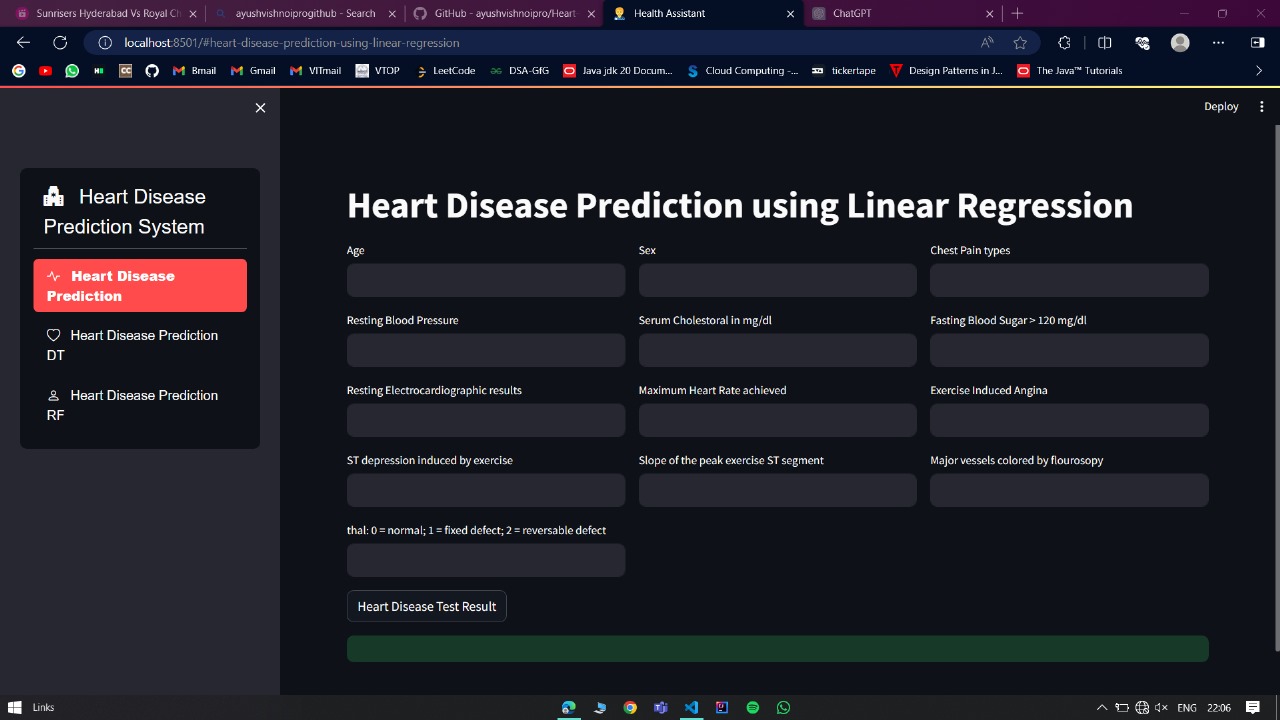


Fig 6.7: User Interface

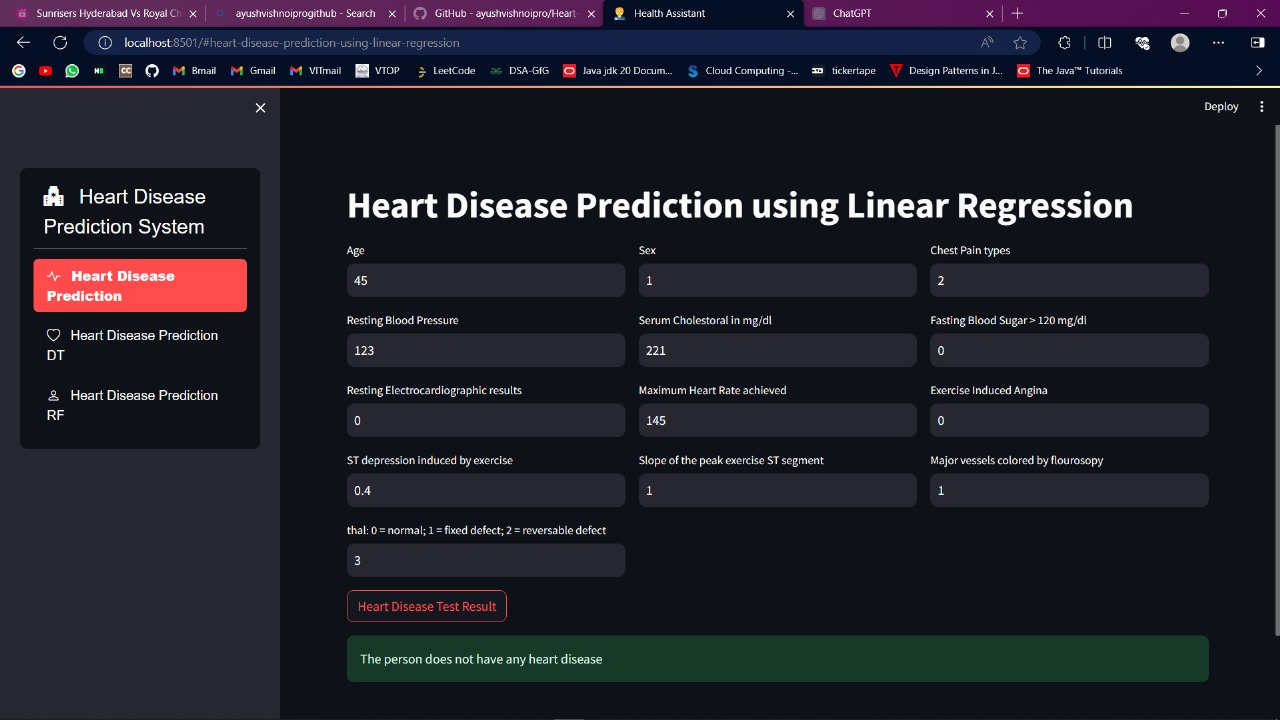


Fig 6.8: Results for heart disease of a particular patient

**CHAPTER 7**

**FUTURE SCOPE AND CONCLUSIONS**

**7.1 FUTURE SCOPE**

As illustrated before the system can be used as a clinical assistant for any clinicians. The disease prediction through the risk factors can be hosted online and hence any internet users can access the system through a web browser and understand the risk of heart disease. The proposed model can be implemented for any real time application. Using the proposed model other type of heart disease also can be determined. Different heart diseases as rheumatic heart disease, hypertensive heart disease, ischemic heart disease, cardiovascular disease and inflammatory heart disease can be identified.

Other health care systems can be formulated using this proposed model in order to identify the diseases in the early stage. The proposed model requires an efficient processor with good memory configuration to implement it in real time. The proposed model has wide area of application like grid computing, cloud computing, robotic modeling, etc. To increase the performance of our classifier in future, we will work on ensembling two algorithms called Random Forest and Adaboost. By ensembling these two algorithms we will achieve high performance.

**7.2 CONCLUSION**

In this project, we introduce about the heart disease prediction system with different classifier techniques for the prediction of heart disease. The techniques are Decision Tree Classifier, SVM Random Forest and Logistic Regression we have analyzed that the Decision Tree Classifier and Random Forest has better accuracy as compared to Logistic Regression and SVM. Our purpose is to improve the performance of the Decision Tree Classifier and Random Forest by removing unnecessary and irrelevant attributes from the dataset and only picking those that are most informative for the classification task.

**REFERENCES**

1. Abdul-Wahab,S.A., Al-Alawi,S.M. and El-Zawahry, Patterns of S02 emission: a refinery case study, Environmental modeling & software, 2002, 17, 563-570.
2. Aggarwal A.L, Sivacoumar R. and Goyal SK Air Quality Prediction: influence of model parameters and sensitivity analysis, Indian Journal of Environmental Protection, 1997, 17(9), 650-655.
3. <https://www.scribd.com/document/515626292/1-Heart-Disease-Prediction-Using-Machine-Learning>.
4. Dane Bertram, Amy Voida, Saul Greenberg, Robert Walker, “Communication, Collaboration, and Bugs: The Social Nature of Issue Tracking in Small, Collocated Teams”.
5. C. Kalaiselvi, PhD, “Diagnosis of Heart Disease Using K-Nearest Neighbor Algorithm of Data Mining”, IEEE, 2016
6. Chaitrali S. Dangare Sulabha S. Apte, Improved Study of Heart Disease Prediction System using Data Mining Classification Techniques” International Journal of Computer Applications (0975 – 888).