HEART DISEASE PREDICTION

A MINI PROJECT REPORT

Submitted by

A.MOHAMMED AYAAN(311518104022)

MUNNA MANISH BALAJI(311518104025)

R.K SAIKRISHNA THIWAKAR(311518104037)

In partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

MEENAKSHI SUNDARARAJAN ENGINEERING COLLEGE,

KODAMBAKKAM, CHENNAI-24

ANNA UNIVERSITY: CHENNAI 600 025

APRIL 2021

ANNA UNIVERSITY: CHENNAI 600 025

BONAFIDE CERTIFICATE

Certified that this mini project report **"HEART DISEASE PREDICTION"** is the bonafide work of "**A.MOHAMMED AYAAN (311517104022),MUNNA MANISH BALAJI(311517104025), RK.SAIKRISHNA THIWAKAR (311517104037)"** who carried out the project work under my supervision.

B.Monica Jenefer

SIGNATURE

Dr.B.MonicaJenefer,M.E,Ph.D

HEAD OF THE DEPARTMENT

Computer Science and Engineering

Meenakshi Sundararajan Engineering College,

No.363, Arcot Road, Kodambakkam,

Chennai -600 024.

C. Jerin Mahibha

SIGNATURE

Mrs.C.Jerin Mahibha, M.E.

SENIOR ASSISTANT PROFESSOR

Computer Science and Engineering

Meenakshi Sundararajan Engineering College,

No.363, Arcot Road, Kodambakkam,

Chennai -600 024.

Submitted for the project viva voce of Bachelor of Engineering in Computer Science and Engineering held on 10/08/2021.

INTERNAL EXAMINER **EXTERNAL EXAMINER**

.

ACKNOWLEDGEMENT

First and foremost we express our sincere gratitude to our Respected Correspondent **Dr.K.S.Lakshmi**, our beloved Secretary **Dr.K.S.Babai**, Principal **Dr.P.K.Suresh** and Dean Academics **Dr.K.Umarani** for their constant encouragement, which has been our motivation to strive towards excellence.

Our primary and sincere thanks goes to **Dr.B.Monica Jenefer**, Head of the Department, Department of Computer Science and Engineering, for her profound inspiration, kind cooperation and guidance.

We're grateful to **Mrs.C.Jerin Mahibha**, Internal guide, Senior Assistant Professor, Department of Computer Science and Engineering. We are extremely thankful and indebted for sharing expertise, and sincere and valuable guidance and encouragement extended to us. We would like to express our sincere gratitude to **Mr. Venkatesh** Assistant Professor, Department of Computer Science and Engineering for the constant supervision and providing necessary support during the course of our project.

Above all, we extend our thanks to God Almighty without whose grace and blessings it wouldn't have been possible.

ABSTRACT

Health and fitness are among the top priorities of people from all walks of life but it has not been used up to its full potential.Many lives can be saved if Heart Disease is predicted using its early symptoms.In this project, we aim to monitor the health parameters of the patients efficiently and use the monitored data combined with their medical history to predict whether the patient may suffer from Heart Disease or not.Our project aims to ease the job by automating the conventional methods. This is done in the four phases of: Data Capturing, Data Collection and Processing, Data Monitoring and Data Prediction.Using high accuracy heart rate sensors to capture the health parameters, a Ardunio to process the data, remote monitoring can be done efficiently.This real- time data will also be stored in the database for the prediction of Heart Disease.It is an automation of the conventional methods providing constant vigilance and care.Heart rate and the medical history of the patients can be used to predict the susceptibility of Heart Disease .

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 ABOUT THE PROJECT:

Technology is being used everywhere in our daily life to fulfill our requirements and aid our lives in every sphere including communication, travelling, entertainment etc. Health and fitness are among the top priorities of people from all walks of life but it has not been used up to its full potential.Many lives can be saved if Heart Disease is predicted using its early symptoms.In this project, we aim to monitor the health parameters of the patients efficiently and use the monitored data combined with their medical history to predict whether the patient may suffer from Heart Disease or not.And this project is purely based on heart disease prediction and not detection.Our project aims to ease the job by automating the conventional methods. This is done in the four phases of: Data Capturing, Data Collection and Processing, Data Monitoring and Data Prediction.Using high accuracy sensors to capture the health parameters, a Raspberry Pi to process the data, remote monitoring can be done efficiently.This real- time data will also be stored in the database for the prediction of Heart Disease.It is an automation of the conventional methods providing constant vigilance and care.Heart rate and the medical history of the patients can be used to predict the susceptibility of Heart Disease .The logistic regression(machine learning)algorithm is used and the health care data which classifies the patients whether they are having heart diseases or not according to the information in the record.The database server stores the captured parameters as well as the medical history of the patients, through which they can be remotely accessed.

1.2 DOMAIN OVERVIEW:

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

A subset of machine learning is closely related to computational Statistics, which focuses on making predictions using computers; but not all machine learning is statistical learning. The study of mathematical optimization delivers methods, theory and application domains to the field of machine learning. Data mining is a related field of study, focusing on exploratory data analysis through unsupervised learning. In its application across business problems, machine learning is also referred to as predictive analysis.

Machine learning involves computers discovering how they can perform tasks without being explicitly programmed to do so. It involves computers learning from data provided so that they carry out certain tasks. For simple tasks assigned to computers, it is possible to program algorithms telling the machine how to execute all steps required to solve the problem at hand; on the computer's part, no learning is needed. For more advanced tasks, it can be challenging for a human to manually create the needed algorithms. In practice, it can turn out to be more effective to help the machine develop its own algorithm, rather than having human programmers specify every needed step.

1.3 EXISTING SYSTEM

In the existing system, the input details are obtained from the patient. Then from the user inputs, using ML techniques heart disease is analyzed. Now, the obtained results are compared with the results of existing models within the same domain and found to be improved. The data of heart disease patients collected from the UCI laboratory is used to discover patterns with NN, DT, Support Vector machines SVM, and Naive Bayes.

1.4 PROBLEM STATEMENT:

Many patients require continuous monitoring of various parameters, which can be expensive and cumbersome. Thus, remote monitoring of patients is a major concern for the health care sector. Often the medical history of the patient is not readily available or compiled.Heart Disease is a very critical heart condition resulting in a large number of deaths. Many lives can be saved if Heart Disease is predicted using its early symptoms. We have proposed a system which will attempt to solve the above-mentioned problems. Sensors can be used to capture the vitals of the patients. The database server stores the captured parameters as well as the medical history of the patients, through which they can be remotely accessed. We expect that, using these vitals and the medical history, the logistic regression algorithm will efficiently predict Heart Disease. The proposed system can save time and effort, improving the health care of patients.

1.5 CHAPTER OVERVIEW

The project report is organized with various chapters that denote the various functionalities and aspects of the system being developed.

Chapter 1 gives a general description about the project. It represents the basic idea of the project and introduces the topics of the existing system and proposed system.

Chapter 2 deals with the related works of the project. A literature review for each related work is explained in detail.

Chapter 3 presents the system architecture and requirements. It specifies the hardware and software components that are required. It also lists the technologies used in the implementation of the project.

Chapter 4 explains the system design with the use of UML diagrams and the data flow diagrams.

Chapter 5 contributes a detailed description of different modules that are there in the design and how they are implemented.

Chapter 6 gives a detailed description of the different test cases that were performed on the system.

Chapter 7 provides the conclusion. It also elucidates how the project can be further enhanced.

CHAPTER 2

LITERATURE SURVEY

2.1 HEART DISEASE PREDICTION USING EVOLUTIONARY RULE LEARNING

This study eliminates the manual task that additionally helps in extracting the information directly from the electronic records. To generate strong association rules, we have applied frequent pattern growth association mining on the patient's dataset. This will facilitate in decreasing the amount of services and shown that overwhelming majority of the rules helps within the best prediction of coronary sickness

Disadvantages:

- Can improve accuracy using ML algorithms

2.2 PREDICTION OF HEART DISEASE USING A HYBRID TECHNIQUE IN DATA MINING CLASSIFICATION

Heart disease prediction is treated as the most complicated task in the field of medical sciences. Thus there arises a need to develop a decision support system for detecting heart disease of a patient. In this paper, an efficient genetic algorithm hybrid with the back propagation technique approach for heart disease prediction is proposed. The main objective of this paper is to develop a prototype which can determine and extract unknown knowledge (patterns and relations) related to heart disease from a past heart disease database record. It

can solve complicated queries for detecting heart disease and thus assist medical practitioners to make smart clinical decisions which traditional decision support systems were not able to. By providing efficient treatments, it can help to reduce costs of treatment.

Disadvantages:

- Hybrid models lack accuracy.

2.3 HEART DISEASE PREDICTION USING NAÏVE BAYES

Here, a web application that allows users to get instant guidance on their heart disease through an intelligent system online is proposed. The application is fed with various details and the heart disease associated with those details. The application allows users to share their heart related issues. It then processes user specific details to check for various illnesses that could be associated with it. Here we use some intelligent data mining techniques to guess the most accurate illness that could be associated with a patient's details. Based on the result, the system automatically shows the result to specific doctors for further treatment. The system allows users to view doctor’s details. The system can be used in case of emergency. The main goal of this system is to predict heart disease using data mining techniques such as Naive Bayesian Algorithm. Raw hospital data set is used and then preprocessed and transformed the data set. Then apply the data mining technique such as Naïve Bayes algorithm on the transformed data set. After applying the data mining algorithm, heart disease is predicted and the user is given the result based on the prediction whether the risk of heart disease is low, average or high.

Disadvantages:

- Can improve using better web development techniques.

2.4 EFFICIENT HEART DISEASE PREDICTION USING DECISION TREE ALGORITHM

The decision-tree algorithm is one of the most effective and efficient classification methods available. It has been shown that, by using a decision tree, it is possible to predict heart disease vulnerability in diabetic patients with reasonable accuracy. Classifiers of this kind can help in early detection of the vulnerability of a diabetic patient to heart disease. Preprocessing of a data set for the removal of duplicate records, normalizing the values used to represent information in the database. Clustering technique, simple k-means algorithm is used. Thus, the patients can be forewarned to change their lifestyles. This will result in preventing diabetic patients from being affected by heart diseases, thereby resulting in low mortality rates as well as reduced cost on health care for the state.

Disadvantages:

- Can improve accuracy using better dataset

2.5 DISEASE RISK PREDICTION BY USING CONVOLUTIONAL NEURAL NETWORK

This study proposes an efficient neural network with convolutional layers to classify significantly class-imbalanced clinical data. The data is curated from the National Health and Nutritional Examination Survey (NHANES) with the

goal of predicting the occurrence of Coronary Heart Disease (CHD). While the majority of the existing machine learning models that have been used on this class of data are vulnerable to class imbalance even after the adjustment of class-specific weights, the simple two-layer CNN exhibits resilience to the imbalance with fair harmony in class-specific performance. Given a highly imbalanced dataset, it is often challenging to simultaneously achieve a high class 1 (true CHD prediction rate) accuracy along with a high class 0 accuracy, as the test data size increases. The model architecture exhibits a way forward to develop better investigative tools, improved medical treatment and lower diagnostic costs by incorporating a smart diagnostic system in the healthcare system.

Disadvantages:

- Can improve accuracy using ML algorithms

2.6 MINING CONSTRAINED ASSOCIATION RULES TO PREDICT HEART DISEASE

In this paper, the prediction of heart disease with the help of association rules. A simple mapping algorithm is used. This algorithm treats attributes as numeric or categorical. It is used to convert medical records into transaction formats. Enhanced algorithms are used to minify restricted association rules. The mapping table is prepared and the attribute value is mapped to the item. Decision trees are used for data mining because they automatically divide numeric values . Split points selected in the decision tree are rarely used. Clustering is used to gain an overall understanding of the data.

Disadvantages:

- Performance of each constraint needs to be assessed.

CHAPTER 3

SYSTEM ARCHITECTURE

3.1 PROJECT ARCHITECTURE

System Architecture defines a comprehensive solution based on principles, concepts, and properties logically related and consistent with each other. The architecture has features, properties, and characteristics satisfying, as far as possible, the problem or opportunity expressed by a set of system requirements and life cycle concepts (e.g., operational, support) and is implementable through technologies (e.g., software, services, procedures, human activity).

The Architecture explains how the patient's details are taken and based on the data collected it can be predicted if the patient has heart disease or not.

3.2 SYSTEM ARCHITECTURE

The patient’s details are taken and formed into a dataset.The dataset is then segregated based on the attributes selected.After this the dataset is pre processed and the unwanted and irrelevant data is removed.Further Logistic Regression is applied and the prediction of heart disease is made.After collecting various amounts of results the accuracy is measured.

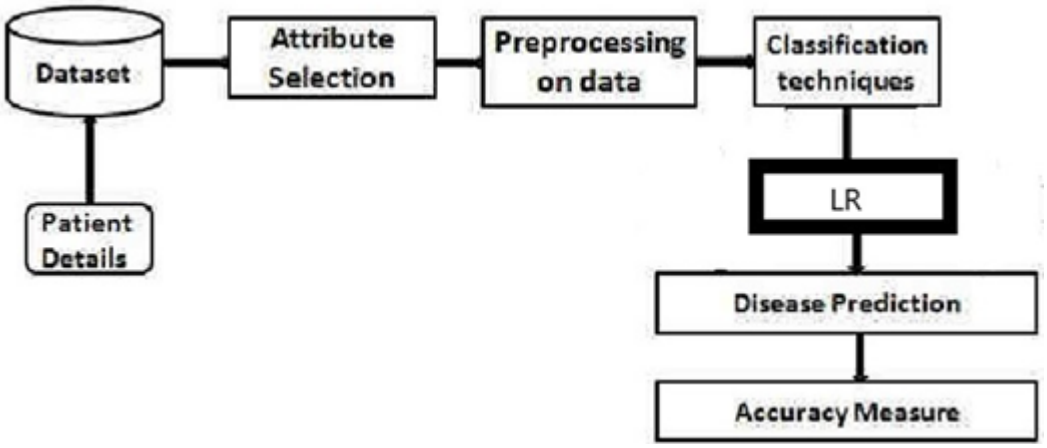

Figure 3.1 System Architecture

3.3 Hardware Requirements

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

S.No	REQUIREMENTS	RECOMMENDED	MINIMUM REQUIREMENTS
1	Operating System	Windows 10	Windows 8
2	RAM	4 GB	2 GB
3	HDD	1 TB	500 GB
4	Processor	Intel Quad Core PENTIUM IV	Intel Dual Core PENTIUM III
5	Arduino	12V	7V

TABLE 3.1 Hardware Requirements

3.4 SOFTWARE REQUIREMENTS

Requirements	Specification
TOOL	Arduino,Pytest,Jupyter Notebook
CODING LANGUAGE	Python

Table 3.2 Software Requirements

PYTHON:

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed. Python supports modules and packages, which encourages program modularity and code reuse.

ARDUINO:

Arduino is an open-source platform used for building electronics projects. Arduino consists of both a physical programmable circuit board often referred to as a microcontroller and a piece of software, or Integrated Development Environment that runs on your computer, used to write and upload computer code to the physical board.

PYTEST:

Pytest is a python based testing framework, which is used to write and execute test codes. In the present days of REST services, pytest is mainly used for API testing even though we can use pytest to write simple to complex tests, that is we can write codes to test API, database, UI, etc.

JUPYTER NOTEBOOK:

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.

CHAPTER 4

SYSTEM MODELING

4.1 UNIFIED MODELING LANGUAGE (UML)

Unified Modeling Language is a standardized modeling language consisting of an integrated set of diagrams, developed to help system and software developers for specifying, visualizing, constructing, and documenting the artifacts of software systems, 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. The UML is a very important part of developing object-oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects. Using the UML helps project teams communicate, explore potential designs, and validate the architectural design of the software.

The primary goals in the design of the UML as follows:

- Provide users with a ready-to-use, expressive visual modeling language so they can develop and exchange meaningful models.
- Provide extensibility and specialization mechanisms to extend the core concepts.
- Be independent of particular programming languages and development processes.
- Provide a formal basis for understanding the modeling language.
- Encourage the growth of the OO tools market.

- Support higher-level development concepts such as collaborations, frameworks, patterns and components.
- Integrate best practices.

4.2 USE CASE DIAGRAM

The use case diagram is used to define the core elements and processes that make up a system. The key elements are termed as “actors” and the processes are called “use cases”. The use case diagram shows which actors interact with each use case. This definition defines what a use case diagram is primarily made up of – actors and use cases.

In software and system engineering, a use case is a list of steps, typically defining interactions between a role (known in UML as an “actor”) and a system, to achieve a goal. The actor can be a human or an external system. In system engineering, use cases are used at a higher level than within software engineering, often representing missions or stakeholder goals.

The purposes of use case diagrams can be as follows:

1. Used to gather requirements of a system.
2. Used to get an outside view of a system.
3. Identify external and internal factors influencing the system.
4. Showing the interacting among the requirements are actors.

Use cases help in identifying the operations that can be performed by an actor. It gives a list of the various applications that can be utilized by the system. The actor can be a real time human or a system. It helps in identifying the various

modules present in the system. A single use case diagram captures a particular functionality of a system. Hence to model the entire system, a number of use case diagrams are used.

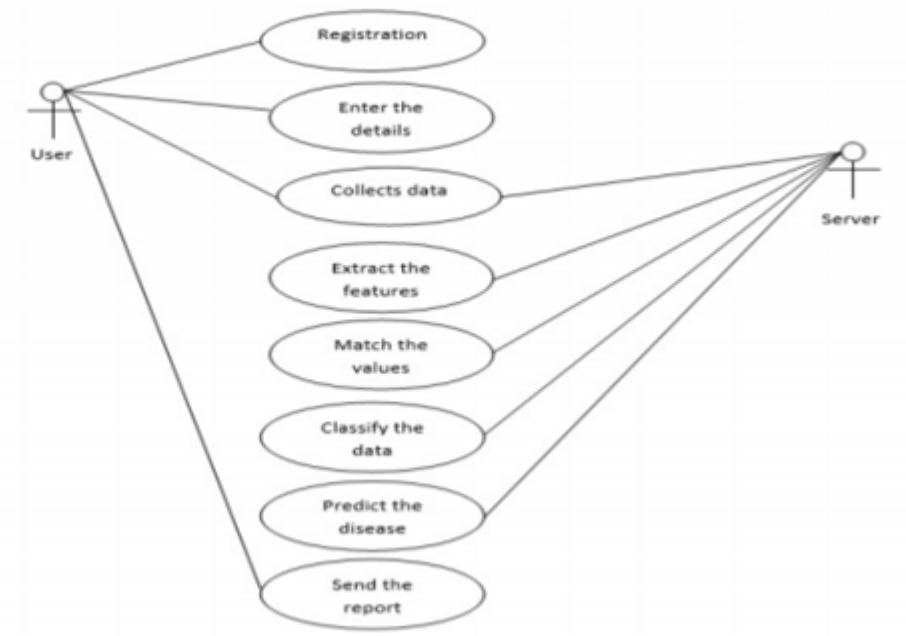

Figure 4.1 Use Case Diagram

In the system, the actors engaged are the patient and the server. The medical documents are uploaded by the patient to the system and stored in the server. The patient can view and upload documents.Various classifications are done by the server and then the result is also displayed.

4.3 CLASS DIAGRAM

Class diagram is a static diagram. It is the building block of every object-oriented system and helps in visualizing and describing the system. A class diagram depicts the structure of the system through its classes, their

attributes, operations and relationships among the objects. A class is a blueprint that defines the variables and methods common to all objects of a certain kind. Class diagram shows a collection of classes, interfaces, associations, collaborations, and constraints. The characteristics of Class Diagram are:

1. Each class is represented by a rectangle having a subdivision of three compartments - name, attributes and operations.

2. There are three types of modifiers which are used to decide the visibility of attributes and operations : + is used for public visibility, # is used for protected visibility, – is used for private visibility.

In the diagram, classes are represented with boxes that contain three compartments. The top compartment contains the name of the class. It is printed in bold and centered, and the first letter is capitalized. The middle compartment contains the attributes of the class. They are left-aligned and the first letter is lowercase. The bottom compartment contains the operations the class can execute. They are also left-aligned and the first letter is lowercase.

The main modules that are involved in this system are patient,and the database. The patient is the end user of the system who uploads the medical documents. The system stores the uploaded documents and the result is displayed to the patient.

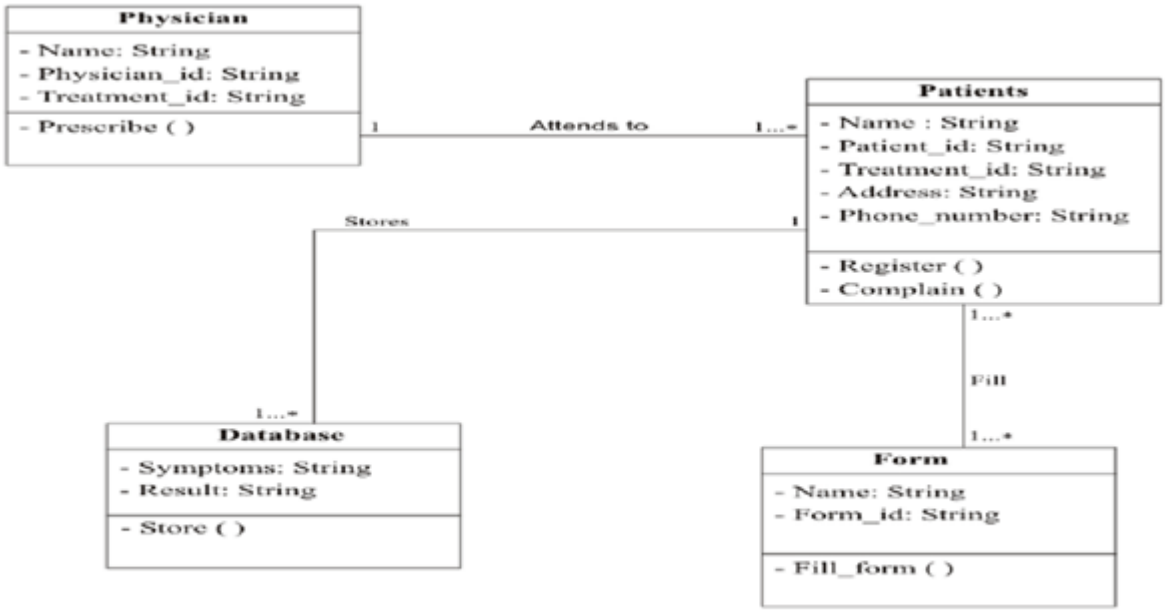

Figure 4.2 Class Diagram

4.4 SEQUENCE DIAGRAM

A sequence diagram is a kind of interaction diagram that shows how processes operate with one another and in which order. It is a construct of a Message Sequence Chart. A sequence diagram shows object interactions arranged in time sequence.

Sequence diagrams are a popular dynamic modeling solution in UML because they specifically focus on lifelines, or the processes and objects that live simultaneously, and the messages exchanged between them to perform a function before the lifeline ends.

It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario.

A sequence diagram shows different processes or objects that live simultaneously as parallel vertical lines (lifelines) and the messages exchanged between them and the order in which they occur as horizontal arrows.

The main purpose of the Sequence diagram is

- To capture the dynamic behavior of a system.
- To describe the message flow in the system.
- To describe the structural organization of the objects.
- To describe the interaction among objects.

Sequence diagrams can be used

- To model the flow of control by time sequence.
- To model the flow of control by structural organizations.
- For forward engineering
- For reverse engineering

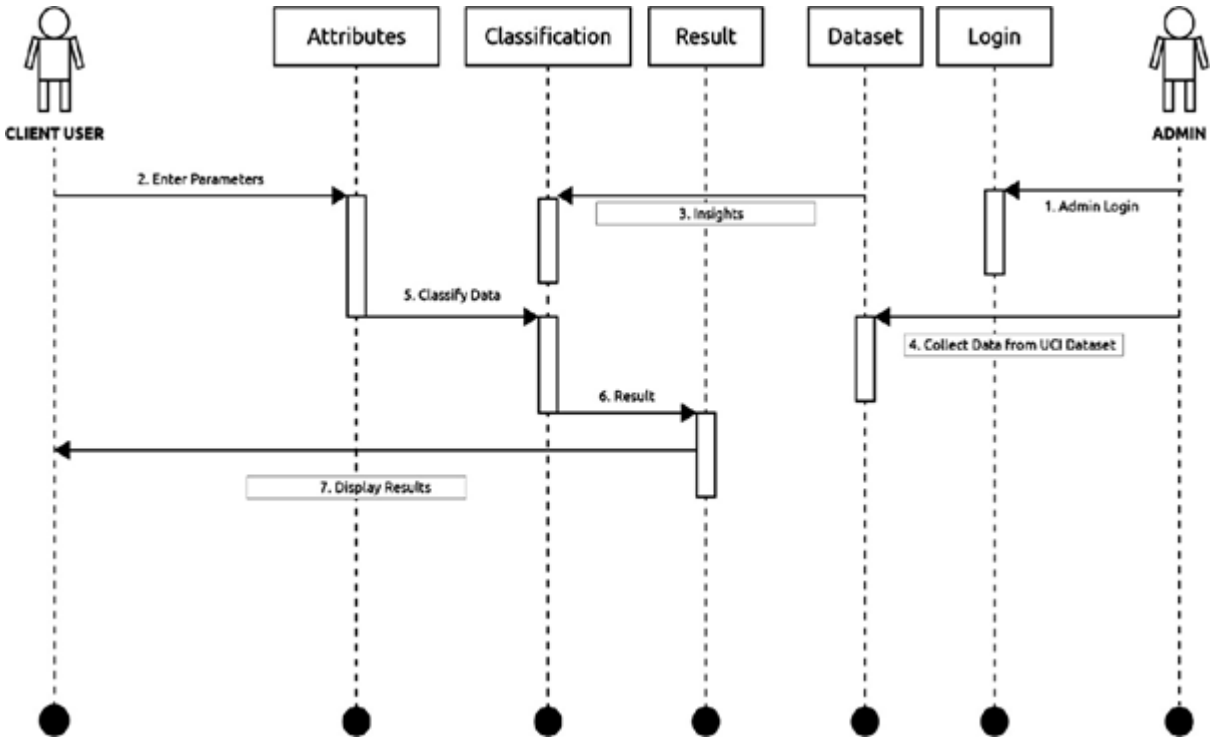

Figure 4.3 Sequence Diagram

In this system, the patient enters the parameters.The admin meanwhile collects the datasheet.Various insights of the data is found out and sent for further classification.Once the results have come out it is displayed to the patient.

4.5 COLLABORATION DIAGRAM

A collaboration diagram, also called a communication diagram or interaction diagram, is an illustration of the relationships and interactions among objects in the Unified Modeling Language (UML).

Collaboration diagrams convey the same information as sequence diagrams, but focus on object roles instead of the timings of messages. It illustrates messages being sent between classes and objects (instances).

Collaboration diagrams represent a combination of information taken from class, sequence and use case diagrams describing both the static structure

and dynamic behavior of a system. The collaboration diagram describes the messages or roles sent between objects.

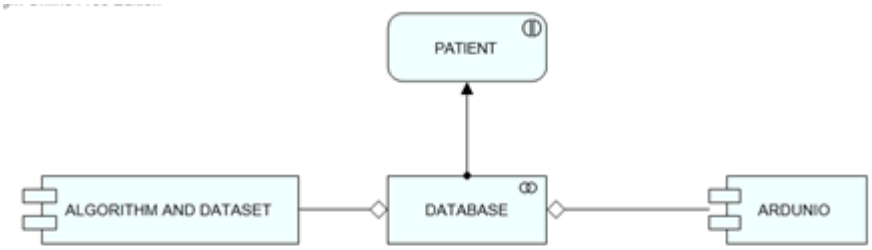

Figure 4.4 Collaboration Diagram

In this system, the patient, system and database are the lifeline elements of the system. The algorithm performs the required action**.**

4.6 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 are intended to model both computational and organizational processes (i.e., workflows), as well as the data flows intersecting with the related activities. Although activity diagrams primarily show the overall flow of control, they can also include elements showing the flow of data between activities through one or more data stores.

Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system. The control flow is drawn from one operation to another. This flow can be sequential, branched, or concurrent.

Activity diagrams deal with all types of flow control by using different elements such as fork, join, etc. Activity diagrams are constructed from a limited number of shapes, connected with arrows.

The most important shape types:

- rounded rectangles represent actions;
- diamonds represent decisions;
- bars represent the start (split) or end (join) of concurrent activities;
- a black circle represents the start (initial node) of the workflow;
- an encircled black circle represents the end (final node).

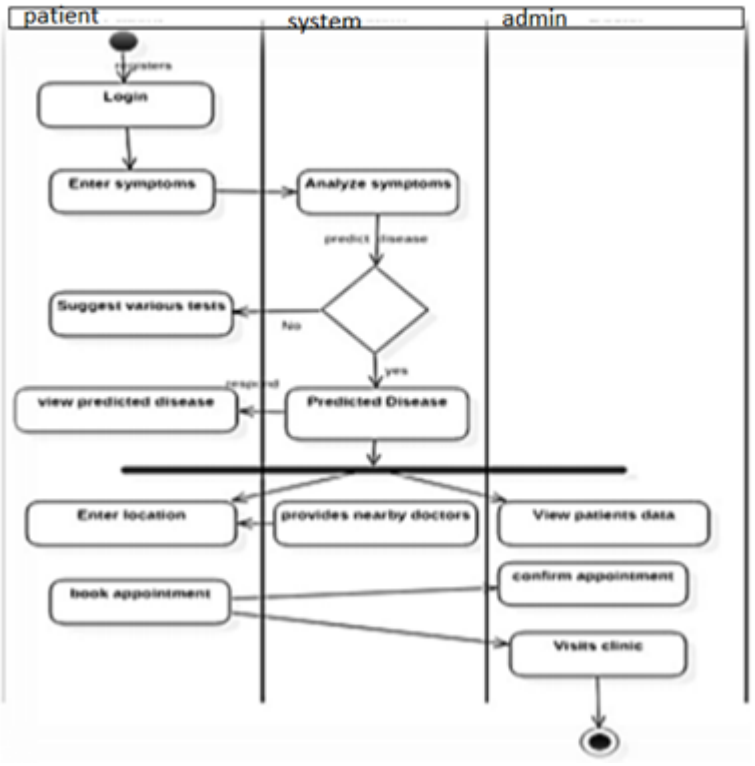

Figure 4.5 Activity Diagram

In this system, initially the patient logs in to the system and enters his details,the system based on the details provided predicts if the patient has heart disease or not and the result is sent to the patient.The admin then views the data and the system is closed.

4.7 STATE CHART DIAGRAM

A State chart diagram describes a state machine. State machine can be defined as a machine which defines different states of an object and these states are controlled by external or internal events. It describes different states of a component in a system. The states are specific to a component/object of a system. State chart diagrams are used to model the dynamic nature of a system. They define different states of an object during its lifetime and these states are changed by events. State chart diagrams are useful to model the reactive systems.

State chart diagram describes the flow of control from one state to another state. States are defined as a condition in which an object exists and it changes when some event is triggered. The most important purpose of State chart diagrams is to model the lifetime of an object from creation to termination.

The main purposes of using State chart diagrams

- To model the dynamic aspect of a system.
- To model the lifetime of a reactive system.
- To describe different states of an object during its life time
- an encircled black circle represents the end (final node).

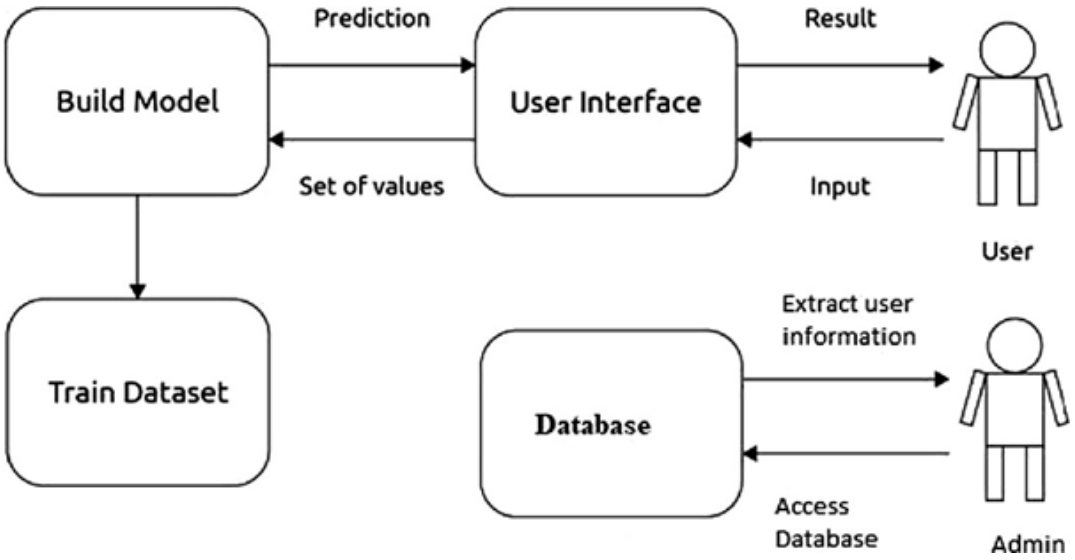

Figure 4.6 State Chart Diagram

In this system, the patient signs up and performs various activities like viewing and uploading documents.The entered values in the system are used as a training dataset and a model is built to predict the heart disease.

Meanwhile the admin can access the database and extract data.

4.8 COMPONENT DIAGRAM

In the Unified Modeling Language, a component diagram depicts how components are wired together to form larger components or software systems. They are used to illustrate the structure of arbitrarily complex systems. Component diagrams are different in terms of nature and behavior. Component diagrams are used to model the physical aspects of a system. Physical aspects are the elements such as executables, libraries, files, documents, etc. which reside in a node.

Component diagrams are used to visualize the organization and relationships among components in a system. These diagrams are also used to make executable systems. Component diagram is a special kind of diagram in UML.

The purpose is also different from all other diagrams. It does not describe the functionality of the system but it describes the components used to make those functionalities. Component diagrams can also be described as a static implementation view of a system. Static implementation represents the organization of the components at a particular moment.

The purpose of the component diagram can be summarized as Visualizing the components of a system.

- Construct executables by using forward and reverse engineering.
- Describe the organization and relationships of the components

Figure 4.7 Component Diagram

In this system, there are four main components- the design ui, the patient details, the dataset used and the database which is the file storage component.

4.9 PACKAGE DIAGRAM

Package diagram is a UML structure diagram which shows packages and dependencies between the packages. The package diagram shows the arrangement and organization of the model elements in a middle to large scale project. The package diagram can show both the structure and dependencies

between subsystems or modules. A package is rendered as a tabbed folder – a rectangle with a small tab attached to the left side of the top of the rectangle. If the members of the package are not shown inside the package rectangle, then the name of the package should be placed inside. The members of the package may be shown within the boundaries of the package. In this case, the name of the package should be placed on the tab. A diagram showing a package with content can show only a subset of the contained elements according to some criterion.

Members of the package may be shown outside of the package by branching lines from the package to the members. A plus sign (+) within a circle is drawn at the end attached to the namespace (package).

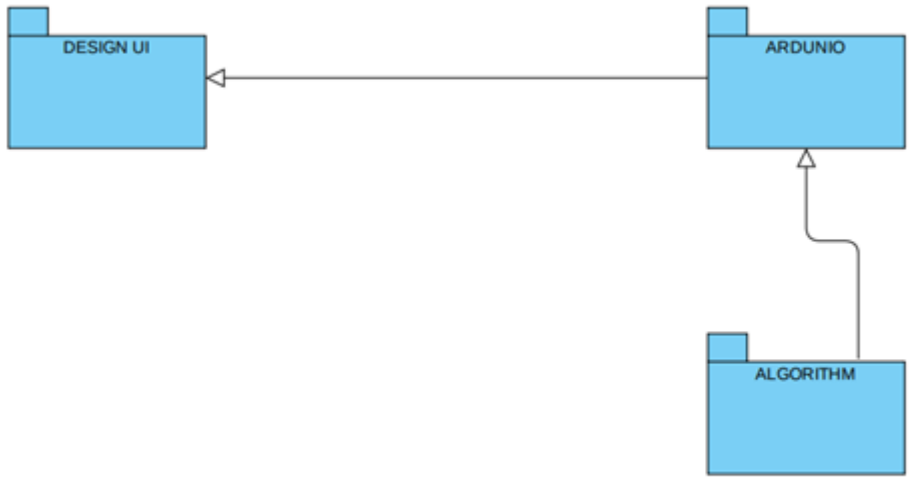

Figure 4.8 Package Diagram

In this system, there are packages that are necessary for any process or system namely User Interface, Database and Technical. The user interface is

where the user interacts with the system Database is used to store the medical documents uploaded by the user. The algorithm and dataset are used to perform analysis on the patient details.

4.10 DEPLOYMENT DIAGRAM

A deployment diagram in the Unified Modeling Language models the physical deployment of artifacts on nodes. To describe a website, for example, a deployment diagram would show what hardware components (“nodes”) exist (e.g., a web server, an application server, and a database server), what software components (“artifacts”) run on each node (e.g., web application, database), and how the different pieces are connected (e.g., JDBC, REST, RMI).

The nodes appear as boxes, and the artifacts allocated to each node appear as rectangles within the boxes. Nodes may have sub nodes, which appear as nested boxes. A single node in a deployment diagram may conceptually represent multiple physical nodes, such as a cluster of database servers. Deployment diagrams are used by system engineers.

The purposes of deployment diagrams can be as follows:

1. Visualize the hardware topology of a system.
2. Describe the hardware components used to deploy software components.
3. Describe the runtime processing nodes.

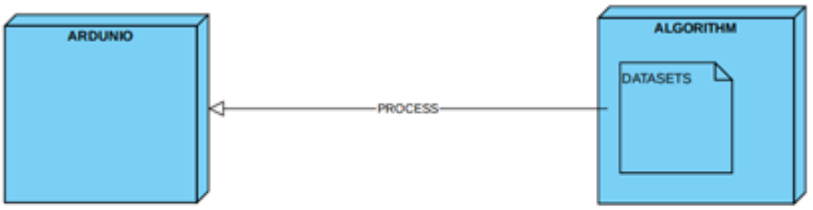

Figure 4.9 Deployment Diagram

In this system, the data from arduino is taken and using the appropriate algorithm on the datasets the patient's results are found.

CHAPTER 5

SYSTEM IMPLEMENTATION

5.1 PROPOSED SYSTEM

A system for predicting the chances of people who might get heart diseases with the help of algorithms in a jupyter environment is proposed. Firstly a dataset with the collection of people's health record which contains details like (cholesterol levels,fat levels etc..),is collected.Then with the help of the logistic regression and the values of dataset of those given number of people ,the system calculates the number of people and their chance of getting a heart disease.The system also produces the accuracy of which the system is as worked with the given set of dataset.

In the system construction, the jupyter notebook is an application based on Python,it is a cell based format ,which can be executed individually or in group cells.The dataset is added with the logic in the system and then the output is obtained.

5.2 MODULE DESCRIPTION

In the system construction, the jupyter notebook is an application based on Python,it is a cell based format ,which can be executed individually or in group cells.The dataset is added with the logic in the system and then the output is obtained.

A dataset with the collection of people's health record which contains details like (cholesterol levels,fat levels etc..),is collected.Then with the help of the

logistic regression and the values of dataset of those given number of people ,the system calculates the number of people and their chance of getting a heart disease.The system also produces the accuracy of which the system is as worked with the given set of dataset.

5.2.1.PROCESS OF UPLOADING DATASET:

The dataset values are saved as a csv file .They are then uploaded using the reading the values and then rereading the information and then assigning it as target.

5.2.2. ANALYSING THE DATASET RECORDS:

The information collected from the dataset is taken and then arranged in an orderly manner and then the initial graph is plotted.Then with the categorized basis the graphs are plotted to find and train the system on the information from the dataset. Then the correlation matrix which is the finally and crucial graph by which the values for the people close to the proximity of getting an heart disease is got.Then has a regression method-The K-neighbours classifier(present best predicting algorithm in regression method) is applied on the graph.The graph plotted by this method ,the

1) Abscissa is the number of people who are most likely to get the disease
2) Ordinate is the percentage of those people getting heart disease with the present data collected from them.

In the end the system also calculates its efficiency of the work it did on the dataset and displays the percentage.

The data-set provides the patient's information.This database contains 76 attributes, but all published experiments refer to using a subset of 14 of them. In

particular, the Cleveland database is the only one that has been used by ML researchers to this date. The "goal" field refers to the presence of heart disease in the patient. It is integer valued from 0 (no presence) to 4.Each attribute is a potential risk factor.The 14 attributes used are:

1. age.
2. sex.
3. cp- chest pain type (4 values).
4. trestbps- resting blood pressure.
5. chol- serum cholesterol in mg/dl.
6. fbs- fasting blood sugar > 120 mg/dl.
7. Rest ceg- resting electrocardiographic results (values 0,1,2).
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- test required for patient suffering from pain in chest or difficulty in breathing.3 = normal; 6 = fixed defect; 7 = reversible defect.
14. target- It is the final column of the dataset. It is a class or label Column.

Libraries used:

1. Pandas.
2. Matplotlib.
3. Seaborn.
4. Scikit-learn(sklearn).

Scikit-learn(sklearn):

1. Logistic Regression: Classification model.
2. Stratified Shuffle Split: used to split the data-set into train and test sets, especially good for unbalanced data.
3. confusion_matrix: the visualization of the performance of an algorithm.
4. accuracy_score: this function computes model accuracy.

CHAPTER 6

SYSTEM TESTING

6.1 INTRODUCTION

Software testing is a process of evaluating a software in a comprehensive manner in order to verify whether it is running in the desired fashion and to identify any bugs present if any. This is done in order to keep the quality of the deliverable high.

In accordance with the standard of ANSI, the definition of Software Testing is – A process of analyzing a software item to detect the differences between existing and required conditions (i.e., defects) and to evaluate the features of the software item.

6.2 TESTING APPROACHES

There are two main kinds of testing approaches under software testing, which are:

i. White Box Testing

ii. Black Box Testing

6.2.1 WHITE BOX TESTING

White box testing involves analyzing the internal structures of the code and not merely the functionality. It is used to identify any errors present in the code structure. It is usually done for unit testing although it can be done for integration and system testing too. White box testing is used for debugging code within a subsystem, between subsystems and so on. In white-box testing, an internal perspective of the system, as well as programming skills, are used to design test cases. It is also called Glass Box, Clear Box, Structural Testing.

Advantages of White Box Testing

- White Box Testing has simple and clear rules to let a tester know when the testing is done.
- White Box Testing techniques are easy to automate, this results in a developer having to hire fewer testers and smaller expenses.
- It shows bottlenecks which makes the optimization quite easy for the programmers.

6.2.2 BLACK BOX TESTING

As opposed to white box testing, black box testing is used to analyze the functionality of the software. It can be done in all levels of testing from unit testing to system testing. It is generally done for testing higher levels of code. It is also called Behavioral/Specification-Based/Input-Output Testing.

Advantages of Black Box Testing

- Black box tests are always executed from a user's point of view since it would help in exposing discrepancies significantly.
- Black box testers also do not need to know any programming languages.

6.3 TESTING LEVELS

Tests are grouped together based on the level of detailing they contain.The purpose of levels of testing is to make software testing systematic and easily identify all possible test cases at a particular level. In general, there are four levels of testing:

i. Unit Testing

ii. Integration Testing

iii. System Testing

iv. Acceptance Testing.

6.3.1 UNIT TESTING

A Unit is a smallest testable portion of a system or application which can be compiled, linked, loaded, and executed. This kind of testing helps to test each module separately.

6.3.2 INTEGRATION TESTING

Integration means combining. In this testing phase, different software modules are combined and tested as a group. Integrating testing checks the data flow from one module to other modules.

6.3.3 SYSTEM TESTING

System testing is performed on a complete, integrated system. It does checking of the system's compliance as per the requirements. It tests the overall interaction of components. It involves load, performance, reliability and security testing. System testing most often the final test to verify that the system meets the specification. It evaluates both functional and non-functional needs.

6.3.4 ACCEPTANCE TESTING

Acceptance testing is a test conducted to find if the requirements of a specification or contract are met as per its delivery. Acceptance testing is basically done by the user or customer. However, other stockholders can be involved in this process.

6.4 TESTING TYPES

There are two types of testing which are classified as such based on the manner in which the testing is done. They are:

i. Manual Testing

ii. Automation Testing

6.4.1 MANUAL TESTING

Manual testing is the process of testing software by hand. This usually includes verifying all the features specified in requirements documents, but often also includes the testers trying the software with the perspective of their end users in mind. Manual test plans vary from fully scripted test cases, giving testers detailed steps and expected results to high-level guides that steer exploratory testing sessions.

6.4.2 AUTOMATION TESTING

Automation testing is the process of testing the software using an automation tool to find the defects. In this process, testers execute the test scripts and generate the test results automatically by using automation tools.

6.5 JUPYTER NOTEBOOK TESTING USING PYTEST

The jupyter notebook used is tested with the help of pytest.It is an exclusive testing tool to python which uses unit based testing. The use of nbmake which is a pytest plugin to automate end-to-end testing of notebooks.The notebook is (stored has .ipynb file),is first collected.Then with the help of the the nbmake plugin ,the cells of the jupyter notebook is tested and passed if there is no error.

6.6 TEST RESULTS

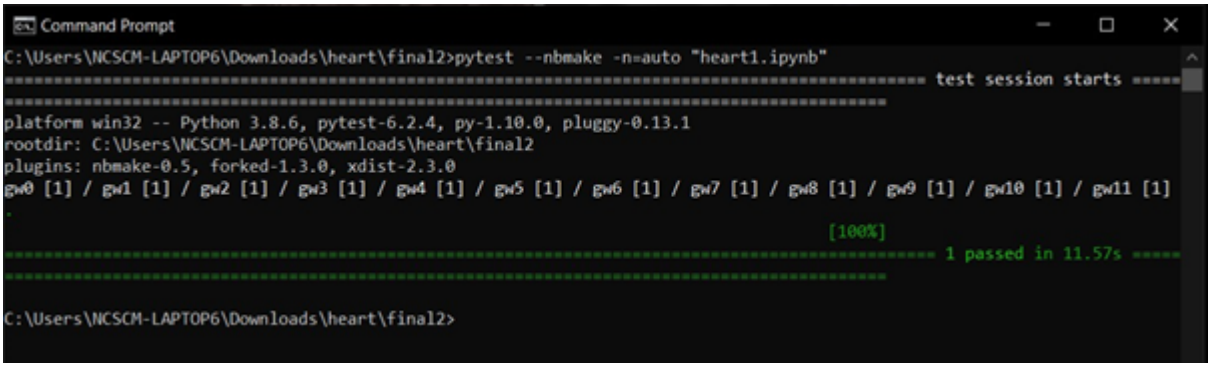

Figure 6.1 Pytest Passed

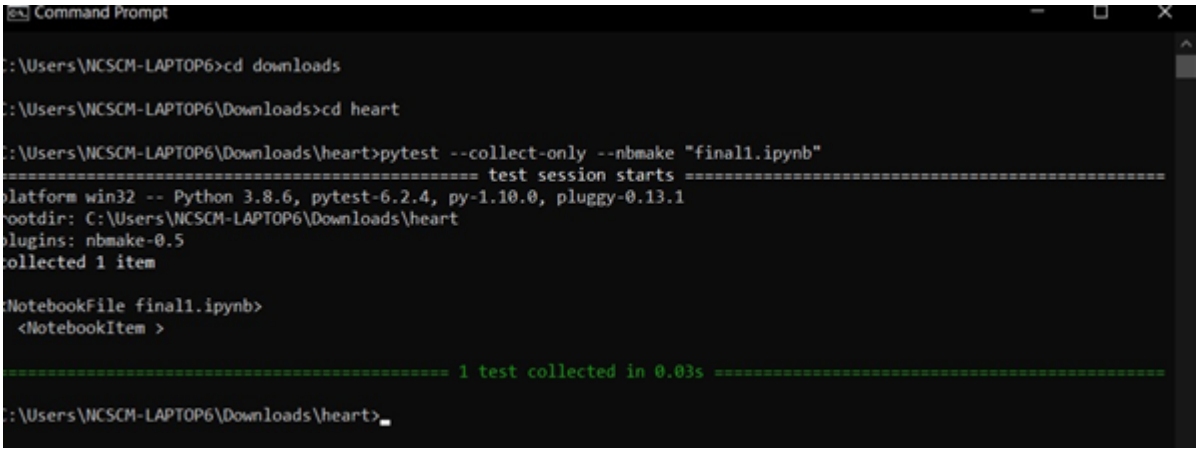

Figure 6.2 Pytest Collection

CHAPTER 7

CONCLUSION AND FUTURE ENHANCEMENT

7.1 CONCLUSION

A management system for storing medical documents is created with the help of React for creating the frontend and InterPlanetary File System (IPFS) for storing the patient's medical records on a blockchain environment. The decentralised structure is achieved with the help of IPFS. Also the stimulation of the blockchain environment has been done using Ganache and a cryptocurrency wallet namely Metamask, an extension on the chrome browser that enables the user to clearly comprehend the gas price involved in each transaction. Also, a neural-named entity recognition and multi-type normalization tool called BERN is used for biomedical text mining from the uploaded medical records for analysing the extracted keywords as drugs or diseases.

In this project, we aim to monitor the health parameters of the patients efficiently and use the monitored data combined with their medical history to predict whether the patient may suffer from HD (Heart Disease) or not. Also, the medical history of the patients will be updated regularly and stored in a systematic manner for quick references. We hope this proposed system will prove useful in saving time as well as lives. This system can be expanded by using an Arduino along with The Raspberry Pi to accommodate up to 120 sensors. Also, the patient data can be stored on the cloud, to access remotely. Security provisions can be made to protect the privacy of the patients and the integrity of the data .With the above modifications, the system will be more

scalable,efficient and secure.And this project mainly aims to predict heart disease and not for detection.

7.2 FUTURE ENHANCEMENT

However, in the future we would like to take in enormous number of patient records with various other attributes.We would like to deploy this project as an app and through the app the patient may be able to know the best possible and suitable treatment for himself.We would also work on the accuracy measure of the project as well.

APPENDIX-I

SCREENSHOTS

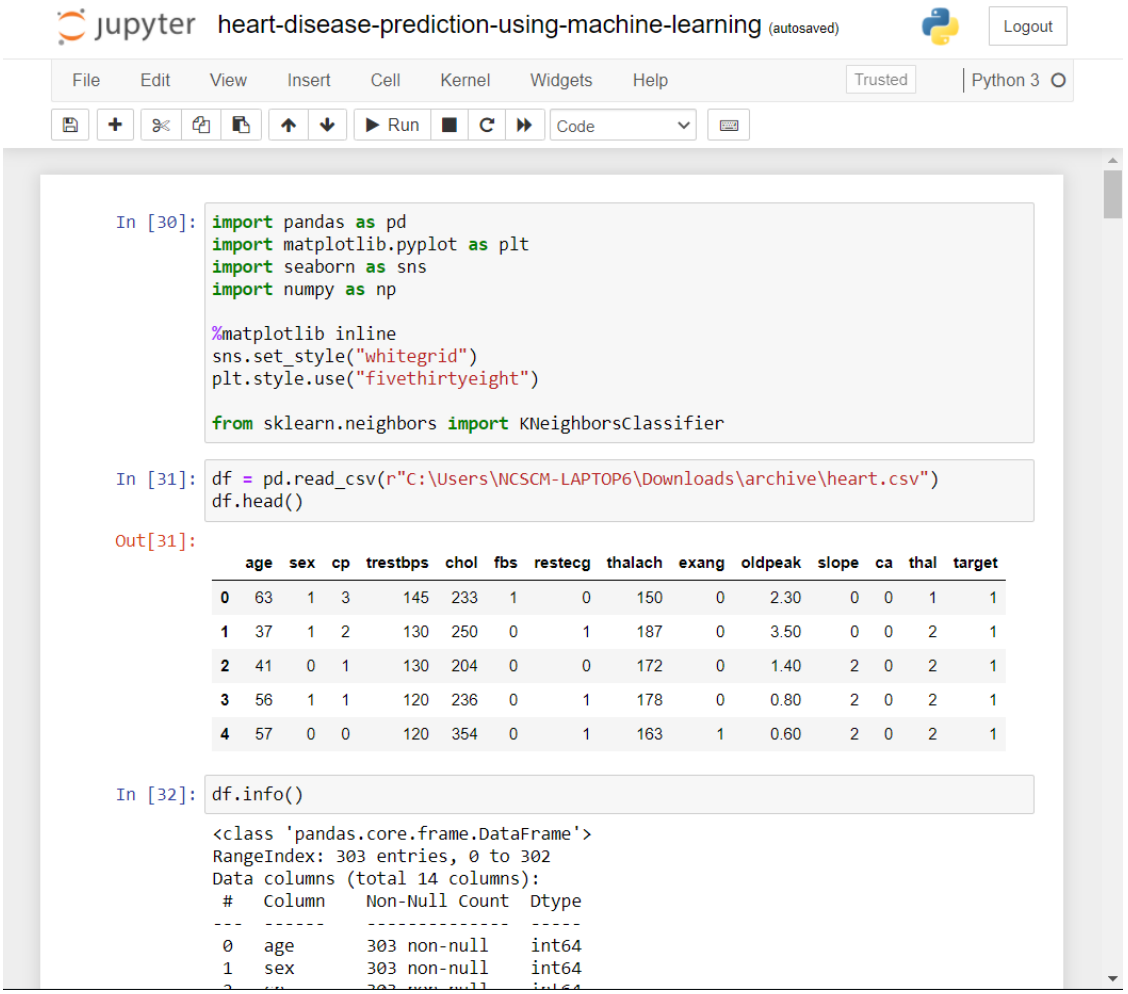

Figure A.1 UPLOADING AND READING OF DATASET

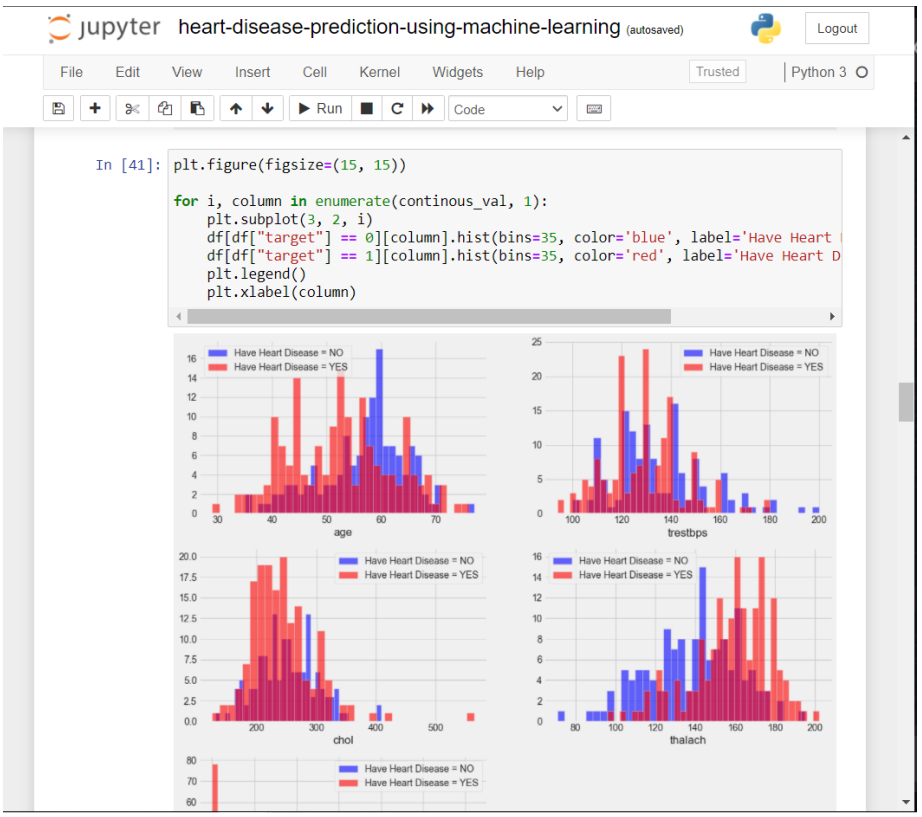

Figure A.2 USING'S THE PATIENTS DATASET TARGET VALUE IS CALCULATED BY PLOTTING GRAPHS

Figure A.3 USING TARGET VALUE GRAPH OTHER GRAPHS ARE PLOTTED

Figure A.4 THE CORRELATION MATRIX IS PLOTTED

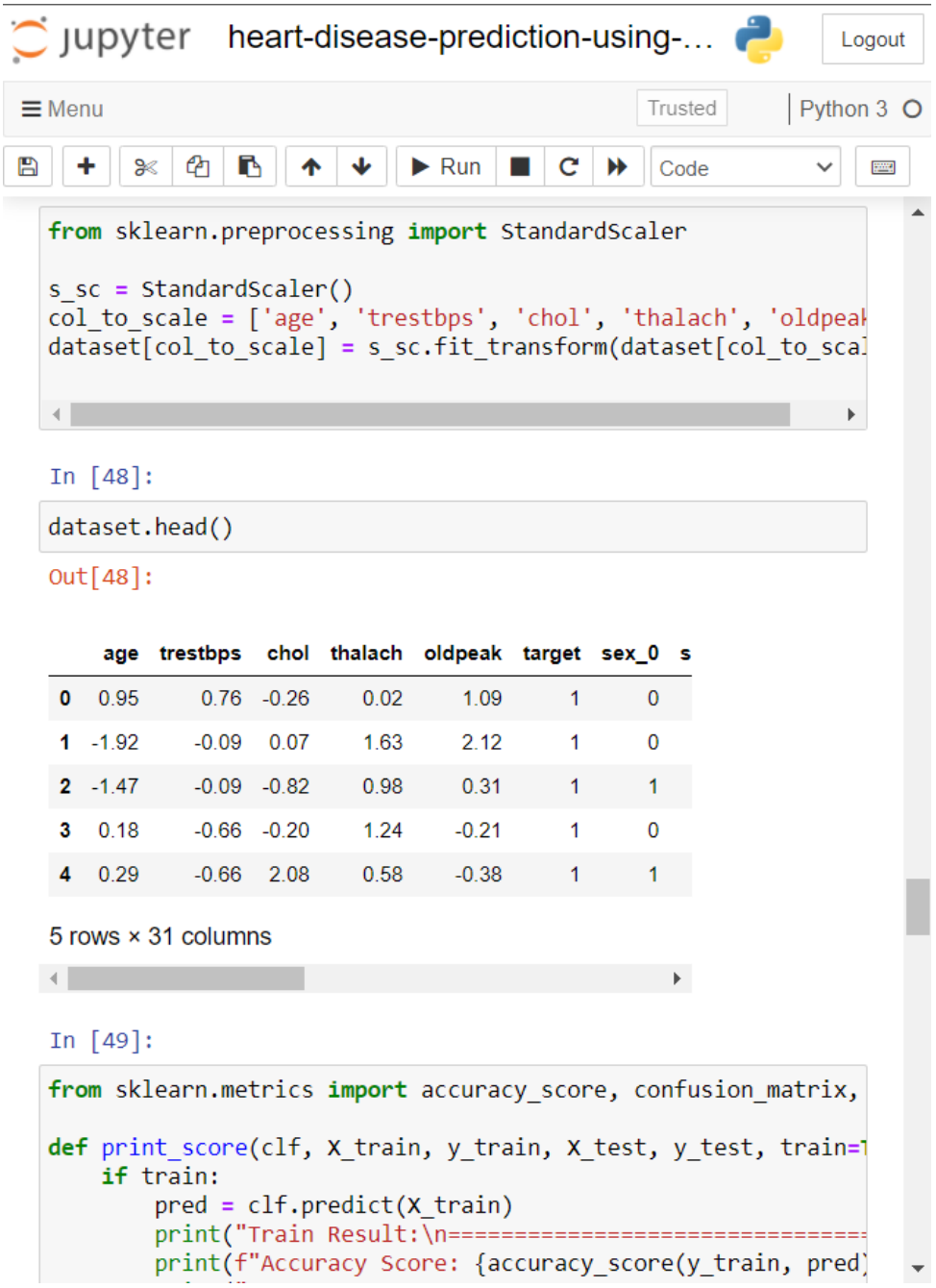

Figure A.5 TO PREDICT THROUGH K-NEIGHBORS CLASSIFIER DATA IS COLLECTED FROM CORRELATION MATRIX

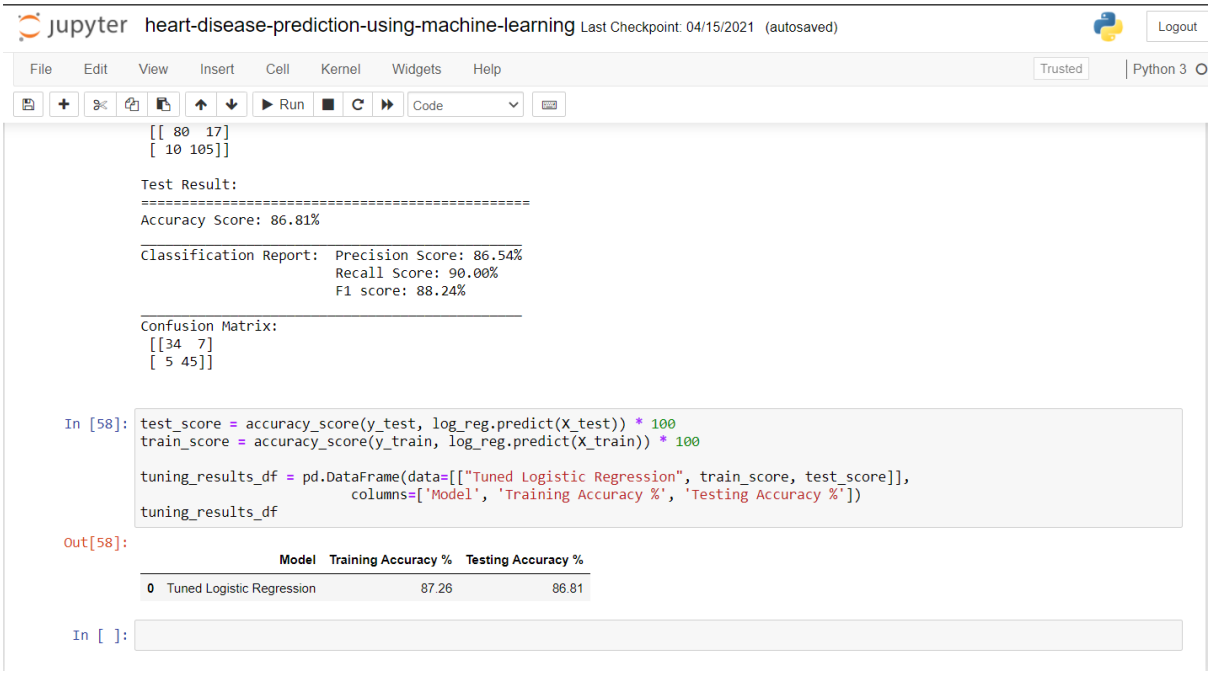

Figure A.6 THE ACCURACY CHECK ACHIEVED DURING THE PREDICTING PHRASE

REFERENCES

1. Mohan, Senthilkumar , Chandrasegar Thirumalai , and Gautam Srivastava,"Effective heart disease prediction using hybrid machine learning techniques." IEEE Access 7 (2019): 81542-81554.
2. Purushottam , Kanak Saxena and Richa Sharma, "Efficient heart disease prediction system." Procedia Computer Science 85 (2016): 962-969.
3. Shinde R, Arjun S, Patil P & Waghmare J (2015). An intelligent heart disease prediction system using k-means clustering and Naïve Bayes algorithm. International Journal of Computer Science and Information Technologies, 6(1), 637-9.
4. Singh, Yeshvendra K., Nikhil Sinha, and Sanjay K. Singh, "Heart Disease Prediction System Using Random Forest", International Conference on Advances in Computing and Data Sciences. Springer, Singapore, 2016.
5. Vembandasamy K, Sasipriya R, Deepa E. "heart diseases detection using naive Bayes algorithm", IJISET-international journal of innovative science. Eng Technol. 2015;2:441–4.

Course Outcomes

C318.1	Ability to understand and develop machine learning concepts
C318.2	Implement pytest to get the results
C318.3	Develop and deploy prediction system using logistic regression

CO – PO, PSO Mapping

CO	PO1	PO2	P03	PO4	P05	P06	PO7	PO8	PO9	PO10	PO11	PO12	PS01	PSO2	PS03
C318.1	3	3	3	3	3	2	2	2	2	3	3	3	3	3	3
C318.2	3	3	3	2	3	2	3	3	3	3	3	2	3	3	3
C318.3	3	3	2	3	3	3	2	3	2	3	3	2	2	3	3
C318	3	3	2.6	2.6	3	2.3	2.3	2.6	2.3	3	3	2.3	2.6	3	3

Justification:

C318.1	Acquired knowledge of machine learning
C318.2	Understood the functionalities of pytest
C318.3	Learnt to implement logistic regression