IBM PROJECT

Visualizing And Predicting Heart Diseases With an Interactive dashboard

Team Id : PNT2022TMID30634

Team Leader : K.Indhubala

Team Member 1 : A.Rasiga

Team Member 2 : A.Akalya

Team Member 3: B.Akila

ABSTRACT :

Heart attack disease is one of the leading causes of the death worldwide. In today‟s common modern life, deaths due to the heart disease had become one of major issues, that roughly one person lost his or her life per minute due to heart illness. Predicting the occurrence of disease at early stages is a major challenge nowadays. Machine learning when implemented in health care is capable of early and accurate detection of disease. In this work, the arising situations of Heart Disease illness are calculated. Datasets used have attributes of medical parameters. The datasets are been processed in python using ML Algorithm i.e., Decision Tree Classifier. This technique uses the past old patient records for getting prediction of new one at early stages preventing the loss of lives. In this work, reliable heart disease prediction system is implemented using strong Machine Learning algorithm which is the Random Forest algorithm. Which read patient record data set in the form of CSV file. After accessing dataset the operation is performed and effective heart attack level is produced. Advantages of proposed system are High performance and accuracy rate and it is very flexible and high rates of success are achieved.

TABLE OF CONTENTS

LIST OF FIGURES

FIGURE NO	FIGURE NAME	PAGE NO
5.1	System Architecture	14
5.2	Data Flow Diagram	16
5.5	Use Case Diagram	18
5.6	Class Diagram	19
5.7	Sequence Diagram	20
5.8	Activity Diagram	21
A3	Screen short	73

LIST OF ABBREVATIONS

ASF	Apache Software Foundation
DNS	Domain Name Server
J2EE	Java 2 Enterprises Edition
JDBC	Java Database Connectivity
JNI	Java Native Interfaces
JSP	Java Server Pages
JVM	Java Virtual Machine
MSCS	Microsoft Cluster Server
MVC	Model View Controller

RMI	Remote Method Inovation
URL	Uniform Resource Locator

CHAPTER – 1

1. INTRODUCTION

1.1. GENERAL:

Heart disease effects the functioning of the heart. World Health Organization had made a survey and made a conclusion that 10 million people are affected with heart disease and lost their lives. The problem that the Healthcare industry faces in today‟s life is early prediction of disease after a person is affected. Records or data of medical history is very large and the data in real world might be incomplete and inconsistent. In past predicting the disease effectively and treatment to patients might not be possible for every patient at early stages under these circumstances.

Many scientists tried to build a model which is capable of predicting the heart disease in the early stage, but they are not able to build a perfect model. Every proposed system has disadvantages in its own way. In the

existing system, Shen et al. had initially, proposed a system which is based on self applied questionnaire. In this system the user need to enter all the symptoms which he is suffering from , based on that the result is predicted. This study is based on the analysis data collected in SAQ.

Chen et al. came up with an idea to predict heart disease. He used the technique of Vector Quantization which is one of the artificial intelligence techniques for classification and prediction purpose. Training of neural networks is performed using back propagation to evaluate the prediction system. In the testing phase approximately 80% accuracy is achieved on testing set. Practical use of data collected from previous records is time consuming. Low accuracy rate. So to overcome this we are implementing Random forest algorithm in order to achieve accurate results in less time. Machine learning is given a major priority in modern life in many applications and in healthcare sector. Prediction is one of area where machine learning plays a vital role, our topic is to predict heart disease by processing patient‟s dataset and a data of

Patients.

CHAPTER – 2

2. LITERATURE SURVEY

.

2.1 Diagnosis of heart disease using genetic algorithm based trained recurrent fuzzy neural networks:

Authors: Kaan Uyar and Ahmet İlhan

The World Health Organization (WHO) estimated one third of all global deaths reason as cardiovascular diseases in 2015. Some computational techniques were proposed for investigation of heart diseases. This study proposes a genetic algorithm (GA) based trained recurrent fuzzy neural networks (RFNN) to diagnosis of heart diseases. The University of California Irvine (UCI) Cleveland heart disease dataset is used in this study. Out of total 297 instances of patient data, 252 are used for training and 45 of them are chosen to be the testing. The results showed that 97.78% accuracy was obtained from testing set. In addition to the accuracy, root mean square error, the probability of the misclassification error, specificity, sensitivity, precision and F-score are calculated. The results were found to be satisfying based on comparison.

2.2 Heart Disease Prediction Using Data Mining Techniques:

Authors: Ashish Chhabbi, Lakhan Ahuja, Sahil Ahir and Y. K. Sharma

The healthcare industry collects large amounts of Healthcare data, but unfortunately not all the data are mined which is required for discovering hidden patterns and effective decision making. We propose efficient genetic algorithm with the back propagation technique approach for heart disease prediction. This paper has analyzed prediction systems for Heart disease using more number of input attributes. The System uses medical terms such as Gender, blood pressure, cholesterol like13 attributes to predict the likelihood of patient getting a Heart disease.

2.3 Human Heart Disease Prediction System using Data Mining Techniques:

Authors: Theresa Princy and R, J. Thomas

Nowadays, health disease are increasing day by day due to life style, hereditary. Especially, heart disease has become more common these days, i.e. life of people is at risk. Each individual has different values for Blood pressure, cholesterol and pulse rate. But according to medically proven results the normal values of Blood pressure is 120/90, cholesterol is and pulse rate is 72. This paper gives the survey about different

classification techniques used for predicting the risk level of each person based on age, gender, Blood pressure, cholesterol, pulse rate. The patient risk level is classified using datamining classification techniques such as Naïve Bayes, KNN, Decision Tree Algorithm, Neural Network. etc., Accuracy of the risk level is high when using more number of attributes.

2.4 Review on Heart Disease Prediction System using Data Mining Techniques:

Authors: Beant Kaur, Williamjeet Singh

Data mining is the computer based process of analyzing enormous sets of data and then extracting the meaning of the data. Data mining tools predict future trends, allowing business to make proactive, knowledge-driven decisions. Data mining tools can answer business questions that traditionally taken much time consuming to resolve. The huge amounts of data generated for prediction of heart disease are too complex and voluminous to be processed and analyzed by traditional methods. Data mining provides the methodology and technology to transform these mounds of data into useful information for decision making. By using data mining techniques it takes less time for the prediction of the disease with more accuracy. In this paper we survey different papers in which one or more algorithms of data mining used for the prediction of heart disease. Result from using neural networks is nearly 100% in one paper. So that the prediction by using data mining algorithm given efficient

results. Applying data mining techniques to heart disease treatment data can provide as reliable performance as that achieved in diagnosing heart disease.

2.5Prediction of heart disease using decision tree a data mining technique:

Authors: Kirmani, M.M., Ansarullah, S.I.:

In today's modern world cardiovascular disease is the most lethal one. This disease attacks a person so instantly that it hardly gets any time to get treated with. So diagnosing patients correctly on timely basis is the most challenging task for the medical fraternity. Poor clinical decisions may end to patient death and which cannot be afforded by the hospital as it loses its reputation. In this paper using a data mining technique Decision Tree is used an attempt is made to assist in the diagnosis of the disease, Keeping in view the goal of this study to predict heart disease using classification techniques, I have used a supervised machine learning algorithms i.e., Decision Tree.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.

CHAPTER – 3

3 SYSTEM ANALYSIS

3.1 EXISTING SYSTEM:

- The existing system which is based on self applied questionnaire. In this system the user need to enter all the symptoms which he is suffering from , based on that the result is predicted. This study is based on the analysis data collected in SAQ.

- The technique of Vector Quantization which is one of the artificial intelligence techniques for classification and prediction purpose. Training of neural networks is performed using back propagation to evaluate the prediction system. In the testing phase approximately 80% accuracy is achieved on testing set. Practical use of data collected from previous records is time consuming

3.2 DISADVANTAGES OF EXISTING SYSTEM:

- Low accuracy rate.

- Time consuming process.

- 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.

- Most of these studies are theoretical analysis at the macro level and there is a lack of quantitative investigations**.**

3.4 PROPOSED SYSTEM:

- To overcome this we are implementing Decision Tree Classifier in order to achieve accurate results in less time. Machine learning is given a major priority in modern life in many applications and in healthcare sector. Prediction is one of area where machine learning plays a vital role, our topic is to predict heart disease by processing patient‘s dataset and a data of patients i.e., user of whom we need to predict the chances of occurrence of a heart disease.

- Our aim is to build an application of heart disease prediction system using robust Machine Learning algorithm which is Decision Tree Classifier. A CSV file is given as input. After the successful completion of operation the result is predicted and displayed.

3.5 ADVANTAGES OF PROPOSED SYSTEM:

- The advantages of proposed model are High performance and accuracy rate.
- It is very flexible and high rates of success are achieved.
- The application when implemented using Decision Tree Classifier has more accuracy rate when compared to other algorithms. In this system, we achieved around 96%.

CHAPTER – 4

4. SYSTEM SPECIFICATION

4.1 HARDWARE REQUIREMENTS

- System : Pentium IV 2.4 GHz.
- Hard Disk : 500 GB.
- Floppy Drive : 1.44 Mb.
- Monitor : 15 VGA Colour.
- Mouse : Logitech.
- Ram : 8 Gb.

4.2 SOFTWARE REQUIREMENTS

- Operating system: Windows 10(64-BIT System).
- Coding Language: Python.
- Software : Visual Studio Code.

CHAPTER – 5

5. SYSTEM DESIGN

5.1 SYSTEM ARCHITECTURE

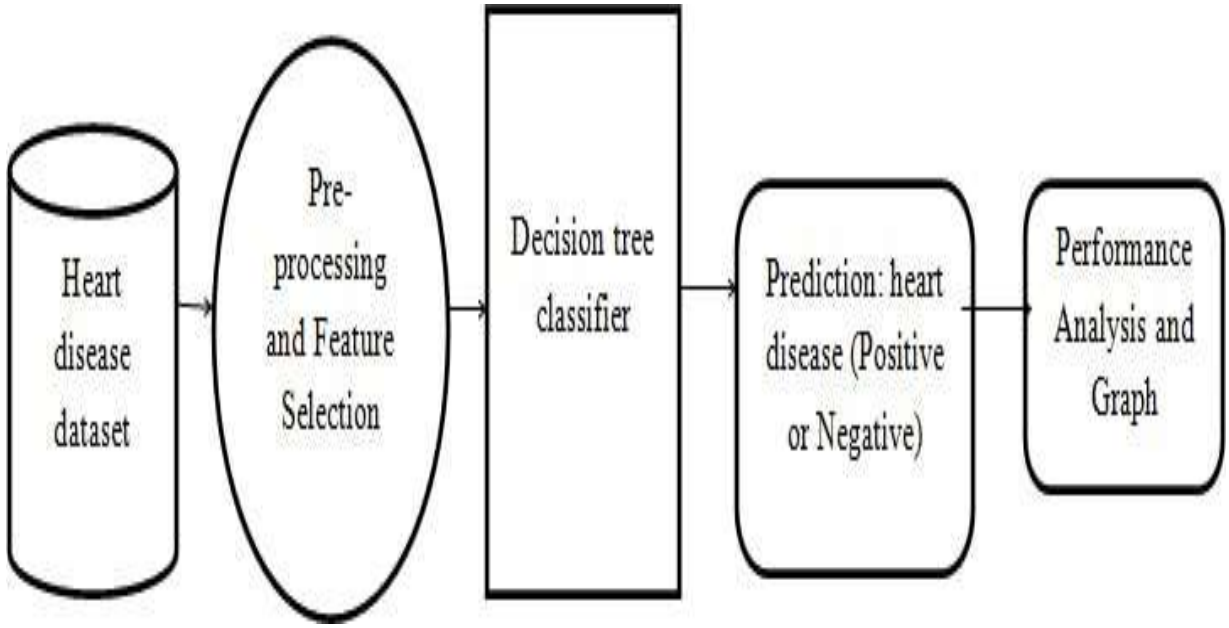


Figure:5.1 System Architecture

5.2 DATA FLOW DIAGRAM

1. The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of input data to the system, various processing carried out on this data, and the output data is generated by this system.
2. The data flow diagram (DFD) is one of the most important modeling tools. It is used to model the system components. These components are the system process, the data used by the process, an external entity that interacts with the system and the information flows in the system.
3. DFD shows how the information moves through the system and how it is modified by a series of transformations. It is a graphical technique that depicts information flow and the transformations that are applied as data moves from input to output.
4. DFD is also known as bubble chart. A DFD may be used to represent a system at any level of abstraction. DFD may be partitioned into levels that represent increasing information flow and functional detail.

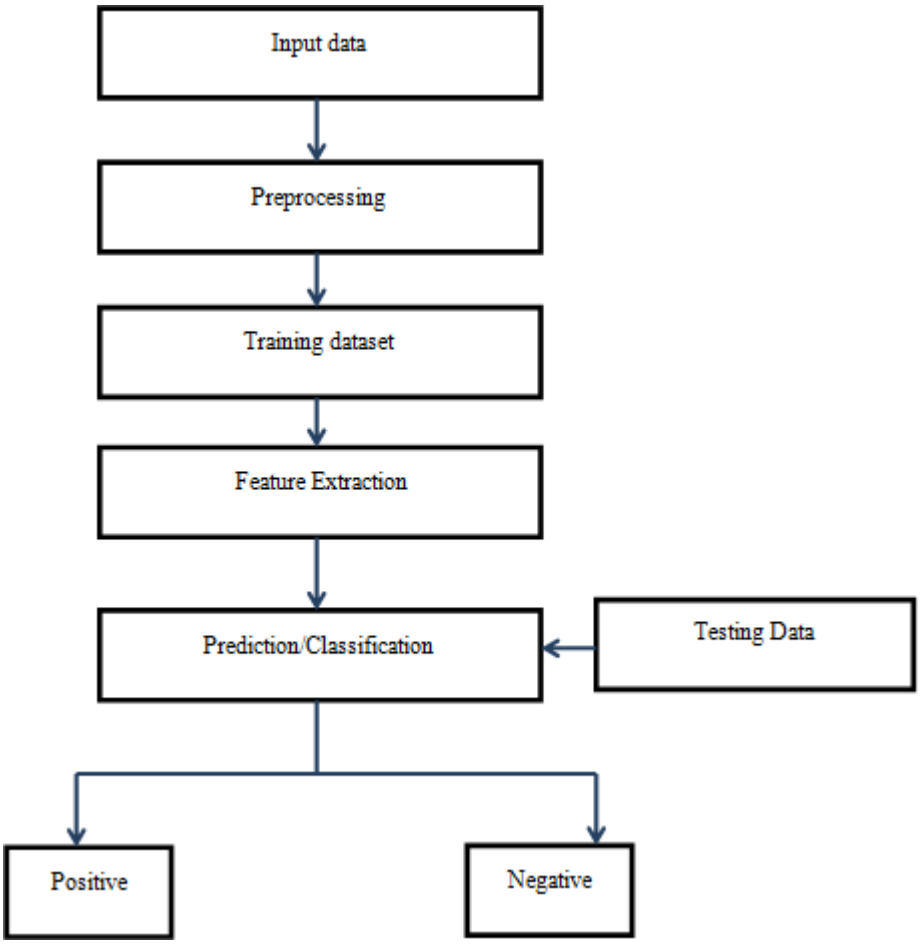


Figure:5.2 Data Flow Diagram

5.3 UML DIAGRAMS

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

The goal is for UML to become a common language for creating models of object oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems.

The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems.

The UML is a very important part of developing objects oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

5.4 GOALS

The Primary goals in the design of the UML are as follows:

1. Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.
2. Provide extendibility and specialization mechanisms to extend the core concepts.
3. Be independent of particular programming languages and development process.

5.5 USE CASE DIAGRAM

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases.

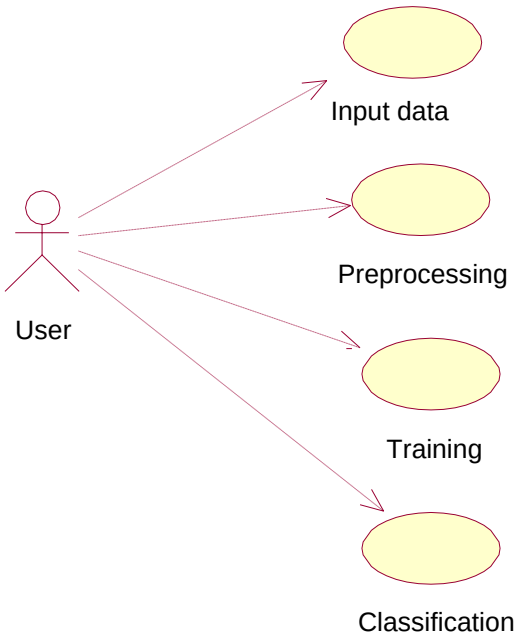


Figure:5.5 Use Case Diagram

5.6 CLASS DIAGRAM

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes,

operations (or methods), and the relationships among the classes. It explains which class contains information.

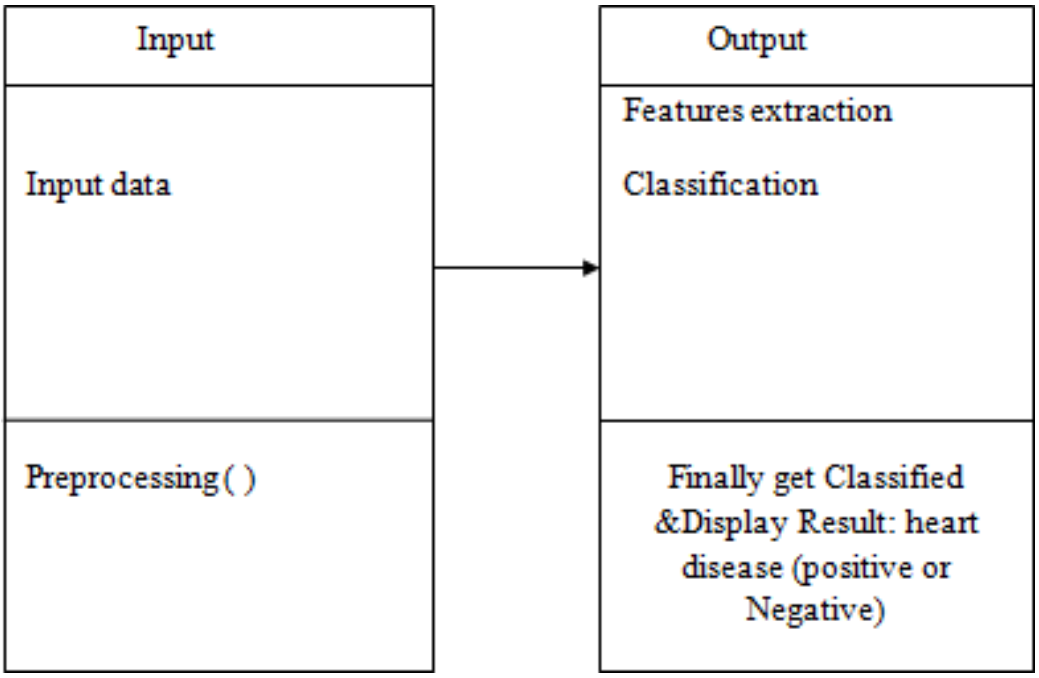


Figure:5.6 Class Diagram

5.7 SEQUENCE DIAGRAM

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

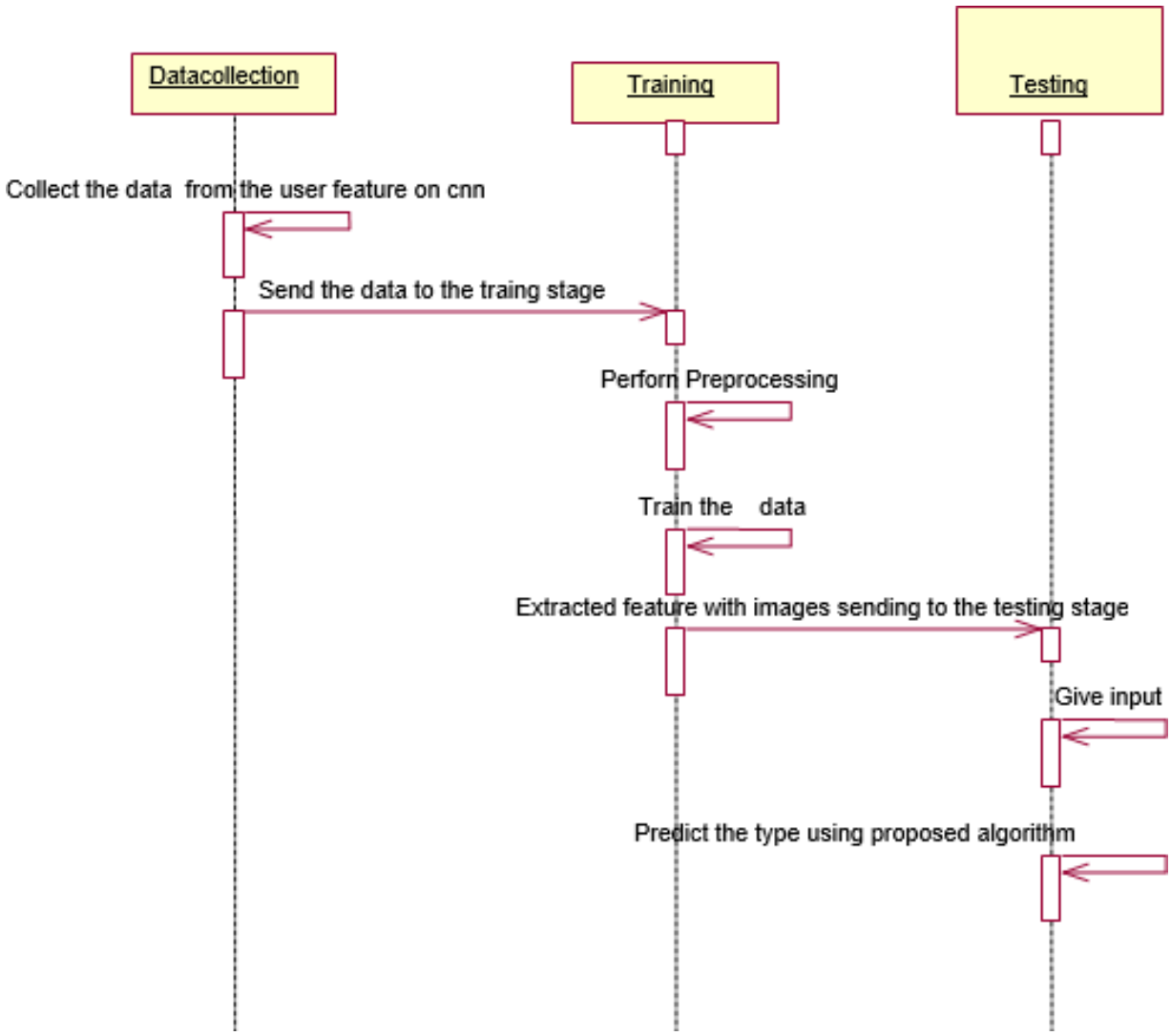


Figure:5.7 Sequence Diagram

5.8 ACTIVITY DIAGRAM

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

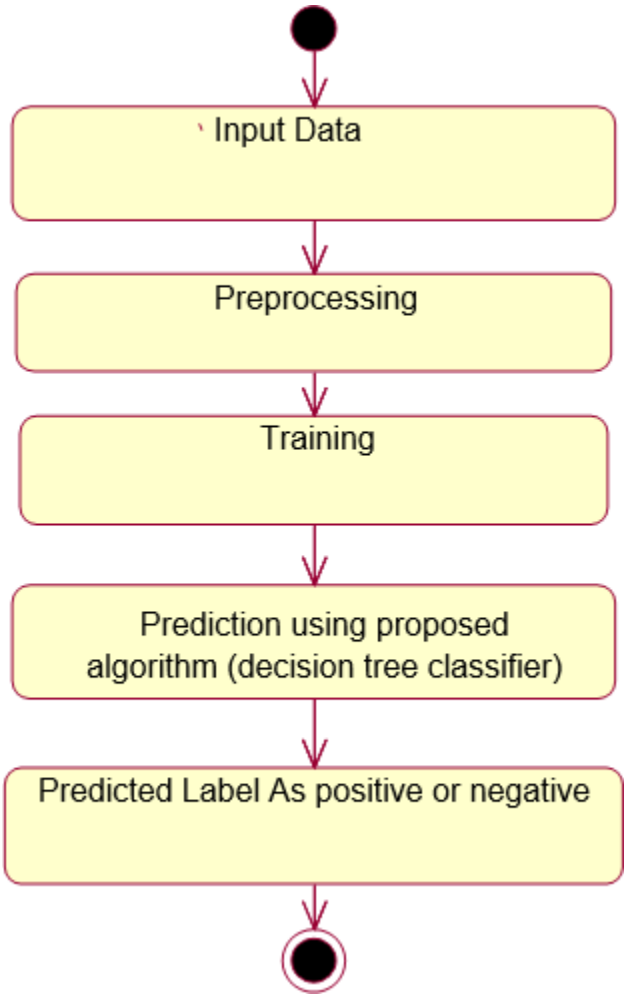


Figure:5.8 Activity Diagram

CHAPTER – 6

6. MODULE DESCRIPTION

6.1 MODULES

Our project modules are as follows.

- Data Collection
- Dataset
- Data Preparation
- Model Selection
- Analyze and Prediction
- Accuracy on test set
- Saving the Trained Model

6.2 MODULE DESCRIPTION

6.2.1 Data Collection:

This is the first real step towards the real development of a machine learning model, collecting data. This is a critical step that will cascade in how good the model will be, the more and better data that we get, the better our model will perform.There are several techniques to collect the data, like web scraping, manual interventions and etc.Heart Disease dataset taken from Kaggle (https://www.kaggle.com/ronitf/heart-disease-uci

6.2.2 Dataset

The dataset consists of 303 individual data. There are 14 columns in the dataset, which are described below.

1. ***Age***: displays the age of the individual.
2. ***Sex***: displays the gender of the individual using the following format :
   1=male
   0 = female
3. ***Chest-pain type(*cp*)***: displays the type of chest-pain experienced by the individual using the following format :
   1=typicalangina
   2=atypical angina
   3=non—anginalpain
   4 = asymptotic
4. ***Resting Blood Pressure(*trestbps*)***: displays the resting blood pressure value of an individual in mmHg (unit)
5. ***Serum Cholestrol(*chol*)***: displays the serum cholesterol in mg/dl (unit)
6. ***Fasting Blood Sugar(*fbs*)***: compares the fasting blood sugar value of an individual with 120mg/dl.If fasting blood sugar > 120mg/dl then : 1 (true)else : 0 (false)

7. ***Resting ECG (*restecg*)***: displays resting electrocardiographic results0 = normal1 = having ST-T wave abnormality2 = left ventricular hyperthrophy
8. ***Max heart rate achieved*** : displays the max heart rate achieved by an individual.
9. ***Exercise induced angina*** :1 = yes0 = no
10. ***ST depression induced by exercise relative to rest***: displays the value which is an integer or float.
11. ***Peak exercise ST segment*** :1 = upsloping2 = flat3 = downsloping
12. ***Number of major vessels (0–3) colored by flourosopy*** : displays the value as integer or float.
13. ***Thal*** : displays the thalassemia :3 = normal6 = fixed defect7 = reversible defect
14. ***Diagnosis of heart disease*** : Displays whether the individual is suffering from heart disease or not :0 = absence1 = present.

6.2.3 Data Preparation:

Wrangle data and prepare it for training. Clean that which may require it (remove duplicates, correct errors, deal with missing values, normalization, data type conversions, etc.)

Randomize data, which erases the effects of the particular order in which we collected and/or otherwise prepared our data

Visualize data to help detect relevant relationships between variables or class imbalances (bias alert!), or perform other exploratory analysis

Split into training and evaluation sets

6.2.4 Model Selection:

We used Decision Tree Classifier machine learning algorithm, We got a accuracy of 96.7% on test set so we implemented this algorithm.

6.2.5 Decision Tree Classification Algorithm

Decision Tree is a Supervised learning technique that can be used for both classification and Regression problems, but mostly it is preferred for solving Classification problems. It is a tree-structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome.

In a Decision tree, there are two nodes, which are the Decision Node and Leaf Node. Decision nodes are used to make any decision and have multiple branches, whereas Leaf nodes are the output of those decisions and do not contain any further branches.

The decisions or the test are performed on the basis of features of the given dataset.

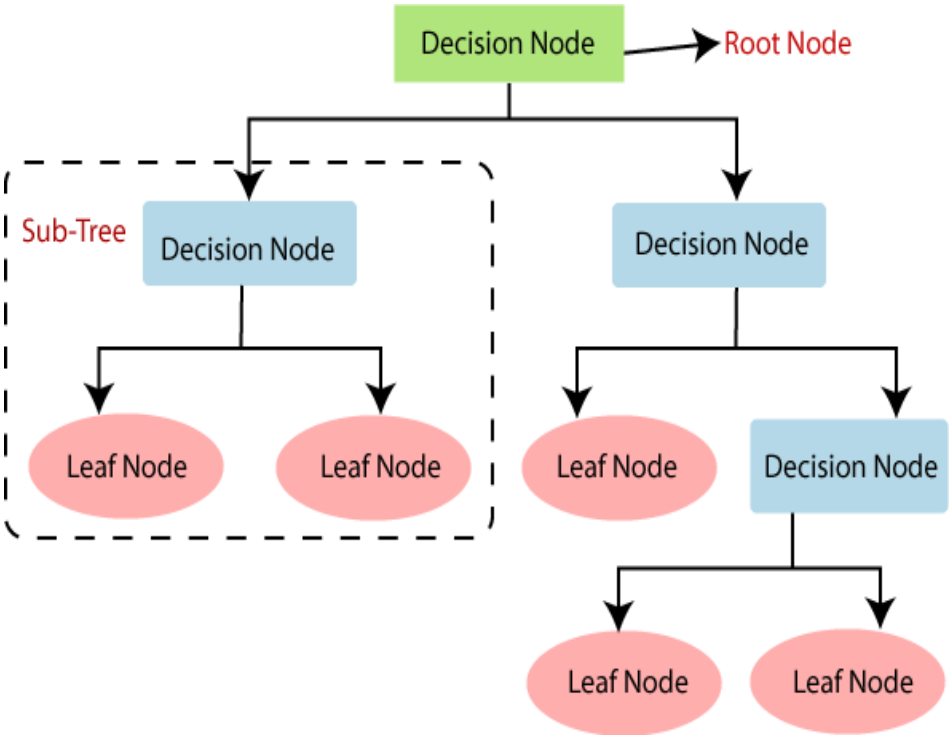
It is a graphical representation for getting all the possible solutions to a problem/decision based on given conditions.

It is called a decision tree because, similar to a tree, it starts with the root node, which expands on further branches and constructs a tree-like structure.

In order to build a tree, we use the CART algorithm**,** which stands for Classification and Regression Tree algorithm**.**

A decision tree simply asks a question, and based on the answer (Yes/No), it further split the tree into subtrees.

Below diagram explains the general structure of a decision tree:



6.2.6 Decision Tree Terminologies

Root Node: Root node is from where the decision tree starts. It represents the entire dataset, which further gets divided into two or more homogeneous sets.

Leaf Node: Leaf nodes are the final output node, and the tree cannot be segregated further after getting a leaf node.

Splitting: Splitting is the process of dividing the decision node/root node into sub-nodes according to the given conditions.

Branch/Sub Tree: A tree formed by splitting the tree.

Pruning: Pruning is the process of removing the unwanted branches from the tree.

Parent/Child node: The root node of the tree is called the parent node, and other nodes are called the child nodes.

6.2.7 Analyze and Prediction:

In the actual dataset, we chose only 7 features :

1. **Age**: displays the age of the individual.
2. **Sex**: displays the gender of the individual using the following format :1 = male0 = female
3. **Chest-pain type(cp)**: displays the type of chest-pain experienced by the individual using the following format :1 = typical angina2 = atypical angina3 = non — anginal pain4 = asymptotic
4. **Resting Blood Pressure(trestbps)**: displays the resting blood pressure value of an individual in mmHg (unit)
5. **Serum Cholestrol(chol)**: displays the serum cholesterol in mg/dl (unit)
6. **Fasting Blood Sugar(fbs)**: compares the fasting blood sugar value of an individual with 120mg/dl.If fasting blood sugar > 120mg/dl then : 1 (true) else : 0 (false)

7. **Resting ECG (restecg)**: displays resting electrocardiographic results0 = normal
   1 = having ST-T wave abnormality2 = left ventricular hyperthrophy
8. **Max heart rate achieved** : displays the max heart rate achieved by an individual.
9. **Exercise induced angina** :1 = yes0 = no
10. **ST depression induced by exercise relative to rest**: displays the value which is an integer or float.
11. **Peak exercise ST segment** : 1 = upsloping 2 = flat 3 = downsloping

6.2.8 Accuracy on test set:

We got an accuracy of 96.7% on test set.

6.2.9 Saving the Trained Model:

Once you‘re confident enough to take your trained and tested model into the production-ready environment, the first step is to save it into a .h5 or . pkl file using a library like `pickle` .

Make sure you have `pickle` installed in your environment.

Next, let‘s import the module and dump the model into **.** pkl file

CHAPTER – 7

7. SYSTEM TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

7.1 TYPES OF TESTS

7.1.1 UNIT TESTING

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business

process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

7.1.2 INTEGRATION TESTING

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

7.1.3 FUNCTIONAL TEST

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

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

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

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

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

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

7.1.4 SYSTEM TEST

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

7.1.5 WHITE BOX TESTING

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

7.1.6 BLACK BOX TESTING

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

7.1.7 UNIT TESTING

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

TEST STRATEGY AND APPROACH

Field testing will be performed manually and functional tests will be written in detail.

TEST OBJECTIVES

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

FEATURES TO BE TESTED

Verify that the entries are of the correct format

No duplicate entries should be allowed

All links should take the user to the correct page.

7.2 INTEGRATION TESTING

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects. The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

7.3 ACCEPTANCE TESTING

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

CHAPTER – 8

8. APPLICATION AREAS

This application can extended by updating some features like, if the user is effected with heart disease all his family members will be notified with a message in early. And also the information should be passed to the nearest hospital. Another feature is there should be online doctor consultation with the nearest doctor available.In this regard, it is important to note that, DL applications using various efficient algorithms are utilized not only in disease prediction and diagnosis but also in the field of radiology, bioinformatics and medical imaging diagnosis etc...

CHAPTER – 9

9. SYSTEM STUDY

9.1 FEASIBILITY STUDY

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

Three key considerations involved in the feasibility analysis are

Economical Feasibility

Technical Feasibility

Social Feasibility

9.1.1 ECONOMICAL FEASIBILITY

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

technologies used are freely available. Only the customized products had to be purchased.

9.1.2 TECHNICAL FEASIBILITY

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

9.1.3 SOCIAL FEASIBILITY

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

CHAPTER –10

10. CONCLUSION & FUTURE ENCHANCEMENT

10.1 CONCLUSION

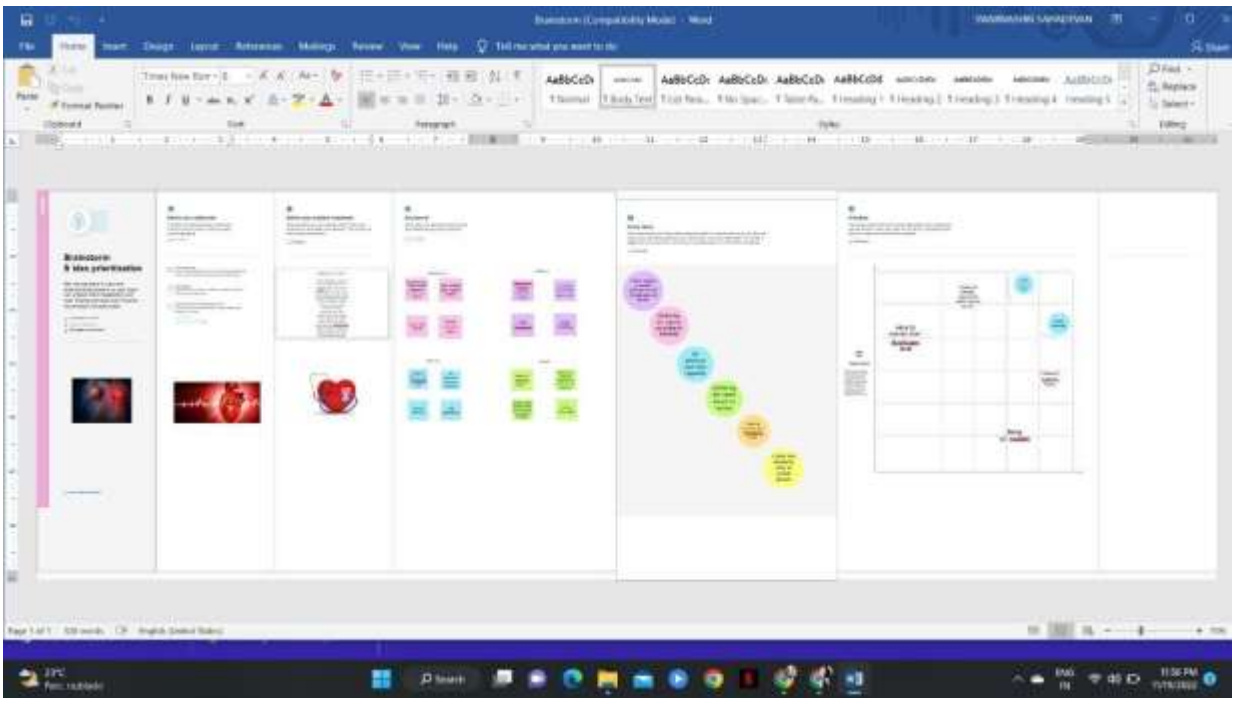
The primary objective of the proposed algorithm is to minimize MakeSpan and improve fitness function. Improving the Load Balancing process through Task Scheduling can result in efficient utilization of cloud resources. The objective of this proposed work was to provide an enhanced Load Balancing algorithm. Results proved that our algorithm reduces Makespan and provide efficient resource utilization of compared to existing Dynamic LBA (Load Balancing Algorithm). It also shows that the proposed algorithm can function in a dynamic cloud environment where user requests arrive in random order and where there are many changes in the length of the user requests. The algorithm is also able to handle large size requests compared to the existing approach.

10.2 FUTURE ENHANCEMENTS

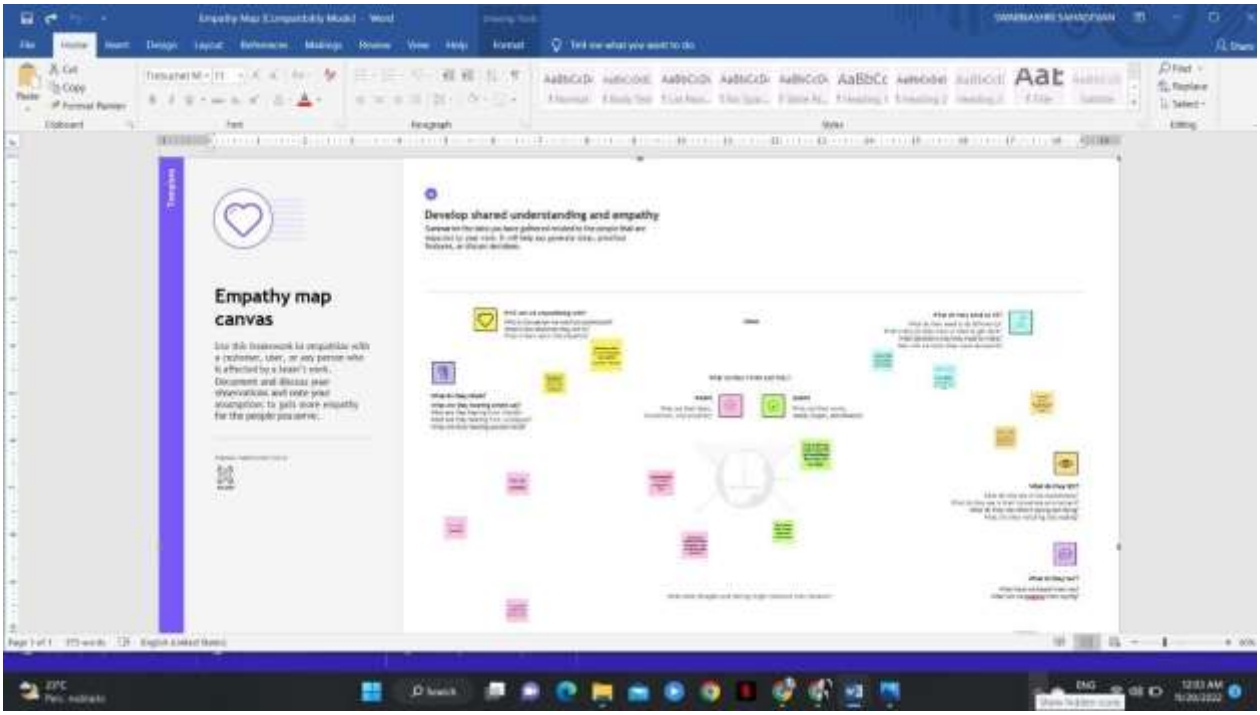
In the future, various other metrics like throughput, average time, resource utilization, waiting time, etc. can be considered. In the future, authors will work to optimize the cloud resources further and enhance cloud-based application performance, such as considering more SLA

(Service Level Agreements) parameters. For example, the algorithm will be tested based on the number of violations and the migration count for better performance. Also, the algorithm will be comprehensively compared to other existing algorithms in the literature.

BRAINSTORM



EMPATHY MAP



APPENDIX 1

A1. SOFTWARE ENVIRONMENT

A1.2.1.1 Python

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

- **Python is Interpreted** − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
- **Python is Interactive** − You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.
- **Python is Object-Oriented** − Python supports Object-Oriented style or technique of programming that encapsulates code within objects.
- **Python is a Beginner's Language** − Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

A1.2.1.2 History of Python

Python was developed by Guido van Rossum in the late eighties and early nineties at the National Research Institute for Mathematics and Computer Science in the Netherlands.

Python is derived from many other languages, including ABC, Modula-3, C, C++, Algol-68, SmallTalk, and Unix shell and other scripting languages.

Python is copyrighted. Like Perl, Python source code is now available under the GNU General Public License (GPL).

Python is now maintained by a core development team at the institute, although Guido van Rossum still holds a vital role in directing its progress.

A1.2.1.3 Python Features

Python's features include −

- **Easy-to-learn** − Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.
- **Easy-to-read** − Python code is more clearly defined and visible to the eyes.
- **Easy-to-maintain** − Python's source code is fairly easy-to-maintain.
- **A broad standard library** − Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.
- **Interactive Mode** − Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.

- **Portable** − Python can run on a wide variety of hardware platforms and has the same interface on all platforms.
- **Extendable** − You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.
- **Databases** − Python provides interfaces to all major commercial databases.
- **GUI Programming** − Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.
- **Scalable** − Python provides a better structure and support for large programs than shell scripting.

Apart from the above-mentioned features, Python has a big list of good features, few are listed below −

- It supports functional and structured programming methods as well as OOP.
- It can be used as a scripting language or can be compiled to byte-code for building large applications.

- It provides very high-level dynamic data types and supports dynamic type checking.
- It supports automatic garbage collection.
- It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java.

Python is available on a wide variety of platforms including Linux and Mac OS X. Let's understand how to set up our Python environment.

A1.2.1.3.1 Getting Python

The most up-to-date and current source code, binaries, documentation, news, etc., is available on the official website of Python https://www.python.org.

Windows Installation

Here are the steps to install Python on Windows machine.

- Open a Web browser and go to https://www.python.org/downloads/.
- Follow the link for the Windows installer python-XYZ.msifile where XYZ is the version you need to install.

- To use this installer python-XYZ.msi, the Windows system must support Microsoft Installer 2.0. Save the installer file to your local machine and then run it to find out if your machine supports MSI.
- Run the downloaded file. This brings up the Python install wizard, which is really easy to use. Just accept the default settings, wait until the install is finished, and you are done.

The Python language has many similarities to Perl, C, and Java. However, there are some definite differences between the languages.

First Python Program

Let us execute programs in different modes of programming.

Interactive Mode Programming

Invoking the interpreter without passing a script file as a parameter brings up the following prompt –

```
$ python

Python2.4.3(#1,Nov112010,13:34:43)

[GCC 4.1.220080704(RedHat4.1.2-48)] on linux2

Type"help","copyright","credits"or"license"for more information.
```

Type the following text at the Python prompt and press the Enter –

```
print"Hello, Python!"
```

If you are running new version of Python, then you would need to use print statement with parenthesis as in **print ("Hello, Python!");**. However in Python version 2.4.3, this produces the following result −

Script Mode Programming

Hello, Python!

Invoking the interpreter with a script parameter begins execution of the script and continues until the script is finished. When the script is finished, the interpreter is no longer active.

Let us write a simple Python program in a script. Python files have extension **.py**. Type the following source code in a test.py file –

print"Hello, Python!"

We assume that you have Python interpreter set in PATH variable.
Now, try to run this program as follows –

$ python test.py

This produces the following result –

Hello, Python!

Flask Framework

Flask is a web application framework written in Python. Armin Ronacher, who leads an international group of Python enthusiasts named

Pocco, develops it. Flask is based on Werkzeug WSGI toolkit and Jinja2 template engine. Both are Pocco projects.

Http protocol is the foundation of data communication in world wide web. Different methods of data retrieval from specified URL are defined in this protocol.

The following table summarizes different http methods –

Sr.No	Methods & Description
1	**GET** Sends data in unencrypted form to the server. Most common method.
2	**HEAD** Same as GET, but without response body
3	**POST** Used to send HTML form data to server. Data received by POST method is not cached by server.
4	**PUT** Replaces all current representations of the target resource with the

uploaded content.

5 **DELETE**

Removes all current representations of the target resource given by a URL

By default, the Flask route responds to the **GET** requests. However, this preference can be altered by providing methods argument to **route()** decorator.

In order to demonstrate the use of **POST** method in URL routing, first let us create an HTML form and use the **POST** method to send form data to a URL.

Save the following script as login.html

```
<html>

<body>

<formaction="http://localhost:5000/login"method="post">

<p>Enter Name:</p>

<p><inputtype="text"name="nm"/></p>

<p><inputtype="submit"value="submit"/></p>
```

```
</form>
</body>
</html>
```

Now enter the following script in Python shell.

```
from flask importFlask, redirect,url_for, request
app=Flask(__name__)
@app.route('/success/<name>')
def success(name):
return'welcome %s'% name
@app.route('/login',methods=['POST','GET'])
def login():
ifrequest.method=='POST':
user=request.form['nm']
return redirect(url_for('success',name= user))
else:
user=request.args.get('nm')
return redirect(url_for('success',name= user))
if__name__=='__main__':
```

app.run(debug =True)

After the development server starts running, open login.html in the browser, enter name in the text field and click Submit.

Form data is POSTed to the URL in action clause of form tag.

http://localhost/login is mapped to the **login()** function. Since the server has received data by **POST** method, value of _nm‘ parameter obtained from the form data is obtained by −

user = request.form['nm']

It is passed to **‘/success’** URL as variable part. The browser displays a **welcome** message in the window.

Change the method parameter to **‘GET’** in **login.html** and open it again in the browser. The data received on server is by the **GET** method. The value of _nm‘ parameter is now obtained by –

User = request.args.get(_nm‘)

Here, **args** is dictionary object containing a list of pairs of form parameter and its corresponding value. The value corresponding to _nm‘ parameter is passed on to _/success‘ URL as before.

Python is a popular programming language. It was created in 1991 by Guido van Rossum.

It is used for:

- web development (server-side),
- software development,
- mathematics,
- system scripting.

A1.2.1.4 What can Python do?

- Python can be used on a server to create web applications.
- Python can be used alongside software to create workflows.
- Python can connect to database systems. It can also read and modify files.
- Python can be used to handle big data and perform complex mathematics.
- Python can be used for rapid prototyping, or for production-ready software development.

A1.2.1.4.1Why Python?

- Python works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc).
- Python has a simple syntax similar to the English language.
- Python has syntax that allows developers to write programs with fewer lines than some other programming languages.

- Python runs on an interpreter system, meaning that code can be executed as soon as it is written. This means that prototyping can be very quick.
- Python can be treated in a procedural way, an object-orientated way or a functional way.

Good to know

- The most recent major version of Python is Python 3, which we shall be using in this tutorial. However, Python 2, although not being updated with anything other than security updates, is still quite popular.
- In this tutorial Python will be written in a text editor. It is possible to write Python in an Integrated Development Environment, such as Thonny, Pycharm, Netbeans or Eclipse which are particularly useful when managing larger collections of Python files.

Python Syntax compared to other programming languages

- Python was designed to for readability, and has some similarities to the English language with influence from mathematics.
- Python uses new lines to complete a command, as opposed to other programming languages which often use semicolons or parentheses.
- Python relies on indentation, using whitespace, to define scope; such as the scope of loops, functions and classes. Other programming languages often use curly-brackets for this purpose.

A1.2.5 Python Install

Many PCs and Macs will have python already installed.

To check if you have python installed on a Windows PC, search in the start bar for Python or run the following on the Command Line (cmd.exe):

C:\Users\Your Name>python --version

To check if you have python installed on a Linux or Mac, then on linux open the command line or on Mac open the Terminal and type:

python --version

If you find that you do not have python installed on your computer, then you can download it for free from the following website: https://www.python.org/

A1.2.6 Python Quickstart

Python is an interpreted programming language, this means that as a developer you write Python (.py) files in a text editor and then put those files into the python interpreter to be executed.

The way to run a python file is like this on the command line:

C:\Users\Your Name>python helloworld.py

Where "helloworld.py" is the name of your python file.

Let's write our first Python file, called helloworld.py, which can be done in any text editor.

helloworld.py

```
print("Hello, World!")
```

Simple as that. Save your file. Open your command line, navigate to the directory where you saved your file, and run:

C:\Users\Your Name>python helloworld.py

The output should read:

Hello, World!

Congratulations, you have written and executed your first Python program.

A1.2.7 The Python Command Line

To test a short amount of code in python sometimes it is quickest and easiest not to write the code in a file. This is made possible because Python can be run as a command line itself.

Type the following on the Windows, Mac or Linux command line:

C:\Users\Your Name>python

From there you can write any python, including our hello world example from earlier in the tutorial:

C:\Users\Your Name>python

Python 3.6.4 (v3.6.4:d48eceb, Dec 19 2017, 06:04:45) [MSC v.1900 32 bit (Intel)] on win32

Type "help", "copyright", "credits" or "license" for more information.

>>> print("Hello, World!")

Which will write "Hello, World!" in the command line:

C:\Users\Your Name>python

Python 3.6.4 (v3.6.4:d48eceb, Dec 19 2017, 06:04:45) [MSC v.1900 32 bit (Intel)] on win32

Type "help", "copyright", "credits" or "license" for more information.

>>> print("Hello, World!")Hello, World!

Whenever you are done in the python command line, you can simply type the following to quit the python command line interface:

exit()

Execute Python Syntax

As we learned in the previous page, Python syntax can be executed by writing directly in the Command Line:

>>> print("Hello, World!")Hello, World!

Or by creating a python file on the server, using the .py file extension, and running it in the Command Line:

C:\Users*Your Name*>python myfile.py

Python Indentations

Where in other programming languages the indentation in code is for readability only, in Python the indentation is very important.

Python uses indentation to indicate a block of code.

Example

```
if 5 > 2:
  print("Five is greater than two!")
```

Python will give you an error if you skip the indentation:

Example

if 5 > 2:

print("Five is greater than two!")

Comments

Python has commenting capability for the purpose of in-code documentation.

Comments start with a #, and Python will render the rest of the line as a comment:

Example

Comments in Python:

#This is a comment.print("Hello, World!")

Docstrings

Python also has extended documentation capability, called docstrings.

Docstrings can be one line, or multiline.

Python uses triple quotes at the beginning and end of the docstring:

Example

Docstrings are also comments:

"""Thisis a multiline docstring."""print("Hello, World!")

APPENDIX 2

A2.SOURCE CODE

import numpy as np

```
import pandas as pd
from flask import Flask, request, jsonify, render_template, redirect, flash, send_file
from sklearn.preprocessing import MinMaxScaler
from werkzeug.utils import secure_filename
import pickle
import numpy as np
import pandas as pd
from flask import Flask, request, jsonify, render_template, redirect, flash, send_file
from sklearn.preprocessing import MinMaxScaler
from werkzeug.utils import secure_filename
import pickle
import numpy as np
import pandas as pd
from sklearn.tree import DecisionTreeClassifier
from sklearn.svm import SVC
```

```
app = Flask(__name__) #Initialize the flask App
hart = pickle.load(open('heartd.pkl','rb'))
@app.route('/')
@app.route('/index')
def index():
    return render_template('index.html')
@app.route('/chart')
def chart():
    return render_template('chart.html')
#@app.route('/future')
#def future():
#    return render_template('future.html')
@app.route('/login')
def login():
    return render_template('login.html')
@app.route('/upload')
```

```
def upload():
    return render_template('upload.html')
@app.route('/preview',methods=["POST"])
def preview():
    if request.method == 'POST':
        dataset = request.files['datasetfile']
        df = pd.read_csv(dataset,encoding = 'unicode_escape')
        df.set_index('Id', inplace=True)
        return render_template("preview.html",df_view = df)
#@app.route('/home')
#def home():
 #   return render_template('home.html')
@app.route('/prediction', methods = ['GET', 'POST'])
def prediction():
    return render_template('prediction.html')
#@app.route('/upload')
#def upload_file():
```

```
#   return render_template('BatchPredict.html')
@app.route('/predict',methods=['POST'])
def predict():
        int_feature = [x for x in request.form.values()]


        final_features = [np.array(int_feature)]
                result=heart.predict(final_features)
        if result == 1:
                        result = "Positive"
        else:
                result = 'Negative'
                return    render_template('prediction.html',    prediction_text=
result)
@app.route('/performance')
def performance():
        return render_template('performance.html')
    if__name__== "__main__":
```

```
    app.run(debug=True)
```

```
<!doctype html>
<html class="no-js" lang="zxx">


<head>
    <meta charset="utf-8">
    <meta http-equiv="x-ua-compatible" content="ie=edge">
    <title>login</title>
    <meta name="description" content="">
    <meta name="viewport" content="width=device-width, initial-
scale=1">
    <!-- <link rel="manifest" href="site.webmanifest"> -->
    <link rel="shortcut icon" type="image/x-icon"
href="img/favicon.png">
    <!-- Place favicon.ico in the root directory -->


    <!-- CSS here -->
      <link rel="stylesheet" href="../static/css/bootstrap.min.css">
    <link rel="stylesheet" href="../static/css/owl.carousel.min.css">
    <link rel="stylesheet" href="../static/css/magnific-popup.css">
    <link rel="stylesheet" href="../static/css/font-awesome.min.css">
    <link rel="stylesheet" href="../static/css/themify-icons.css">
```

```
    <link rel="stylesheet" href="../static/css/nice-select.css">
    <link rel="stylesheet" href="../static/css/flaticon.css">
    <link rel="stylesheet" href="../static/css/gijgo.css">
    <link rel="stylesheet" href="../static/css/animate.css">
    <link rel="stylesheet" href="../static/css/slicknav.css">
    <link rel="stylesheet" href="../static/css/style.css">
    <!-- <link rel="stylesheet" href="css/responsive.css"> -->
</head>

<body>
    <!--[if lte IE 9]>
            <p class="browserupgrade">You are using an
<strong>outdated</strong> browser. Please <a
href="https://browsehappy.com/">upgrade your browser</a> to improve
your experience and security.</p>
        <![endif]-->
    <!-- header-start -->
    <header>
        <div class="header-area ">
            <div id="sticky-header" class="main-header-area">
                <div class="container">
                    <div class="row align-items-center">
                        <div class="col-xl-3 col-lg-3">
```

```
                    <div class="logo-img">
                        <a href="index.html">
                            <img src="img/logo.png" alt="">
                        </a>
                    </div>
                </div>
                <div class="col-xl-9 col-lg-9">
                    <div class="menu_wrap d-none d-lg-block">
                        <div class="menu_wrap_inner d-flex align-items-
center justify-content-end">
                            <div class="main-menu">
                                <nav>
                                    <ul id="navigation">
                                        <li><a
href="{{url_for('index')}}">Home</a></li>
                                        <li><a
href="{{url_for('login')}}">Login</a></li>

        <li><a href="{{url_for('upload')}}">Upload</a></li>

                                    </ul>
                                </nav>
                            </div>
```

```
                    </div>
                </div>
            </div>
            <div class="col-12">
                <div class="mobile_menu d-block d-lg-none"></div>
            </div>
        </div>
    </div>
</div>
</header>
<!-- header-end -->

<!-- bradcam_area_start  -->
<div class="bradcam_area breadcam_bg_1">
    <div class="container">
        <div class="row">
            <div class="col-xl-12">
                <div class="bradcam_text">
                    <h3>Login</h3>
                </div>
            </div>
```

```
            </div>
          </div>
      </div>
      <!-- bradcam_area_end  -->

      <!-- ================ contact section start
================= -->

      <section>
      </br>
          </br>
          </br>
                <div class="row">
                    <div class="col-12">
 <center><h2>Login</h2></center>
                    </div>

                                        <head>
           <style>
</style>
          <div class="container">
             <div class="row">
       </style>
```

```
        <script>
addEventListener("load", function () {
    setTimeout(hideURLbar, 0);
}, false);

function hideURLbar() {
    window.scrollTo(0, 1);
}
function login(){
var uname = document.getElementById("uname").value;
var pwd = document.getElementById("pwd").value;

 if(uname == "admin" && pwd == "admin")
 {
   alert("Login Success!");

   window.location = "{{url_for('upload')}}";
   return false;
 }
 else
 {
 alert("Invalid Credentials!")
 }
```

```
        }

    </script>
        </head>
<body id="page-top">

 <!-- Portfolio Section -->
 <section class="page-section portfolio" id="portfolio">

     <br>
         <br>
     <!-- Portfolio Section Heading -->
     <!-- Icon Divider -->

     <!-- Portfolio Grid Items -->
     <div class="row">
       <!-- Portfolio Item 1 -->
       <div class="col-md-6 col-lg-4" style="margin-left:580px">
                <div class="control-group">
                            <!-- Username -->

                            <label class="control-label"
for="username"><b>Username</b></label>
```

```
                        <div class="controls">
                          <input type="text" id="uname" name="uname"
placeholder="" class="form-control">

                            </div>
                        </div>
                            </br>
                        <div class="control-group">
                          <!-- Password-->

                          <label class="control-label"
for="password"><b>Password</b></label>
                          <div class="controls">
                            <input type="password" id="pwd" name="pwd"
placeholder="" class="form-control">

                            </div>
                        </div>
                            <div class="col-md-6 col-lg-4" style="margin-
left:-120px">
                        <div class="control-group">
                          <!-- Button -->
                            <br>
```

```
                        <div class="controls">

                            <input type="button" class="btn btn-primary"
value="Login" style="margin-left: 180px" onclick="login()">

                        </div>
                    </div>
                        </div>
  </div>
  </div>
   </section>
    </body>
        </div>

       </section>
    <!-- ================ contact section end
================ -->
    </br>
        </br>
        </br>
        </br>
    <footer class="footer">
       <div class="footer_top">
```

```
        <div class="container">
            <div class="row">
                <div class="col-xl-4 col-md-6 col-lg-4 ">
                    <div class="footer_widget">
                                            </div>
                </div>
                <div class="col-xl-4 col-md-6 col-lg-4">

                </div>
                            </div>
        </div>
    </div>
    <div class="copy-right_text">
        <div class="container">
            <div class="row">
                <div class="bordered_1px "></div>
                <div class="col-xl-12">

                </div>
            </div>
        </div>
    </div>
</footer>
```

```
<!-- link that opens popup -->
<!-- form itself end-->
    <!-- form itself end -->
      <!-- JS here -->
   <script src="../static/js/vendor/modernizr-3.5.0.min.js"></script>
<script src="../static/js/vendor/jquery-1.12.4.min.js"></script>
<script src="../static/js/popper.min.js"></script>
<script src="../static/js/bootstrap.min.js"></script>
<script src="../static/js/owl.carousel.min.js"></script>
<script src="../static/js/isotope.pkgd.min.js"></script>
<script src="../static/js/ajax-form.js"></script>
<script src="../static/js/waypoints.min.js"></script>
<script src="../static/js/jquery.counterup.min.js"></script>
<script src="../static/js/imagesloaded.pkgd.min.js"></script>
<script src="../static/js/scrollIt.js"></script>
<script src="../static/js/jquery.scrollUp.min.js"></script>
<script src="../static/js/wow.min.js"></script>
<script src="../static/js/nice-select.min.js"></script>
<script src="../static/js/jquery.slicknav.min.js"></script>
<script src="../static/js/jquery.magnific-popup.min.js"></script>
<script src="../static/js/plugins.js"></script>
<script src="../static/js/gijgo.min.js"></script>
<!--contact js-->
```

```
    <script src="../static/js/contact.js"></script>
    <script src="../static/js/jquery.ajaxchimp.min.js"></script>
    <script src="../static/js/jquery.form.js"></script>
    <script src="../static/js/jquery.validate.min.js"></script>
    <script src="../static/js/mail-script.js"></script>
    <script src="../static/js/main.js"></script>
      <script>
        $('#datepicker').datepicker({
          iconsLibrary: 'fontawesome',
          icons: {
           rightIcon: '<span class="fa fa-caret-down"></span>'
         }
        });
        $('#datepicker2').datepicker({
          iconsLibrary: 'fontawesome',
          icons: {
           rightIcon: '<span class="fa fa-caret-down"></span>'
         }
          });
      </script>
    </body>
    </html>
================= -->
```

```
</br>
    </br>
    </br>
    </br>
<footer class="footer">
   <div class="footer_top">
      <div class="container">
         <div class="row">
            <div class="col-xl-4 col-md-6 col-lg-4 ">
               <div class="footer_widget">
               </div>
            </div>
            <div class="col-xl-4 col-md-6 col-lg-4">
                                 </div>

         </div>
      </div>
   </div>
   <div class="copy-right_text">
      <div class="container">
         <div class="row">
            <div class="bordered_1px "></div>
            <div class="col-xl-12">
            </div>
         </div>
      </div>
   </div>
</footer>
<!-- link that opens popup -->
<!-- form itself end-->
```

```
<!-- form itself end -->
      <!-- JS here -->
  <script src="../static/js/vendor/modernizr-3.5.0.min.js"></script>
<script src="../static/js/vendor/jquery-1.12.4.min.js"></script>
<script src="../static/js/popper.min.js"></script>
<script src="../static/js/bootstrap.min.js"></script>
<script src="../static/js/owl.carousel.min.js"></script>
<script src="../static/js/isotope.pkgd.min.js"></script>
<script src="../static/js/ajax-form.js"></script>
<script src="../static/js/waypoints.min.js"></script>
<script src="../static/js/jquery.counterup.min.js"></script>
<script src="../static/js/imagesloaded.pkgd.min.js"></script>
<script src="../static/js/scrollIt.js"></script>
<script src="../static/js/nice-select.min.js"></script>
   <script src="../static/js/jquery.magnific-popup.min.js"></script>
<script src="../static/js/plugins.js"></script>
<script src="../static/js/gijgo.min.js"></script>
<script src="../static/js/contact.js"></script>
<script src="../static/js/jquery.ajaxchimp.min.js"></script>
<script src="../static/js/jquery.form.js"></script>
<script src="../static/js/jquery.validate.min.js"></script>
<script src="../static/js/mail-script.js"></script>
<script src="../static/js/main.js"></script>
   <script>
      $('#datepicker').datepicker({
         iconsLibrary: 'fontawesome',
         icons: {
          rightIcon: '<span class="fa fa-caret-down"></span>'
       }
```

```
        });
        $('#datepicker2').datepicker({
            iconsLibrary: 'fontawesome',
            icons: {
             rightIcon: '<span class="fa fa-caret-down"></span>'
         }
            });
    </script>
</body>

</html>
```

APPENDIX 3

A3.SCREEN SHOTS


Home
Login
Upload
Login


Login
admin
Login

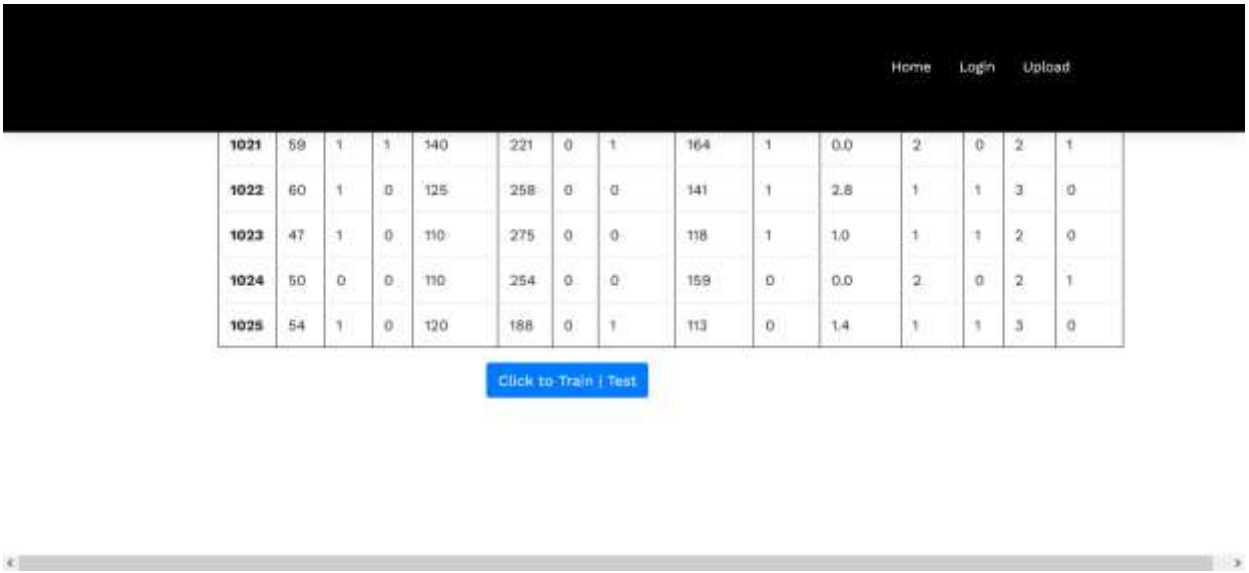
Upload

Browse... No file selected.

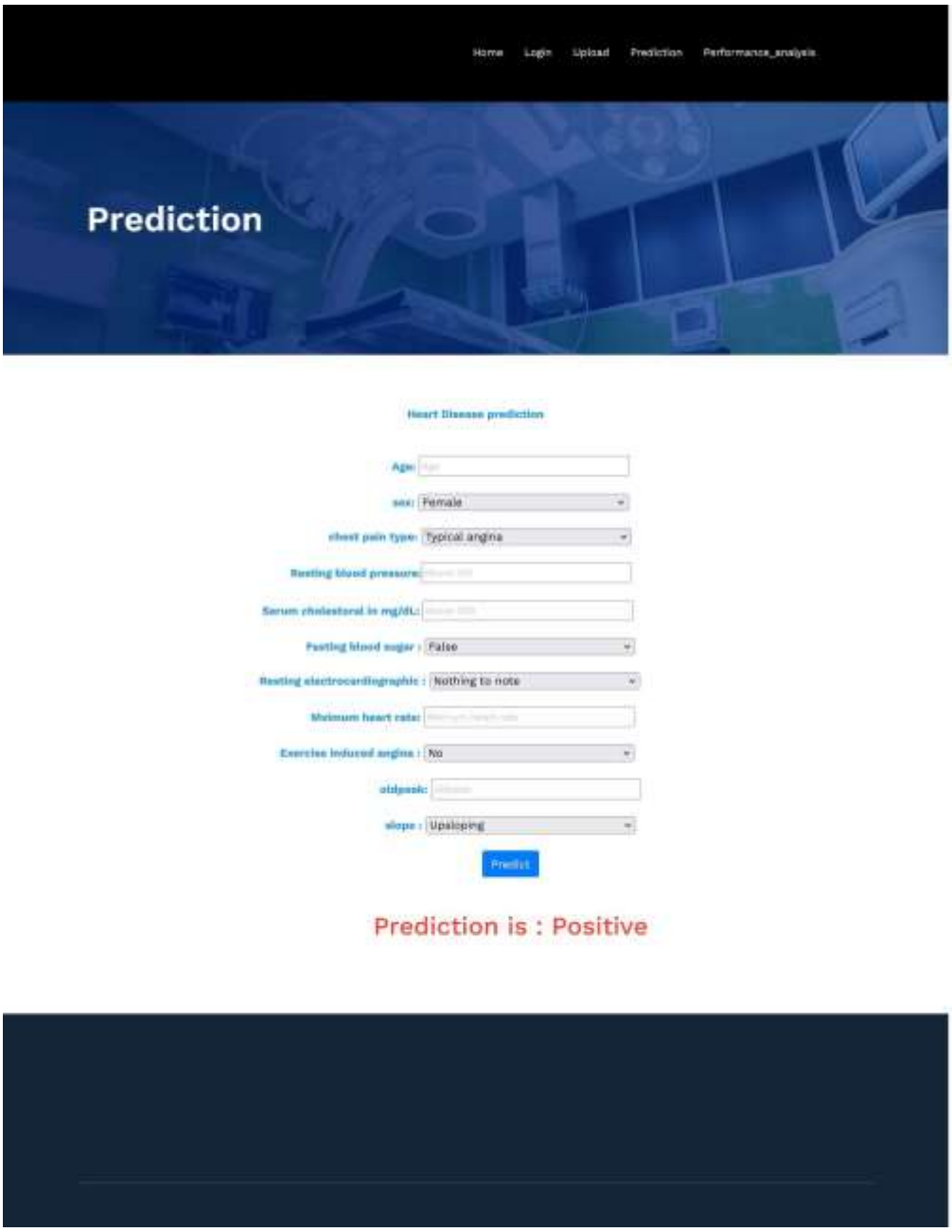
Upload

Home Login Upload

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
id														
1	62	1	0	125	212	0	1	168	0	1.0	2	2	3	0
2	53	1	0	140	203	1	0	155	1	3.1	0	0	3	0
3	70	1	0	145	174	0	1	125	1	2.6	0	0	3	0
4	61	1	0	148	203	0	1	161	0	0.0	2	1	3	0
5	62	0	0	138	294	1	1	106	0	1.9	1	3	2	0
6	58	0	0	100	248	0	0	122	0	1.0	1	0	2	1
7	58	1	0	114	318	0	2	140	0	4.4	0	3	1	0


Home
Login
Upload
1021 59 1 1 140 221 0 1 164 1 0.0 2 0 2 1
1022 60 1 0 125 258 0 0 141 1 2.8 1 1 3 0
1023 47 1 0 110 275 0 0 118 1 1.0 1 1 2 0
1024 50 0 0 110 254 0 0 159 0 0.0 2 0 2 1
1025 54 1 0 120 188 0 1 113 0 1.4 1 1 3 0
Click to Train | Test


Prediction
Heart Disease prediction
Age:
sex: Female
chest pain type: Typical angina
Resting blood pressure:
Serum cholesterol in mg/dL:
Fasting blood sugar : False
Resting electrocardiographic : Nothing to note
Maximum heart rate:
Exercise induced angina : No
oldpeak:
slope : Upsloping
Predict
Prediction is :

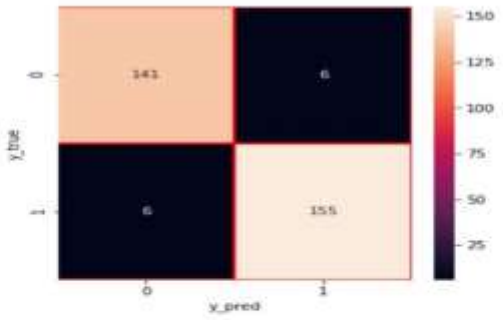

Home
Login
Upload
Prediction
Performance_analysis
Prediction
Heart Disease prediction
Age:
sex:
Female
chest pain type:
Typical angina
Resting blood pressure:
Serum cholesterol in mg/dL:
Fasting blood sugar :
False
Resting electrocardiographic :
Nothing to note
Maximum heart rate:
Exercise induced angina :
No
oldpeak:
slope :
Upsloping
Predict
Prediction is : Positive

PERFORMANCE ANALYSIS

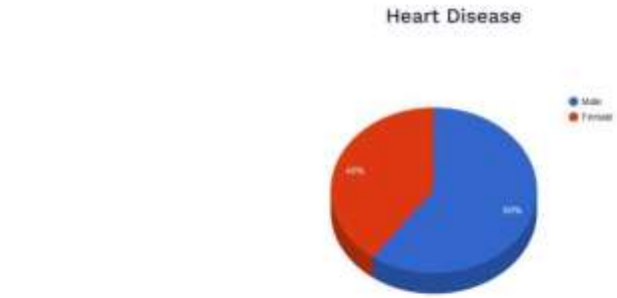
Precision and recall

	Recall	Precision
Negative(0)	0.96	0.96
Positive(1)	0.96	0.96

Confusion Matrix




Home Login Upload Prediction Performance_analysis Chart Logout
Chart


Heart Disease
Male
Female

Demo video link:

https://drive.google.com/file/d/1vwclg0LZIXe2cePNkk2nsPP7YCDJf2Iz/view?usp=drivesdk

Github link:

IBM-Project-15892-1659605994