## 

A project report in partial fulfillment for the award of the degree of

**BACHELOR OF TECHNOLOGY**



Heart Disease Prediction Using Machine Learning

Department of Computer Science and Systems Engineering

Andhra University College Of Engineering (A)

Andhra University, Visakhapatnam -530003

May 2022

**Submitted by,**

* Gnana Sanjana Killi **(318506402009)**
* Jyoti Swaroop Naidu Yelaka **(318506402010)**
* Lokesh Madem **(318506402012)**
* Rahul Bathula **(318506402025)**

**Under the guidance of**

**Prof. S. Pallam Setty**

**Professor**

**Department of CS&SE**

Department of Computer Science and Systems Engineering

Andhra University College Of Engineering (A)

Andhra University, Visakhapatnam -530003



**CERTIFICATE**

This is to certify that this is a bonafide project work entitled **“Heart Disease Prediction Using Machine Learning”**, done by **Ms. Gnana Sanjana Killi (318506402009), Mr. Jyoti Swaroop Naidu Yelaka (318506402010), Mr. Lokesh Madem (318506402012) and Mr. Rahul Bathula (318506402025)**  students of Department of Computer Science & Systems Engineering, Andhra University College of Engineering (A) during the period 2017 - 2021 in the partial fulfillment of the requirements for the award of a degree of **Bachelor of Technology** in **Computer Science and Engineering**.

Prof.Kunjam Nageswara Rao Prof.S.Pallam Setty

Head of the Department Project Guide

Dept of CS&SE Dept of CS&SE

AUCE(A) AUCE(A)

Visakhapatnam Visakhapatnam

**DECLARATION**

We hereby declare that the project entitled **“Heart Disease Prediction Using Machine Learning”** has been prepared by us during the period Feb 2022 – May 2022 in partial fulfillment of the requirements for the award of a degree of **Bachelor of Technology** in **Computer Science and Engineering**.

**Gnana Sanjana Killi**

**(318506402009)**

**Jyoti Swaroop Naidu Yelaka**

**(318506402010)**

**Lokesh Madem**

**(318506402012)**

**Rahul Bathula**

**(318506402025)**

**Place:** Visakhapatnam

**Date:**

## 

**ACKNOWLEDGEMENT**

We have immense pleasure in expressing our earnest gratitude to our Project Guide **Prof. S . PALLAM SETTY SIR**, for his immensely inspiring nature and scholarly guidance. Despite his preoccupation with several assignments, he has been kind enough to spare his valuable time and gave us the necessary counsel and guidance at every stage of planning and constitution of this work. We express our sincere gratitude for his authorization to take up this project. and for helping us graciously throughout the execution of this work.

We express sincere Thanks to **Prof.KUNJAM NAGESHWARA RAO SIR**, Head of the Department of Computer Science and Systems Engineering, Andhra University College of Engineering(A) for his keen interest and for providing the necessary facilities for this project study.

We express sincere Thanks to Prof. **P. SRINIVAS RAO SIR**, Principal, Computer Science and Systems Engineering, Andhra University College of Engineering(A) for his keen interest and for providing the necessary facilities for this project study.

We express sincere gratitude to Prof. **P.V.G.D. PRASAD REDDY SIR**, Vice-Chancellor, Andhra University College of Engineering(A) for his keen interest and for providing necessary facilities for this project study.

We extend our sincere thanks to our academic teaching staff and non-teaching staff for their help throughout our study.

# Table of Contents

# Table of Contents...............................................................................................................................................5

# 1. Abstract.......................................................................................................................................................8

# 2. Introduction.................................................................................................................................................9

# 2.1 Purpose..............................................................................................................................................10

# 2.2 Document Convention........................................................................................................................10

# 2.3 Problem Definition.............................................................................................................................10

# 2.4 Intended Audience and Reading Suggestions.....................................................................................11

# 2.5 Product Scope....................................................................................................................................11

# 2.6 References.........................................................................................................................................11

# 3. Literature Survey.......................................................................................................................................12

# 4. Machine Learning......................................................................................................................................14

# 4.1 Introduction.......................................................................................................................................14

# 4.2 Data in Machine Learn.......................................................................................................................15

# 4.3 Advantages of Machine Learning.......................................................................................................16

# 4.4 Drawbacks of Machine Learning........................................................................................................17

# 4.5 Types of Machine Learning.................................................................................................................18

# 4.6 History of Machine Learning..............................................................................................................21

# 4.7 Key Characteristics of Machine Learning...........................................................................................23

# 4.8 Applications of Machine Learning......................................................................................................25

# 4.9 Key Elements of Machine Learning....................................................................................................28

# 5. Machine Learning Algorithms….................................................................................................................30

# 5.1 Decision Tree Classification Algorithm...............................................................................................30

# 5.2 Naive Bayes Classifier Algorithm.......................................................................................................31

# 5.3 Logistic Regression............................................................................................................................32

# 5.4 Random Forest Algorithm.................................................................................................................33

# 6. Random Forest Algorithm.........................................................................................................................34

# 6.1 Introduction.....................................................................................................................................34

# 6.2 Assumptions for Random Forest......................................................................................................35

# 6.3 How does the Random Forest Algorithm work..................................................................................35

# 6.4 Applications of Random Forest.........................................................................................................36

# 6.5 Advantages of Random Forest..........................................................................................................37

# 6.6 Disadvantages of Random Forest.....................................................................................................37

# 6.7 Python Implementation Steps..........................................................................................................37

**7 Flask........................................................................................................................................................38**

**7.1 Introduction.....................................................................................................................................38**

**7.2 History..............................................................................................................................................38**

**7.3 Components.....................................................................................................................................38**

**7.4 Features...........................................................................................................................................39**

**7.5 Example...........................................................................................................................................40**

**7.6 Templates........................................................................................................................................40**

**7.7 Testing Flask Applications...............................................................................................................40**

**7.8 Handling Application Errors............................................................................................................41**

**7.9 Debugging Application Errors..........................................................................................................41**

**7.10 Flask Applications..........................................................................................................................42**

**8 External Interface Requirement...............................................................................................................43**

**8.1 User Interfaces.................................................................................................................................43**

**8.2 Hardware Interfaces.........................................................................................................................43**

**8.3 Software Interfaces..........................................................................................................................43**

**8.4 Communication Interfaces...............................................................................................................44**

# 9 System Architecture.................................................................................................................................45

# 10 UML Diagrams.........................................................................................................................................46

# 10.1 Activity Diagram.................................................................................................................................46

# 10.2 Architecture Diagram....................................................................................................................47

# 10.3 Data Flow Diagram….....................................................................................................................48

# 10.4 Use Case Diagram..........................................................................................................................49

# 10.5 Sequence Diagram........................................................................................................................50

# 10.6 Flowchart Diagram.......................................................................................................................51

# 10.7 Block Diagram .............................................................................................................................52

# 11 Experimental Analysis.............................................................................................................................53

# 11.1 Sample Code.................................................................................................................................53

# 11.1.1 Main.py ...........................................................................................................................53

# 11.1.2 EasyResponsiveTabs.js.....................................................................................................60

# 11.1.3 JQuery.Slider.js ...............................................................................................................65

# 11.1.4 SwipeBox.css...................................................................................................................76

# 11.1.5 Analyze\_form.html..........................................................................................................86

# 11.2 Sample Dataset............................................................................................................................95

# 11.3 Home Page..................................................................................................................................96

# 11.4 Login Page....................................................................................................................................97

# 11.5 Input.............................................................................................................................................98

# 11.6 Output........................................................................................................................................100

# 12. Conclusion……….......................................................................................................................................101

**13 Future Scope of Development...................................................................................................................101**

# 

# 

# 

# 

# 

# 1. ABSTRACT

Heart disease, alternatively known as cardiovascular disease, encases various conditions that impact the heart and is the primary basis of death worldwide over the span of the past few decades. It associates many risk factors in heart disease and a need for the time to get accurate, reliable, and sensible approaches to make an early diagnosis to achieve prompt management of the disease. Data mining is a commonly used technique for processing enormous data in the healthcare domain. Researchers apply several data mining and machine learning techniques to analyze huge complex medical data, helping healthcare professionals to predict heart disease. Machine Learning is used across many ranges around the world. The healthcare industry is no exception. The Healthcare field has a vast amount of data, for processing those data certain techniques are used. Data mining is one of the techniques often used. Machine Learning can play an essential role in predicting the presence/absence of locomotors disorders, Heart diseases, and more. Such information, if predicted well in advance, can provide important intuitions to doctors who can then adapt their diagnosis and dealing on a per-patient basis.

We work on predicting possible Heart Diseases in people using Machine Learning algorithms. In this project, we perform the comparative analysis of classifiers like decision tree, Naïve Bayes, Logistic Regression, SVM, and Random Forest. The trial results verify that the Random Forest algorithm has achieved the highest accuracy of 90.16% compared to other ML algorithms implemented. Thus, this paper presents a comparative study by analyzing the performance of different machine learning algorithms.The datasets are processed in python programming. In this project, the system evaluates those parameters using the data mining classification technique. The outcomes of this project provide the chances of occurring heart disease in terms of percentage. The given heart disease prediction system enhances medical care and reduces the cost. This project gives us significant knowledge that can help us predict the patients with heart disease.

Keywords: SVM; Naive Bayes; Decision Tree; Random Forest; Logistic Regression; Adaboost; XG-boost; python programming; confusion matrix; correlation matrix.

# 2. INTRODUCTION

The heart is an exquisite machine. The heart is important because it pumps blood around your body, delivering oxygen and nutrients to your cells. According to the World Health Organization, every 17.9 million people surrender their lives to Cardiovascular Diseases and the numbers are going up even now, which could be a great threat. This had been a centipede for years. A lot of people don’t even know that they have heart disease. Heart Disease could be unpredictable and can cause unnatural death in quick succession. It is one of the biggest causes of morbidity and mortality among the population of the world. This had been a major area of study for years. The prediction and prevention of heart diseases could be a great boon to society. Ascertaining cardiovascular disease is considered one of the vital subjects in the section of Data Analytics. Many researches have been conducted in an attempt to pinpoint the most influential factors of heart disease as well as accurately predict the overall risk. As the proverb “Prevention is better than cure” says, predicting heart disease in its early stages and preventing it could be great. The early diagnosis of heart disease plays a vital role in making decisions on lifestyle changes in high-risk patients and in turn, reduces the complications.

Machine learning usually focuses on the use of data and algorithms to imitate the way that humans learn, gradually improving its accuracy. It could be a great advantage in the medical field. This project mainly concentrates on the prediction of future Heart Disease by analyzing data of patients which informs whether they have heart disease or not, using machine-learning algorithms. Considering the key factors of a person’s heart metabolism heart disease can be predicted which might be a life savior. Collecting data from different cases and bringing everything together to analyze all the data and classifying it into different categories by a severity which could be helpful in predicting a heart disease is all about the project.

## 

## 

## 2.1 Purpose

The key purpose of pursuing this project for a while is to create a heart disease prediction model for the prediction of heart disease’s probability in a particular individual. This project can further be developed and can be used to predict different heart diseases and their treatment. This could be a great opportunity to save lives with a portable application. The process starts by collecting required information from an individual and analyzing the data using the most accurate algorithm possible i.e. Random Forest algorithm. After the analysis, the prediction is presented whether the individual has heart disease or not. This will provide researchers and medical practitioners to establish a better way of finding out the disease.

## 2.2 Document Convention

The font used for content text is Open Sans with the font size of 11 and 13. The font used for Headings and Subheadings is PT Sans Narrow with the font size of 18 and 16 respectively. Important points are highlighted.

## 2.3 Problem Definition

The major challenge in the heart is its functioning. If there is an abnormality with its functioning, it should be checked and cleared. There are instruments available that can predict abnormalities in the heart which might be expensive and inefficient in their process. Early detection of cardiac diseases can decrease the risk of danger and all its complications. Monitoring a person all the time can be nearly impossible and consulting a doctor frequently might cost a lot. This is the digital era, where we can get data regarding anything on the internet. With the information available about heart diseases, which could make the process easier. The hidden patterns in machine learning patterns can be used for health diagnosis in medicinal data.

## 2.4 Intended Audience and Reading Suggestions

The intended audience of this document includes all stakeholders of the CMS project who are supposed to review and sign off this document. The primary stakeholders of the system include users, users, administrators, pickup staff, delivery staff, etc.

## 2.5 Product Scope

The project named ‘Heart Disease Prediction’’ aims at implementing an Application Programming Interface that would predict the abnormal activities of an individual's heart. The system shall allow the admin to login from any device, and track their heart’s healthiness. Interesting facts about the heart are also included in the application. The ultimate aim is to ensure the prediction of abnormality in the heart by analyzing various data sets and managing its resources and utilities effectively.

## 2.6 References

* + IEEE Recommended Practice for Software Requirements Specifications
    - IEEE Std 830-1998.
  + Heart Disease Prediction using Machine Learning
    - International Journal of Engineering Research & Technology
  + Software Engineering
    - By Sommerville
  + Software Engineering: Theory and Practice
    - By Forrest Shull and Roseanne Tesoriero
      * + Software Engineering for Students, 4th edition

By DOUGLAS BELL

* + System Analysis and Design
    - By Elias M. Awad
  + Database System ConceptS
    - By Silberschatz,Korth & Sudarshan
      * + [www.wikipedia.org](http://www.wikipedia.org)
        + [www.google.com](http://www.google.com)

# 

# 

# 

# 3. LITERATURE SURVEY

[1]”Design of a Fuzzy-based Decision Support System for Coronary Heart Disease Diagnosis Adel” was a paper proposed by Lahsasna · Raja Noor Ainon · Roziati Zainuddin · Awang Bulgiba which was totally helpful in knowing the essential conditions which can detect small heart disease and knowing how different algorithms could help in the analysis.

[2]Akram Ahmed Mohammed et al proposed a paper “Random Forest Machine Learning technique to predict Heart disease” which did explain how the random forest algorithm works and why it is a better algorithm to predict heart disease. It did show the accuracy of the Random Forest algorithm to be the best among others.

[3] Purushottam et al proposed a paper “Efficient Heart Disease Prediction System” using hill climbing and decision tree algorithms. They used the Cleveland dataset and preprocessing of data is performed before using classification algorithms. The Knowledge Extraction is done based on Evolutionary Learning (KEEL),an open- source data mining tool that fills the missing values in the data set. A decision tree follows top-down order. For each actual node selected by the hill-climbing algorithm, a node is selected by a test at each level. The parameters and their values used are confidence. Its minimum confidence value is 0.25. The accuracy of the system is about 86.7%

[4] Sonam Nikhar et al proposed the paper “ Prediction of Heart Disease Using Machine Learning Algorithms” their research gives point to point explanation of Naïve Bayes and decision tree classifiers that are used especially in the prediction of Heart Disease.

[5]Senthilkumar Mohan et al proposed a paper “Effective Heart Disease Prediction Using Hybrid Machine Learning Techniques” that helped us in checking different techniques other than the traditional machine learning algorithms for more efficient and accurate results. Through hidden layers, each input node is connected to the output layer. This connection is assigned with some random weights.

[6] Aditi Gavhane et al proposed a paper “Prediction of Heart Disease Using Machine Learning”, in which training and testing of the dataset are performed by using neural network algorithm multi-layer perceptron. In this algorithm, there will be one input layer and one output layer, and one or more layers are hidden between these two input and output layers. Through a hidden layer, each input node is connected to the output layer. This connection is assigned with some random weights. The other input is called bias which is assigned with weight-based on requirements the connection between the nodes can be feedforward or feedback.

[7] Avinash Golande et al, proposed “Heart Disease Prediction Using Effective Machine Learning Techniques” in which few data mining techniques are used that support the doctors to differentiate the heart disease. Usually utilized methodologies are k-nearest neighbors, Decision tree, and Naïve Bayes. Other unique characterization-based strategies utilized are packing calculation, Part thickness, consecutive negligible streamlining and neural systems, straight Kernel self-arranging guide, and SVM (Bolster Vector Machine). This model uses deep learning and data mining to give the best precise model and least blunders.

[8] Lakshmana Rao et al,proposed “Machine Learning Techniques for Heart Disease Prediction” in which the contributing elements for heart disease are more. So, it is difficult to distinguish heart disease. To find the seriousness of heart disease among people different neural systems and data mining techniques are used.

[9]Adel Lahsasna et al proposed “Design of a Fuzzy-based Decision Support System for Coronary Heart Disease Diagnosis” which gave us a better idea of how heart diseases could be classified, and what could be some cure with it. Data is directly retrieved from electronic records that reduce the manual tasks. The number of services is decreased and a major number of rules helps within the best prediction of heart disease. This model uses deep learning and data mining to give the best precise model and least blunders.

# 

# 

# 

# 4. MACHINE LEARNING:

## 4.1 Introduction

Machine learning is a branch of Artificial Intelligence(AI) and computer science which focuses on the use of data and algorithms to imitate the way that humans learn, gradually improving its accuracy.

Machine Learning is when a program takes new data, learns from it, and makes changes without being explicitly programmed to do so. Machines perform data sequencing in an automated fashion, combing through sets of data searching for patterns and similarities. Once data patterns and predictive behaviors have been identified, rules must be implemented to take action on learned data. With machine learning, the machine is enabled to create or modify rules to further improve itself and accomplish its primary objectives.

The term Machine Learning was coined by Arthur Samuel in 1959, an American pioneer in the field of computer gaming and artificial intelligence, and stated that “it gives computers the ability to learn without being explicitly programmed”.

So if you want your program to predict, for example, traffic patterns at a busy intersection, you can run it through a machine learning algorithm with data about past traffic patterns and, if it has successfully “learned”, it will then do better at predicting future traffic patterns.

The objectives can vary and have a variety of uses. Machine learning can be used to make routes faster and more efficient for various transportation methods improve sales conversions through product recommendations or tailoring product content to direct purchasing decisions predict hospitalization based on physical 10 behavior and patient data; improve patient diagnostics based on trends or areas of concern; or determining levels of risk for investments or insurance.

Machine learning is important because it gives enterprises a view of trends in customer behavior and business operational patterns, as well as supports the development of new products. Many of today's leading companies, such as Facebook, Google, and Uber, make machine learning a central part of their operations. Machine learning has become a significant competitive differentiator for many companies.

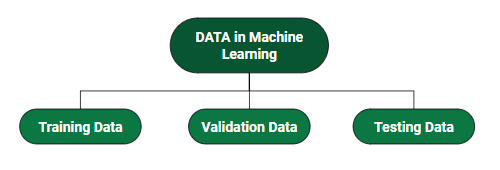
In the times before machine learning, a user must manually provide the program with new sets of rules for any new data in order for any action to occur, as well as decide on what the next step would be to act upon any new rules.

“With machine learning, the program creates algorithms, or a sequence of instructions, to execute in order to accomplish the desired end result as well as recommend and act upon any suggested next steps. Rather than having to manually parse through copious amounts of data, correlate patterns, create algorithms, and execute across systems, users, utilizing automation and machine learning, have been able to improve efficiencies by being more granular and prescriptive, as well as alleviate the workload.”

The highly complex nature of many real-world problems, though, often means that inventing specialized algorithms that will solve them perfectly every time is impractical, if not impossible.

## 4.2 Data in Machine Learning

* **DATA:** It can be any unprocessed fact, value, text, sound, or picture that is not being interpreted and analyzed.
* **INFORMATION**: Data that has been interpreted and manipulated and has now some meaningful inference for the users.
* **KNOWLEDGE**: Combination of inferred information, experiences, learning, and insights. Results in awareness or concept building for an individual or organization.
* **Training Data:** The part of data we use to train our model. This is the data that your model actually sees(both input and output) and learns from.
* **Validation Data:** The part of data that is used to do a frequent evaluation of the model, fits on the training dataset along with improving involved hyperparameters (initially set parameters before the model begins learning). This data plays its part when the model is actually training.
* **Testing Data:** Once our model is completely trained, testing data provides an unbiased evaluation. When we feed in the inputs of Testing data, our model will predict some values(without seeing actual output). After prediction, we evaluate our model by comparing it with the actual output present in the testing data. This is how we evaluate and see how much our model has learned from the experiences fed in as training data, set at the time of training.

****

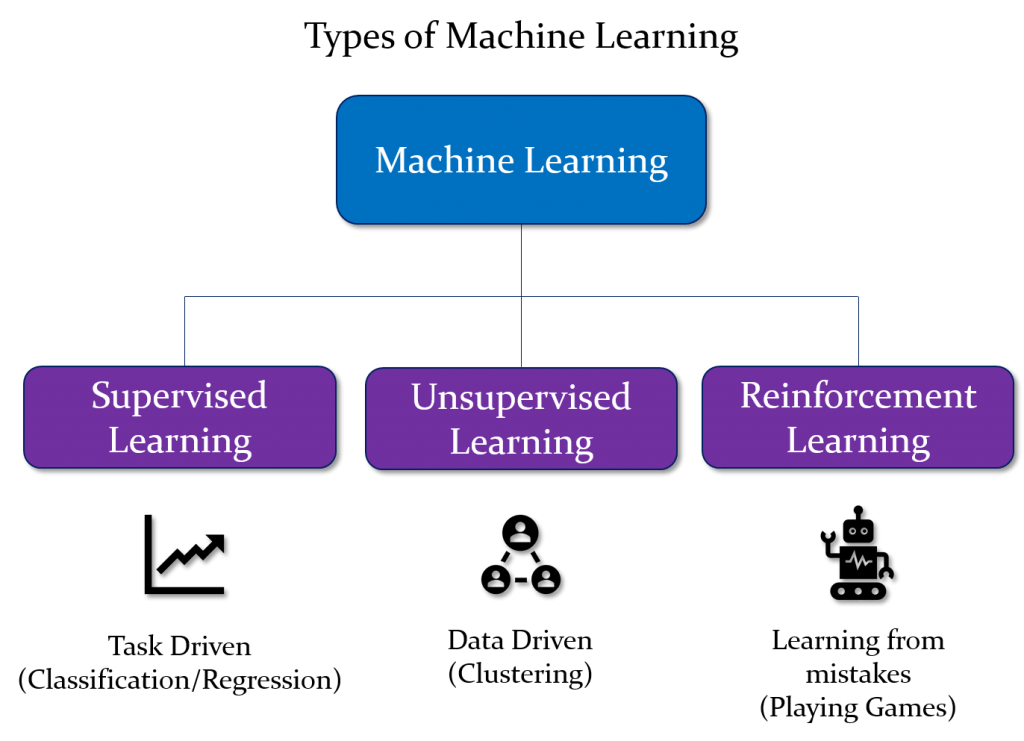
## 4.3 Advantages of machine learning:

* **It is automatic:** In machine learning, the whole process of data interpretation and analysis is done by a computer. No intervention is required for the prediction or interpretation of data. The whole process of machine learning is when the machine starts learning and predicting the algorithm or program to give the best result. One of the examples in the Google home is that detects the voice and then accordingly finds out the result that the user wants, and antivirus software detects the virus of the computer and fixes it.
* **It is used in various fields:** Machine learning is used in various fields of life like education, medicine, engineering, etc. From a very small application to very big and complicated structured machines that help in the prediction and analysis of data. It not only becomes the healthcare provider but also provides more personal services to the potential customer.
* **It can handle varieties of data:** Even in an uncertain and dynamic environment, it can handle a variety of data. It is multidimensional as well as a multitasker.
* **Scope of advancement:** As humans after gaining experience improve themselves in the same way machine learning improves themselves and become more accurate and efficient in work. This led to better decisions. For example, in the weather forecast, the more data. And experience the machine gets the more advanced forecast it will provide.
* **Can identify trends and patterns**: A machine can learn more when it gets more data and since it gets more data it also learns the pattern and trend for example on a social networking site like Facebook people surf and browses several data and their interest is recorded and understand the pattern and shows the same or similar trend to them to keep their interest within the same app. In this way machine learning help in identifying trends and patterns.
* **Considered best for Education:** Machine learning is considered best for education as education is dynamic and nowadays smart classes, distance learning, and e-learning for students have increased a lot. Smart machine learning will act as a teacher and keep students updated with the current scenario of the world. The same thing happens in shopping or e-business. People need to remain updated therefore they are shown the current trends of the world.

## 4.4 Drawbacks of Machine Learning:

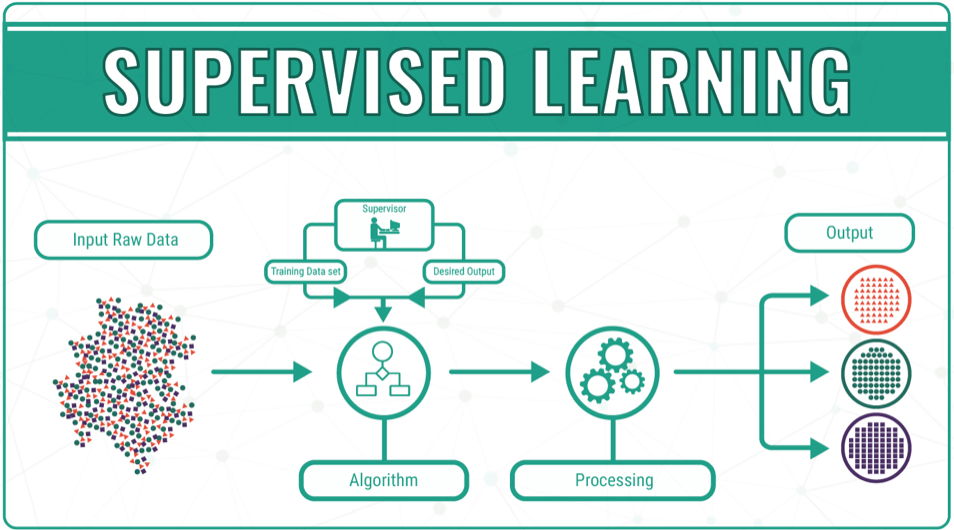
* **Chance of error or fault is more:** Although machine learning is considered to be more accurate it is highly vulnerable. For example, a set of programs provided to the machine may be biased or consist of errors. The same program is used to make another forecast or prediction then there will be a chain of errors that could be formed which may, although recognized but take some time to find out the source of the error
* **Data requirement is more:** The more data a machine gets the more accurate and efficient it becomes thus more data is required to input to the machine for better forecasting or decision making. But it may sometimes not be possible. Also, the data must be unbiased and of good quality. Data requirements are problematic sometimes.
* **Time-consuming and more resources required:** There can be times when the learning process of the machine may take a lot of time because the effectiveness and efficiency can only come through experience which again requires time. Also, the resources required are more for example additional computers may be required.
* **Inaccuracy of interpretation of data:** As we have already seen that a little manipulation or biased data could lead to a long drawn error chain and therefore there are chances of the inaccuracy of interpretation also. Sometimes data without any error could also be interpreted inaccurately by the machine as the data provided previously may not fulfill all the basics of the machine.
* **More space required:** As more data is required for interpretation more space is required to store the data which is one of the shortcomings of machine learning. More data means more knowledge or material to learn from for the machine, this requires a lot of space to store or manage data for further decision making.

## 4.5 Types of Machine Learning methods



* **Supervised Learning:**

Supervised learning is when the model is getting trained on a labeled dataset. A labeled dataset is one that has both input and output parameters. In this type of learning both training and validation, datasets are labeled as shown in the figures below. While training the model, data is usually split in the ratio of 80:20 i.e. 80% as training data and the rest as testing data. In training data, we feed input as well as output for 80% of data. The model learns from training data only. Once the model is ready then it is good to be tested. At the time of testing, the input is fed from the remaining 20% of data that the model has never seen before, the model will predict some value and we will compare it with the actual output and calculate the accuracy.

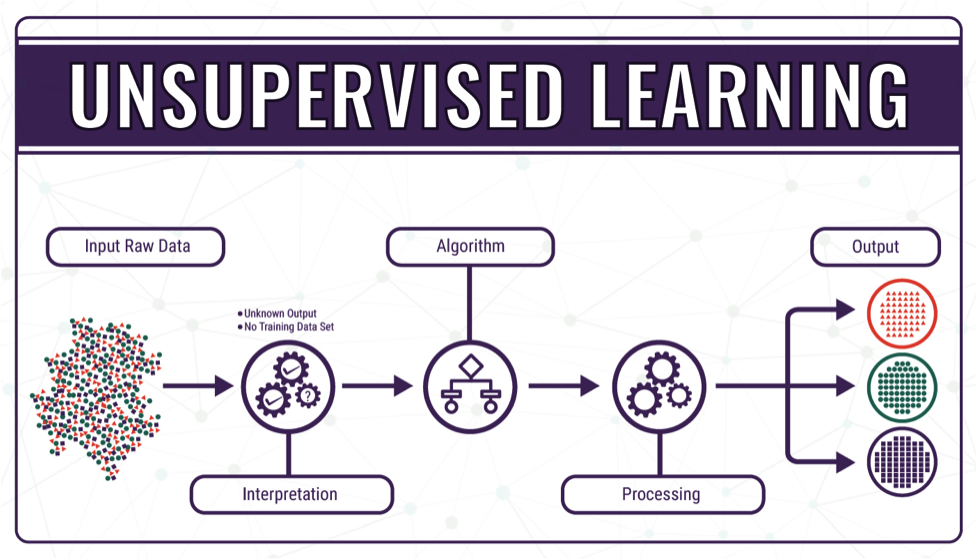


Types of Supervised Learning:

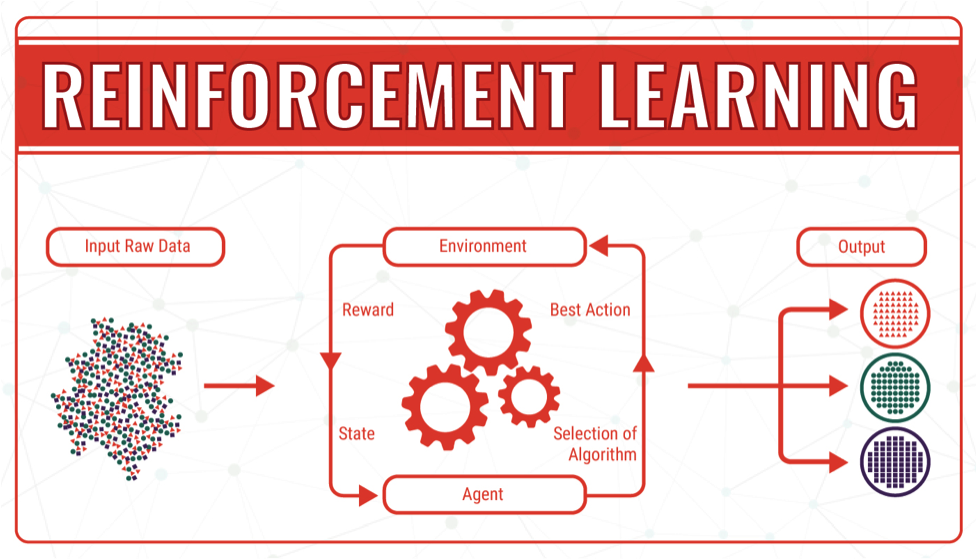
* Classification: It is a Supervised Learning task where output is having defined labels(discrete value).
* Regression: It is a Supervised Learning task where output has continuous value.
* **Unsupervised Learning:**It’s a type of learning where we don’t give a target to our model while training i.e. training model has only input parameter values. The model by itself has to find which way it can learn.

Training data we are feeding is –

* Unstructured Data: May contain noisy(meaningless) data, missing values, or unknown data
* Unlabeled data: Data only contains a value for input parameters, there is no targeted value (output).

****

* **Reinforced Learning:**

This model keeps on increasing its performance by using Reward Feedback to learn the behavior or pattern. These algorithms are specific to a particular problem e.g. Google Self Driving car, AlphaGo where a bot competes with humans and even itself to get better and better performers in the Go Game. Each time we give data, they learn and add the data to their knowledge which is training data. So, the more it learns the better it gets trained and experienced.****

4.6 History of Machine Learning:

**1950** — Alan Turing creates the “Turing Test” to determine if a computer has real intelligence. To pass the test, a computer must be able to fool a human into believing it is also human.

**1952** — Arthur Samuel wrote the first computer learning program. The program was the game of checkers, and the [IBM](https://www.forbes.com/companies/ibm/) computer improved at the game the more it played, studying which moves made up winning strategies and incorporating those moves into its program.

**1957** — Frank Rosenblatt designed the first neural network for computers (the perceptron), which simulate the thought processes of the human brain.

**1967** — The “nearest neighbor” algorithm was written, allowing computers to begin using very basic pattern recognition. This could be used to map a route for traveling salesmen, starting at a random city but ensuring they visit all cities during a short tour.

**1979** — Students at Stanford University invent the “Stanford Cart” which can navigate obstacles in a room on its own.

**1981** — Gerald Dejong introduces the concept of Explanation Based Learning (EBL), in which a computer analyses training data and creates a general rule it can follow by discarding unimportant data.

**1985** — Terry Sejnowski invents NetTalk, which learns to pronounce words the same way a baby does.

**1990s** — Work on machine learning shifts from a knowledge-driven approach to a data-driven approach. Scientists begin creating programs for computers to analyze large amounts of data and draw conclusions — or “learn” — from the results.

**1997** — IBM’s Deep Blue beats the world champion at chess.

**2006** — Geoffrey Hinton coins the term “deep learning” to explain new algorithms that let computers “see” and distinguish objects and text in images and videos.

**2010** — The Microsoft Kinect can track 20 human features at a rate of 30 times per second, allowing people to interact with the computer via movements and gestures.

**2011** — IBM’s Watson beats its human competitors at Jeopardy.

**2011** — Google Brain is developed, and its deep neural network can learn to discover and categorize objects much the way a cat does.

**2012** — Google’s X Lab develops a machine learning algorithm that is able to autonomously browse YouTube videos to identify the videos that contain cats.

**2014** — Facebook develops DeepFace, a software algorithm that is able to recognize or verify individuals on photos to the same level as humans can.

**2015** — Amazon launches its own machine learning platform.

**2015** — Microsoft creates the Distributed Machine Learning Toolkit, which enables the efficient distribution of machine learning problems across multiple computers.

**2015** — Over 3,000 AI and Robotics researchers, endorsed by Stephen Hawking, Elon Musk, and Steve Wozniak (among many others), sign an open letter warning of the danger of autonomous weapons which select and engage targets without human intervention.

**2016** — Google’s artificial intelligence algorithm beats a professional player at the Chinese board game Go, which is considered the world’s most complex board game and is many times harder than chess. The AlphaGo algorithm developed by Google DeepMind managed to win five games out of five in the Go competition.

## 4.7 Key Characteristics of Machine Learning:

Inorder to understand the actual power of machine learning**,** you have to consider the characteristics of this technology. There are lots of examples that echo the characteristics of machine learning in today’s data-rich world. Here are seven key characteristics of machine learning for which companies should prefer it over other technologies.

* **The ability to perform automated data visualization**

A massive amount of data is being generated by businesses and common people on a regular basis. By visualizing notable relationships in data, businesses can not only make better decisions but build confidence as well. Machine learning offers a number of tools that provide rich snippets of data which can be applied to both unstructured and structured data. With the help of user-friendly machine learning, businesses can obtain a wealth of new insights in an effort to automated data visualization platforms to increase productivity in their processes.

* **Automation at its best**

One of the biggest characteristics of machine learning is its ability to automate repetitive tasks and thus, increase productivity. A huge number of organizations are already using machine learning-powered paperwork and email automation. In the financial sector, for example, a huge number of repetitive, data-heavy, and predictable tasks are needed to be performed. Because of this, this sector uses different types of machine learning solutions to a great extent. They make accounting tasks faster, more insightful, and more accurate. Some aspects that have been already addressed by machine learning include addressing financial queries with the help of chatbots, making predictions, managing expenses, simplifying invoicing, and automating bank reconciliations.

* **Customer engagement like never before**

For any business, one of the most crucial ways to drive engagement, promote brand loyalty and establish long-lasting customer 19 relationships is by triggering meaningful conversations with its target customer base. Machine learning plays a critical role in enabling businesses and brands to spark more valuable conversations in terms of customer engagement. The technology analyzes particular phrases, words, sentences, idioms, and content formats that resonate with certain audience members. You can think of Pinterest which is successfully using machine learning to personalize suggestions to its users. It uses the technology to source content in which users will be interested, based on objects which they have pinned already.

* **The ability to take efficiency to the next level when merged with IoT**

Thanks to the huge hype surrounding IoT, machine learning has experienced a great rise in popularity. IoT is being designated as a strategically significant area by many companies. And many others have launched pilot projects to gauge the potential of IoT in the context of business operations. But attaining financial benefits through IoT isn’t easy. In order to achieve success, companies, which are offering IoT consulting services and platforms, need to clearly determine the areas that will change with the implementation of IoT strategies. Many of these businesses have failed to address it. In this scenario, machine learning is probably the best technology that can be used to attain higher levels of efficiency. By merging machine learning with IoT, businesses can boost the efficiency of their entire production processes.

* **The ability to change the mortgage market**

It’s a fact that fostering a positive credit score usually takes discipline, time, and lots of financial planning for a lot of consumers. When it comes to the lenders, the consumer credit score is one of the biggest measures of creditworthiness that involve a number of factors including payment history, total debt, length of credit history, etc. But wouldn’t it be great if there is a simplified and better measure? With the help of machine learning, lenders can now obtain a more comprehensive consumer picture. They can now predict whether the customer is a low spender or a high spender and understand his/her tipping point of spending. Apart from mortgage lending, financial institutions are using the same techniques for other types of consumer loans.

* **Accurate data analysis**

Traditionally, data analysis has always encompassed trial and error methods, an approach that becomes impossible when we are working with large and heterogeneous datasets. Machine learning comes as the best solution to all these issues by offering effective alternatives to analyzing massive volumes of data. By developing efficient and fast algorithms, as well as, data-driven models for processing of data in real-time, machine learning is able to generate accurate analysis and results.

* **Business intelligence at its best**

Machine learning characteristics, when merged with big data analytical work, can generate extreme levels of business intelligence with the help of which several different industries are making strategic initiatives. From retail to financial services to healthcare, and many more — machine learning has already become one of the most effective technologies to boost business operations. Whether you are convinced or not, the above characteristics of machine learning have contributed heavily toward making it one of the most crucial technology trends — it underlies a huge number of things we use these days without even thinking about them.

## 4.8 Applications of Machine Learning:

* **Image Recognition:**

Image recognition is one of the most common applications of machine learning. It is used to identify objects, persons, places, digital images, etc. The popular use case of image recognition and face detection is the Automatic friend tagging suggestion. For example, Facebook provides us with a feature of auto friend tagging suggestions. Whenever we upload a photo with our Facebook friends, then we automatically get a tagging suggestion with a name, and the technology behind this is machine learning's face detection and recognition algorithm. It is based on the Facebook project named "Deep Face," which is responsible for face recognition and person identification in the picture.

* **Speech Recognition:**

While using Google, we get an option of "**Search by voice**," it comes under speech recognition, and it's a popular application of machine learning. Speech recognition is a process of converting voice instructions into text, and it is also known as "**Speech to text**", or "**Computer speech recognition**." At present, machine learning algorithms are widely used in various applications of speech recognition. Google Assistant, Siri, Cortana, and Alexa are using speech recognition technology to follow voice instructions.

* **Traffic prediction:**

If we want to visit a new place, we take the help of Google Maps, which shows us the correct path with the shortest route and predicts the traffic conditions. It predicts the traffic conditions such as whether traffic is cleared, slow-moving, or heavily congested with the help of two ways:

* Real-Time location of the vehicle from Google Map app and sensors
* The average time taken on past days at the same time.

Everyone who is using Google Map is helping this app to make it better. It takes information from the user and sends it back to its database to improve its performance.

* **Product recommendations:**

Machine learning is widely used by various e-commerce and entertainment companies such as Amazon, Netflix, etc., for product recommendations to the user. Whenever we search for some product on Amazon, then we start getting an advertisement for the same product while internet surfing on the same browser, and this is because of machine learning. Google understands the user interest using various machine learning algorithms and suggests the product as per customer interest. As similar, when we use Netflix, we find some recommendations for entertainment series, movies, etc., and this is also done with the help of machine learning.

* **Self-driving cars:**

One of the most exciting applications of machine learning is self-driving cars. Machine learning plays a significant role in self-driving cars. Tesla, the most popular car manufacturing company, is working on self-driving cars. It is using an unsupervised learning method to train the car models to detect people and objects while driving.

* **Email Spam and Malware Filtering:**

Whenever we receive a new email, it is filtered automatically as important, normal, and spam. We always receive important mail in our inbox with the important symbol and spam emails in our spam box, and the technology behind this is Machine learning. Below are some spam filters used by Gmail:

* Content Filter
* Header filter
* General blacklists filter
* Rules-based filters
* Permission filters

Some machine learning algorithms such as Multi-Layer Perceptron, Decision tree, and Naïve Bayes classifier are used for email spam filtering and malware detection.

* **Virtual Personal Assistant:**

We have various virtual personal assistants such as Google Assistant, Alexa, Cortana, and Siri. As the name suggests, they help us in finding the information using our voice instructions. These assistants can help us in various ways just by our voice instructions such as Play music, calling someone, Opening an email, Scheduling an appointment, etc. These virtual assistants use machine learning algorithms as an important part. These assistants record our voice instructions, send them over to the server on a cloud, decode it using ML algorithms and act accordingly.

* **Online Fraud Detection:**

Machine learning is making our online transactions safe and secure by detecting fraud transactions. Whenever we perform some online transaction, there may be various ways that a fraudulent transaction can take place such as fake accounts, fake ids, and stealing money in the middle of a transaction. So to detect this, Feed Forward Neural network helps us by checking whether it is a genuine transaction or a fraud transaction. For each genuine transaction, the output is converted into some hash values, and these values become the input for the next round. For each genuine transaction, there is a specific pattern that gets changed for the fraud transaction hence, it detects it and makes our online transactions more secure.

* **Stock Market trading:**

Machine learning is widely used in stock market trading. In the stock market, there is always a risk of ups and downs in shares, so for this machine learning's **long short-term memory neural network** is used for the prediction of stock market trends.

* **Medical Diagnosis:**

In medical science, machine learning is used for disease diagnoses. With this, medical technology is growing very fast and able to build 3D models that can predict the exact position of lesions in the brain. It helps in finding brain tumors and other brain-related diseases easily.

* **Automatic Language Translation:**

Nowadays, if we visit a new place and we are not aware of the language then it is not a problem at all, as for this also machine learning helps us by converting the text into our known languages. Google's GNMT (Google Neural Machine Translation) provides this feature, which is a Neural Machine Learning that translates the text into our familiar language, and it is called automatic translation. The technology behind the automatic translation is a sequence to the sequence learning algorithm, which is used with image recognition and translates the text from one language to another language.

## 

## 

## 4.9 Key elements of Machine Learning:

There are tens of thousands of machine learning algorithms and hundreds of new algorithms are developed every year. Every machine learning algorithm has three components:

* **Representation:**

How to represent knowledge. Examples include decision trees, sets of rules, instances, graphical models, neural networks, support vector machines, model ensembles, and others.

* **Evaluation:**

The way to evaluate candidate programs (hypotheses). Examples include accuracy, prediction and recall, squared error, likelihood, posterior probability, cost, margin, entropy k-L divergence, and others.

* **Optimization:**

The way candidate programs are generated is known as the search process. For example, combinatorial optimization, convex optimization, and constrained optimization.

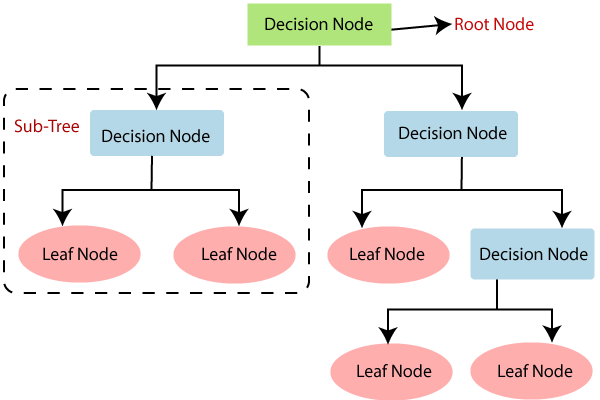
**5. MACHINE LEARNING ALGORITHMS**

****

## 

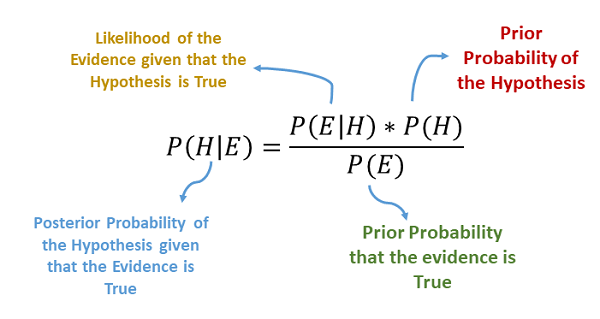
## 5.1 Decision Tree Classification Algorithm:

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 splits the tree into subtrees.



## 5.2 Naïve Bayes Classifier Algorithm

## The Naïve Bayes algorithm is a supervised learning algorithm, which is based on the Bayes theorem and is used for solving classification problems. It is mainly used in text classification that includes a high-dimensional training dataset. Naïve Bayes Classifier is one of the simplest and most effective Classification algorithms which helps in building the fast machine learning models that can make quick predictions. It is a probabilistic classifier, which means it predicts on the basis of the probability of an object. Some popular examples of the Naïve Bayes Algorithm are spam filtration, Sentimental analysis, and classifying articles.



## 

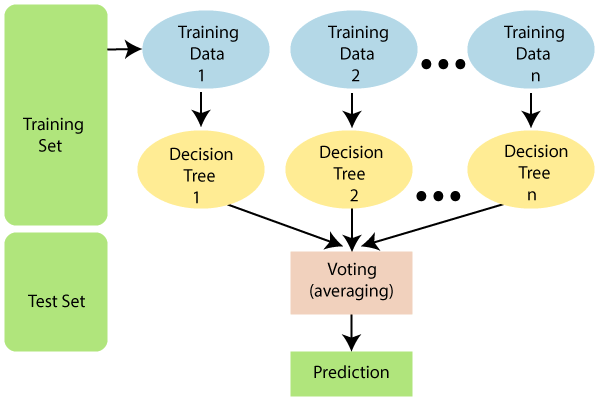
## 5.3 Logistic Regression

Logistic regression is one of the most popular Machine Learning algorithms, which comes under the Supervised Learning technique. It is used for predicting the categorical dependent variable using a given set of independent variables. Logistic regression predicts the output of a categorical dependent variable. Therefore the outcome must be a categorical or discrete value. It can be either Yes or No, 0 or 1, True or False, etc. but instead of giving the exact value as 0 and 1, it gives the probabilistic values which lie between 0 and 1. Logistic Regression is much similar to Linear Regression except that how they are used. Linear Regression is used for solving Regression problems, whereas Logistic regression is used for solving the classification problems. In Logistic regression, instead of fitting a regression line, we fit an "S" shaped logistic function, which predicts two maximum values (0 or 1). The curve from the logistic function indicates the likelihood of something such as whether the cells are cancerous or not, wheter a mouse is obese or not based on its weight, etc. Logistic Regression is a significant machine learning algorithm because it has the ability to provide probabilities and classify new data using continuous and discrete datasets.

## 

## 5.4 Random Forest Algorithm

Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of ensemble learning, which is a process of combining multiple classifiers to solve a complex problem and improve the performance of the model. As the name suggests, "Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset." Instead of relying on one decision tree, the random forest takes the prediction from each tree, and based on the majority votes of predictions, it predicts the final output. The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting.

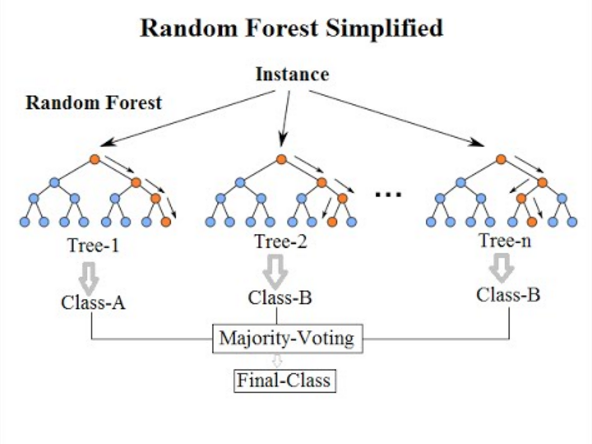


# 6. Random Forest Algorithm:

## 6.1 Introduction

Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of ensemble learning, which is a process of combining multiple classifiers to solve a complex problem and improve the performance of the model.

As the name suggests, "Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset." Instead of relying on one decision tree, the random forest takes the prediction from each tree, and based on the majority votes of predictions, it predicts the final output. The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting.



## 

## 6.2 Assumptions for Random Forest

Since the random forest combines multiple trees to predict the class of the dataset, it is possible that some decision trees may predict the correct output, while others may not. But together, all the trees predict the correct output. Therefore, below are two assumptions for a better Random forest classifier:

* There should be some actual values in the feature variable of the dataset so that the classifier can predict accurate results rather than a guessed result.
* The predictions from each tree must have very low correlations.

## 6.3 How does the Random Forest algorithm work?

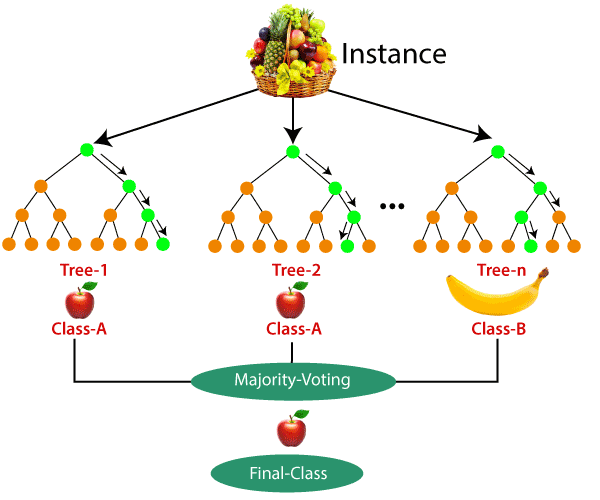
Random Forest works in two-phase first is to create the random forest by combining N decision trees, and the second is to make predictions for each tree created in the first phase.

The Working process can be explained in the below steps and diagram:

* **Step-1:** Select random K data points from the training set.
* **Step-2:** Build the decision trees associated with the selected data points (Subsets).
* **Step-3:** Choose the number N for the decision trees that you want to build.
* **Step-4:** Repeat Step 1 & 2.
* **Step-5:** For new data points, find the predictions of each decision tree, and assign the new data points to the category that wins the majority of votes.

The working of the algorithm can be better understood by the below example:

Example: Suppose there is a dataset that contains multiple fruit images. So, this dataset is given to the Random forest classifier. The dataset is divided into subsets and given to each decision tree. During the training phase, each decision tree produces a prediction result, and when a new data point occurs, then based on the majority of results, the Random Forest classifier predicts the final decision. Consider the below image:



## 6.4 Applications of Random Forest

There are mainly four sectors where Random forest is mostly used:

* **Banking:** The Banking sector mostly uses this algorithm for the identification of loan risk.
* **Medicine:** With the help of this algorithm, disease trends and risks of the disease can be identified.
* **Land Use:** We can identify the areas of similar land use by this algorithm.
* **Marketing:** Marketing trends can be identified using this algorithm.

## 6.5 Advantages of Random Forest

* Random Forest is capable of performing both Classification and Regression tasks.
* It is capable of handling large datasets with high dimensionality.
* It enhances the accuracy of the model and prevents the overfitting issue.

## 

## 6.6 Disadvantages of Random Forest

Although random forest can be used for both classification and regression tasks, it is not more suitable for Regression tasks.

## 6.7 Python Implementation Steps

* Data Preprocessing step
* Fitting the Random forest algorithm to the Training set
* Predicting the test result
* Test accuracy of the result (Creation of Confusion matrix)
* Visualizing the test set result.

## 

## 

## 

## 

## 

## 

## **7. Flask:**

## 7.1 Introduction

## Flask is a micro web framework written in Python. It is classified as a microframework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions. However, Flask supports extensions that can add application features as if they were implemented in Flask itself. Extensions exist for object-relational mappers, form validation, upload handling, various open authentication technologies, and several common framework-related tools.The rest of the docs describe each component of Flask in detail, with a full reference in the API section.

## 7.2 History

Flask was created by Armin Ronacher of Pocoo, an international group of Python enthusiasts formed in 2004. According to Ronacher, the idea was originally an April Fool's joke that was popular enough to make into a serious application. The name is a play on the earlier Bottle framework.

When Ronacher and Georg Brandl created a bulletin board system written in Python in 2004, the Pocoo projects Werkzeug and Jinja were developed.

In April 2016, the Pocoo team was disbanded and the development of Flask and related libraries was passed to the newly formed Pallets project.

Flask has become popular among Python enthusiasts. As of October 2020, it has the second most stars on GitHub among Python web-development frameworks, only slightly behind Django, and was voted the most popular web framework in the Python Developers Survey 2018.

## 7.3 Components

The microframework Flask is part of the Pallets Projects (formerly Pocoo), and based on several others of them, all under a BSD license.

* **Werkzeug** 
  + Werkzeug (German for "tool") is a utility library for the Python programming language for Web Server Gateway Interface (WSGI) applications. Werkzeug can instantiate objects for request, response, and utility functions. It can be used as the basis for a custom software framework and supports Python 2.7 and 3.5 and later.
* **Jinja**
  + Jinja, also by Ronacher, is a template engine for the Python programming language. Similar to the Django web framework, it handles templates in a sandbox.
* **MarkupSafe**
  + MarkupSafe is a string handling library for the Python programming language. The eponymous MarkupSafe type extends the Python string type and marks its contents as "safe"; combining MarkupSafe with regular strings automatically escapes the unmarked strings while avoiding double escaping of already marked strings.
* **ItsDangerous** 
  + ItsDangerous is a safe data serialization library for the Python programming language. It is used to store the session of a Flask application in a cookie without allowing users to tamper with the session contents.

## 7.4 Features

* Development server and debugger
* Integrated support for unit testing
* RESTful request dispatching
* Uses Jinja templating
* Support for secure cookies (client-side sessions)
* 100% WSGI 1.0 compliant
* Complete documentation
* Google App Engine compatibility
* Extensions available to extend the functionality

## 7.5 Example

The following code shows a simple web application that displays "Hello World!" when visited:

from flask import Flask

app = Flask(\_\_name\_\_)

@app.route("/")

def hello() -> str:

return "Hello World"

if \_\_name\_\_ == "\_\_main\_\_":

app.run(debug=False)

## 7.6 Templates

Flask leverages Jinja2 as its template engine. You are obviously free to use a different template engine, but you still have to install Jinja2 to run Flask itself. This requirement is necessary to enable rich extensions. An extension can depend on Jinja2 being present. Generating HTML content from Python code is cumbersome, especially when variable data and Python language elements like conditionals or loops need to be put. This would require frequent escaping from HTML. This is where one can take advantage of the Jinja2 template engine, on which Flask is based. Instead of returning hardcode HTML from the function, a HTML file can be rendered by the render\_template() function.

## 7.7 Testing Flask Applications

Tests are typically located in the tests folder. Tests are functions that start with test\_, in Python modules that start with test\_. Tests can also be further grouped into classes that start with Test. It can be difficult to know what to test. Generally, try to test the code that you write, not the code of libraries that you use, since they are already tested. Try to extract complex behaviors as separate functions to test individually. Pytest fixtures allow writing pieces of code that are reusable across tests. A simple fixture returns a value, but a fixture can also do setup, yield a value, then do the teardown. Fixtures for the application, test client, and CLI runner are shown below, they can be placed in tests/conftest.py.

## 7.8 Handling Application Errors

Applications fail, servers fail. Sooner or later you will see an exception in production.Even if your code is 100% correct, you will still see exceptions from time to time. Why? Because everything else involved will fail. Here are some situations where perfectly fine code can lead to server errors:

* the client terminated the request early and the application was still reading from the incoming data
* the database server was overloaded and could not handle the query
* a filesystem is full
* a hard drive crashed
* a backend server overloaded
* a programming error in a library you are using
* network connection of the server to another system failed

Sending error mails, even if just for critical ones, can become overwhelming if enough users are hitting the error, and log files are typically never looked at. This is why we recommend using Sentry for dealing with application errors. It’s available as a source-available project on GitHub and is also available as a hosted version which you can try for free. Sentry aggregates duplicate errors, captures the full stack trace and local variables for debugging, and sends you emails based on new errors or frequency thresholds.

## 7.9 Debugging Application Errors

The debugger allows executing arbitrary Python code from the browser. It’s protected by a pin, but that should not be relied on for security. Use an error logging tool, such as Sentry, as described in Error Logging Tools, or enable logging and notifications as described in Logging. If you have access to the server, you could add some code to start an external debugger if request.remote\_addr matches your IP. Some IDE debuggers also have a remote mode so breakpoints on the server can be interacted with locally. Only enable a debugger temporarily. The built-in Werkzeug development server provides a debugger that shows an interactive traceback in the browser when an unhandled error occurs during a request. This debugger should only be used during development. To enable the debugger, run the development server with the FLASK\_ENV environment variable set to development. This puts Flask in debug mode, which changes how it handles some errors, and enables the debugger and reloader.

7.10 Flask Applications

* + - Blog App
    - Social Network Web App
    - Weather App
    - Portfolio Website
    - Feedback Form
    - Rest API App
    - Deploying Machine Learning models using the Flask app

# 

# 8. EXTERNAL INTERFACE REQUIREMENT

## 8.1 User Interfaces

The user interface (UI) is the point of human-computer interaction and communication in a device. This can include display screens, keyboards, a mouse, and the appearance of a desktop. It is also the way through which a user interacts with an application or a website.

## 8.2 Hardware Interfaces

* Processor : Any Update Processor
* Ram : Min 4GB
* Hard Disk : Min 100GB

## 8.3 Software Interfaces

## Operating System - Windows XP

* + - Windows : Microsoft Windows commonly referred to as Windows, is a group of several proprietary graphical operating system families, all of which are developed and marketed by Microsoft.

## Web Browser - Google Chrome

* + - Google Chrome : Google Chrome is a cross-platform web browser developed by Google. It was first released in 2008 for Microsoft Windows built with free software components from Apple WebKit and Mozilla Firefox.

## Front end - HTML, CSS, Bootstrap, JavaScript,JQuery

* + - HTML : HTML(HyperText Markup Language) is used to create and save web documents. E.g. Notepad/Notepad++
    - CSS : (Cascading Style Sheets) is used to create an attractive Layout.
    - Bootstrap : It is used for Responsive design and mobile-friendly sites.
    - JavaScript : It is a programming language, commonly used with web browsers.
    - JQuery : jQuery is a JavaScript library designed to simplify HTML DOM tree traversal and manipulation, as well as event handling, CSS animation, and Ajax.

## Back end - Python, MySQL

* + - Python : Python is a high-level, interpreted, interactive, and object-oriented scripting language. Python is designed to be highly readable.
    - MySQL : MySQL is a database, widely used for accessing, querying, updating, and managing data in databases.

## IDE - Visual Studio Code

* + - Visual Studio Code : It is a code editor redefined and optimized for building and debugging modern web and cloud applications.

## 8.4 Communication Interfaces

Users will access the application using web browsers like Google Chrome.The communication will be through standard protocols.

# 

# 

# 

# 

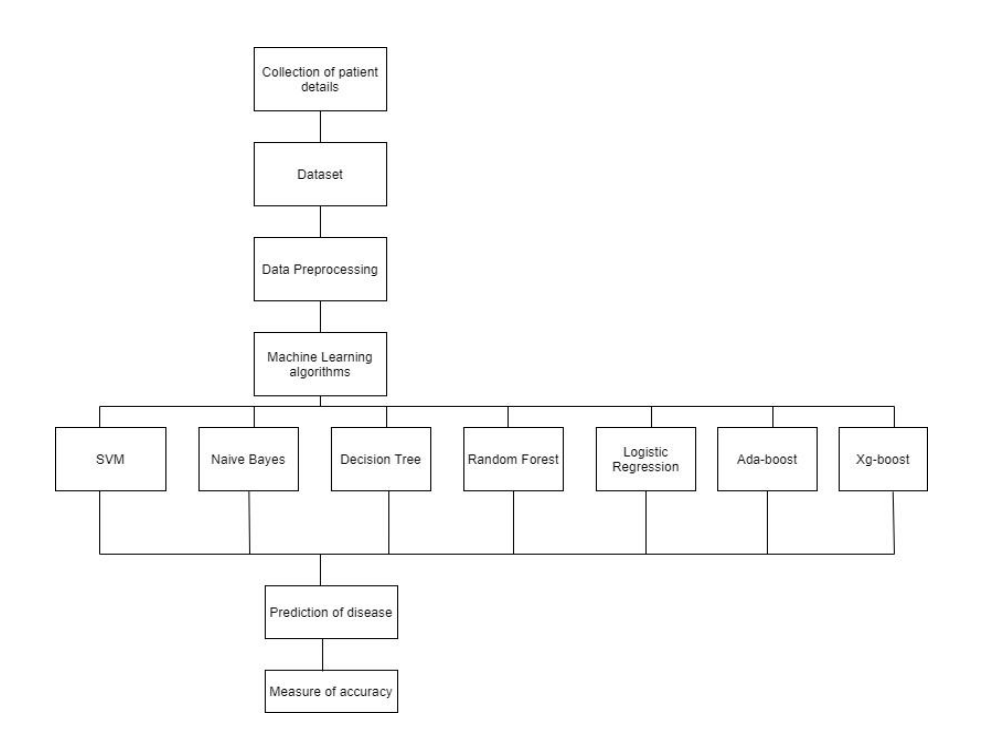
# 

# 

# 

# 9. SYSTEM ARCHITECTURE

The system architecture gives an overview of the working of the system. Dataset collection is collecting data that contains patient details. Attributes the selection process selects the useful attributes for the prediction of heart disease. After identifying the available data resources, they are further selected, cleaned, and made into the desired form. Different classification techniques as stated will be applied to preprocessed data to predict the accuracy of heart disease. Accuracy measure compares the accuracy of different classifiers.



# 10. UML DIAGRAMS

## 10.1 Activity diagram

An 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.



## 

## 10.2 Architecture Diagram

A simple architecture diagram (UML) helps system designers and developers visualize the high-level structure of their system or application to ensure it meets their users' needs. It can also help describe patterns that are used throughout the design.

## 

## 

## 

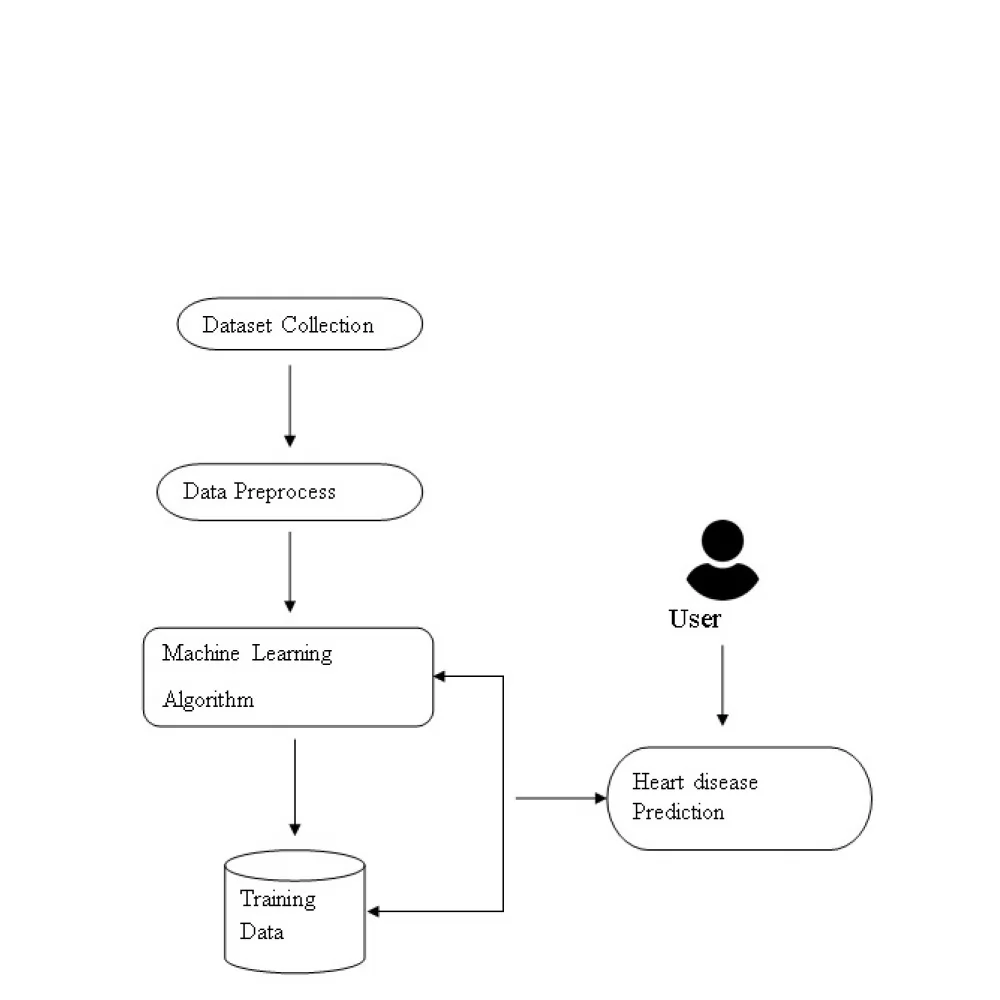
## 

## 

## 

## 10.3 Data Flow Diagrams

A data flow diagram (DFD) maps out the flow of information for any process or system. It uses defined symbols like rectangles, circles, and arrows, plus short text labels, to show data inputs, outputs, storage points, and the routes between each destination. Data flowcharts can range from simple, even hand-drawn process overviews, to in-depth, multi-level DFDs that dig progressively deeper into how the data is handled. They can be used to analyze an existing system or model a new one.



## 

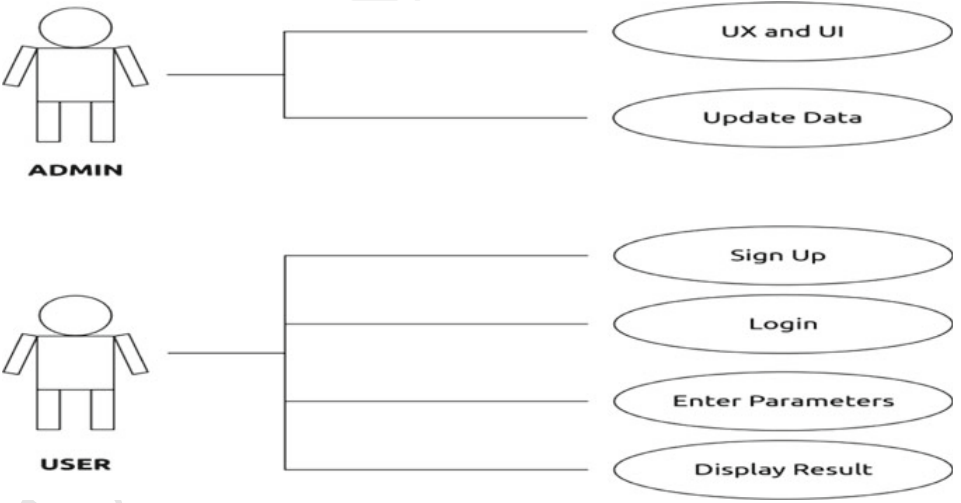
## 10.4 Use Case Diagram

In the Unified Modeling Language (UML), a use case diagram can summarize the details of your system's users (also known as actors) and their interactions with the system. To build one, you'll use a set of specialized symbols and connectors. An effective use case diagram can help your team discuss and represent:

● Scenarios in which your system or application interacts with people, organizations, or external systems.

● Goals that your system or application helps those entities (known as actors) achieve.

● The scope of your system



## 

## 10.5 Sequence Diagram

A sequence diagram is a type of interaction diagram because it describes how and in what order a group of objects works together. These diagrams are used by software developers and business professionals to understand requirements for a new system or to document an existing process. Sequence diagrams are sometimes known as event diagrams or event scenarios. Sequence diagrams can be useful references for businesses and other Organizations.

Try drawing a sequence diagram to:

● Represent the details of a UML use case.

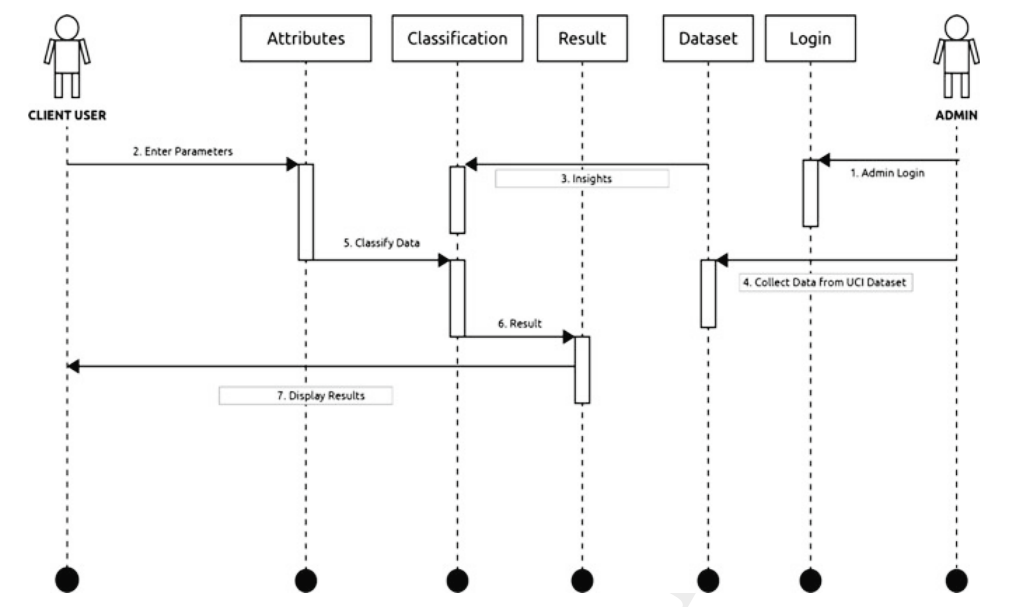
● Model the logic of a sophisticated procedure, function, or operation.

● See how objects and components interact with each other to complete a

process.

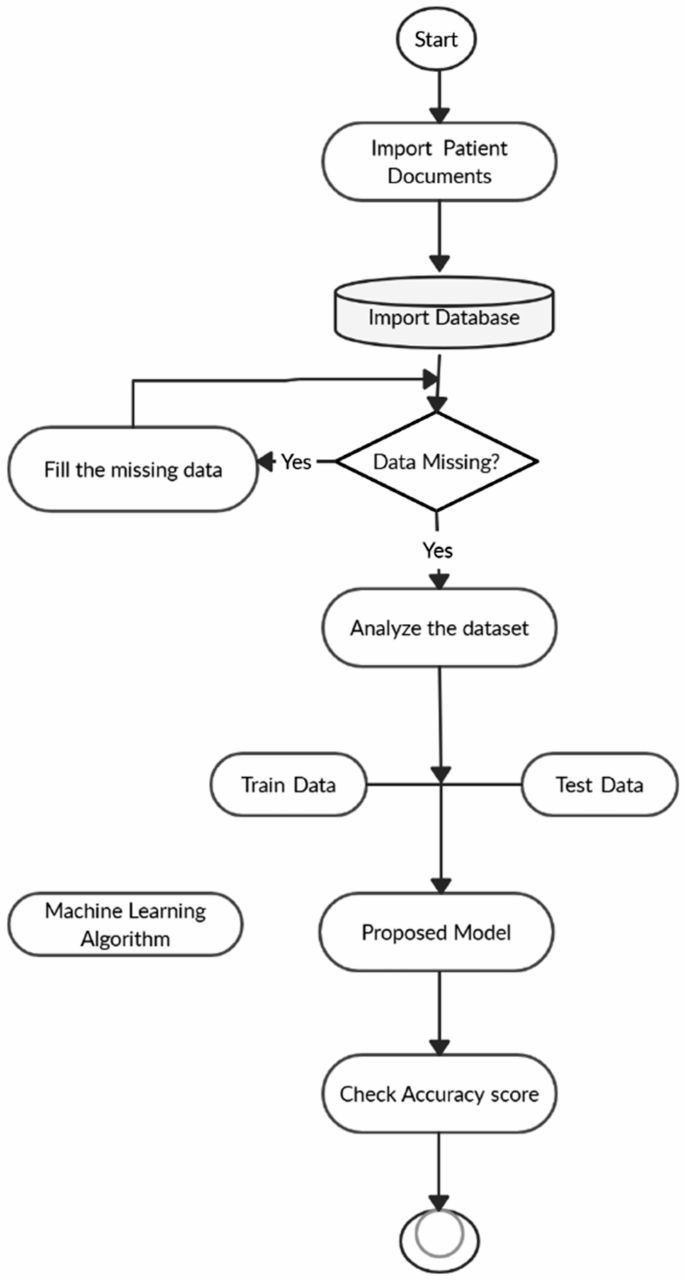
● Plan and understand the detailed functionality of an existing or future

scenario



## 10.6 Flowchart Diagram

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It can be manual, automated, or a combination of both. It shows how data enters and leaves the system, what changes the information, and where data is stored. Data Flow Data is a key and basic tool for the architecture of all other objects. Bubble-bubble or data flow graph is another name for DFD. DFDs are a model of the proposed system.



## 10.7 Block Diagram

A block diagram is a graphical representation of a system – it provides a functional view of a system. Block diagrams give us a better understanding of a system’s functions and help create interconnections within it. They are used to describe hardware and software systems as well as to represent processes.



# 11. Experimental Analysis

11.1 SAMPLE CODE:

11.1.1 Main.py

from flask import Flask,render\_template,url\_for,request

from flask\_material import Material

import matplotlib.pyplot as plt

import seaborn as sns

import pandas as pd

import numpy as np

from sklearn.model\_selection import train\_test\_split

from sklearn.preprocessing import StandardScaler

from sklearn.metrics import classification\_report

from sklearn.ensemble import RandomForestClassifier

data = pd.read\_csv('heart.csv')

X = data.iloc[:, :-1].values

y = data.iloc[:, -1].values

x\_train, x\_test, y\_train, y\_test = train\_test\_split(X,y,test\_size = 0.2, random\_state = 1)

sc = StandardScaler()

x\_train = sc.fit\_transform(x\_train)

x\_test = sc.transform(x\_test)

model = RandomForestClassifier(random\_state=1)

model.fit(x\_train, y\_train)

y\_pred = model.predict(x\_test)

app = Flask(\_\_name\_\_)

Material(app)

@app.route('/index')

def index():

return render\_template("index.html")

@app.route('/page')

def page():

return render\_template("page.html")

@app.route('/login')

def login():

return render\_template("login.html")

@app.route('/login1')

def login1():

return render\_template("login1.html")

@app.route('/single')

def single():

return render\_template("single.html")

@app.route('/upload',methods=('GET', 'POST'))

def upload():

if request.method == 'POST':

print(request.files)

dataset = request.files['datasetfile']

df = pd.read\_csv(dataset)

return render\_template("preview.html",df\_view = df)

return render\_template("upload.html")

@app.post('/preview')

def preview():

if request.method == 'POST':

dataset = request.files['datasetfile']

df = pd.read\_csv(dataset)

return render\_template("preview.html",df\_view = df)

@app.route('/analyze\_form',methods=('GET', 'POST'))

def analyze\_form():

if request.method == 'POST':

print(request.form)

age = request.form['age']

sex = request.form['sex']

cp = request.form['cp']

trestbps = request.form['trestbps']

chol = request.form['chol']

fbs = request.form['fbs']

restecg = request.form['restecg']

thalach = request.form['thalach']

exang = request.form['exang']

oldpeak = request.form['oldpeak']

slope = request.form['slope']

ca = request.form['ca']

thal = request.form['thal']

model\_choice = request.form['model\_choice']

sample\_data = [age,sex,cp,trestbps,chol,fbs,restecg,thalach,exang,oldpeak,slope,ca,thal]

clean\_data = [float(i) for i in sample\_data]

result\_prediction = model.predict(sc.transform([clean\_data]))

return render\_template('result.html',

age = age, sex = sex,cp = cp,trestbps = trestbps,

chol = chol,fbs = fbs,restecg = restecg,thalach = thalach,

exang = exang,oldpeak = oldpeak,slope = slope,ca = ca,

thal = thal,sample\_data1=sample\_data,

result\_prediction=result\_prediction,model\_selected=model\_choice)

return render\_template("analyze\_form.html")

@app.route('/analyze\_form')

def analyze():

print(request.method)

if request.method == 'POST':

age = request.form['age']

sex = request.form['sex']

cp = request.form['cp']

trestbps = request.form['trestbps']

chol = request.form['chol']

fbs = request.form['fbs']

restecg = request.form['restecg']

thalach = request.form['thalach']

exang = request.form['exang']

oldpeak = request.form['oldpeak']

slope = request.form['slope']

ca = request.form['ca']

thal = request.form['thal']

model\_choice = request.form['model\_choice']

sample\_data = [age,sex,cp,trestbps,chol,fbs,restecg,thalach,exang,oldpeak,slope,ca,thal]

clean\_data = [float(i) for i in sample\_data]

result\_prediction = model.predict(sc.transform([clean\_data]))

return render\_template('result.html',

age = age, sex = sex,cp = cp,trestbps = trestbps,

chol = chol,fbs = fbs,restecg = restecg,thalach = thalach,

exang = exang,oldpeak = oldpeak,slope = slope,ca = ca,

thal = thal,sample\_data1=sample\_data,

result\_prediction=result\_prediction,model\_selected=model\_choice)

if \_\_name\_\_ == '\_\_main\_\_':

app.run(debug=True)

**Explanation:**

* **render\_template**:It is used to generate output from a template file based on the Jinja2 engine that is found in the application's templates folder.
* **flask\_material** : into an extension that mostly consists of a blueprint named 'material'. It can also create links to serve Materialize from a CDN and works with no boilerplate code in your application.
* **matplotlib. pyplot**: It is a collection of command style functions that makes matplotlib work like MATLAB.
* **seaborn**: It is a Python data visualization library built on top of Matplotlib. It is used for making statistical graphics in python
* **pandas**: It is usually imported under the pd alias.
* **numpy**: NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays.
* **sklearn.model\_selection import train\_test\_split**: Split arrays or matrices into random train and test subsets.
* **sklearn.preprocessing import StandardScaler**: Standardize features by removing the mean and scaling to unit variance.
* **sklearn.metrics import classification\_report**: Build a text report showing the main classification metrics.
* **from sklearn.ensemble import RandomForestClassifier**: A random forest is a meta estimator that fits a number of decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting. The sub-sample size is controlled with the max\_samples parameter if bootstrap=True (default), otherwise the whole dataset is used to build each tree.

11.1.2 EasyResponsiveTabs.js:

// Easy Responsive Tabs Plugin

(function ($) {

$.fn.extend({

easyResponsiveTabs: function (options) {

//Set the default values, use comma to separate the settings, example:

var defaults = {

type: 'default', //default, vertical, accordion;

width: 'auto',

fit: true

}

//Variables

var options = $.extend(defaults, options);

var opt = options, jtype = opt.type, jfit = opt.fit, jwidth = opt.width, vtabs = 'vertical', accord = 'accordion';

//Main function

this.each(function () {

var $respTabs = $(this);

$respTabs.find('ul.resp-tabs-list li').addClass('resp-tab-item');

$respTabs.css({

'display': 'block',

'width': jwidth

});

$respTabs.find('.resp-tabs-container > div').addClass('resp-tab-content');

jtab\_options();

//Properties Function

function jtab\_options() {

if (jtype == vtabs) {

$respTabs.addClass('resp-vtabs');

}

if (jfit == true) {

$respTabs.css({ width: '100%', margin: '0px' });

}

if (jtype == accord) {

$respTabs.addClass('resp-easy-accordion');

$respTabs.find('.resp-tabs-list').css('display', 'none');

}

}

//Assigning the h2 markup

var $tabItemh2;

$respTabs.find('.resp-tab-content').before("<h2 class='resp-accordion' role='tab'><span class='resp-arrow'></span></h2>");

var itemCount = 0;

$respTabs.find('.resp-accordion').each(function () {

$tabItemh2 = $(this);

var innertext = $respTabs.find('.resp-tab-item:eq(' + itemCount + ')').text();

$respTabs.find('.resp-accordion:eq(' + itemCount + ')').append(innertext);

$tabItemh2.attr('aria-controls', 'tab\_item-' + (itemCount));

itemCount++;

});

//Assigning the 'aria-controls' to Tab items

var count = 0,

$tabContent;

$respTabs.find('.resp-tab-item').each(function () {

$tabItem = $(this);

$tabItem.attr('aria-controls', 'tab\_item-' + (count));

$tabItem.attr('role', 'tab');

//First active tab

$respTabs.find('.resp-tab-item').first().addClass('resp-tab-active');

$respTabs.find('.resp-accordion').first().addClass('resp-tab-active');

$respTabs.find('.resp-tab-content').first().addClass('resp-tab-content-active').attr('style', 'display:block');

//Assigning the 'aria-labelledby' attr to tab-content

var tabcount = 0;

$respTabs.find('.resp-tab-content').each(function () {

$tabContent = $(this);

$tabContent.attr('aria-labelledby', 'tab\_item-' + (tabcount));

tabcount++;

});

count++;

});

//Tab Click action function

$respTabs.find("[role=tab]").each(function () {

var $currentTab = $(this);

$currentTab.click(function () {

var $tabAria = $currentTab.attr('aria-controls');

if ($currentTab.hasClass('resp-accordion') && $currentTab.hasClass('resp-tab-active')) {

$respTabs.find('.resp-tab-content-active').slideUp('', function () { $(this).addClass('resp-accordion-closed'); });

$currentTab.removeClass('resp-tab-active');

return false;

}

if (!$currentTab.hasClass('resp-tab-active') && $currentTab.hasClass('resp-accordion')) {

$respTabs.find('.resp-tab-active').removeClass('resp-tab-active');

$respTabs.find('.resp-tab-content-active').slideUp().removeClass('resp-tab-content-active resp-accordion-closed');

$respTabs.find("[aria-controls=" + $tabAria + "]").addClass('resp-tab-active');

$respTabs.find('.resp-tab-content[aria-labelledby = ' + $tabAria + ']').slideDown().addClass('resp-tab-content-active');

} else {

$respTabs.find('.resp-tab-active').removeClass('resp-tab-active');

$respTabs.find('.resp-tab-content-active').removeAttr('style').removeClass('resp-tab-content-active').removeClass('resp-accordion-closed');

$respTabs.find("[aria-controls=" + $tabAria + "]").addClass('resp-tab-active');

$respTabs.find('.resp-tab-content[aria-labelledby = ' + $tabAria + ']').addClass('resp-tab-content-active').attr('style', 'display:block');

}

});

//Window resize function

$(window).resize(function () {

$respTabs.find('.resp-accordion-closed').removeAttr('style');

});

});

});

}

});

})(jQuery);

11.1.3 JQuery.Slider.js:

/\*!

\* jQuery wmuSlider v2.1

\*

\* Copyright (c) 2011 Brice Lechatellier

\* http://brice.lechatellier.com/

\*

\* Licensed under the MIT license: http://www.opensource.org/licenses/mit-license.php

\*/

;(function($) {

$.fn.wmuSlider = function(options) {

/\* Default Options

================================================== \*/

var defaults = {

animation: 'fade',

animationDuration: 600,

slideshow: true,

slideshowSpeed: 7000,

slideToStart: 0,

navigationControl: true,

paginationControl: true,

previousText: 'Previous',

nextText: 'Next',

touch: false,

slide: 'article',

items: 1

};

var options = $.extend(defaults, options);

return this.each(function() {

/\* Variables

================================================== \*/

var $this = $(this);

var currentIndex = options.slideToStart;

var wrapper = $this.find('.wmuSliderWrapper');

var slides = $this.find(options.slide);

var slidesCount = slides.length;

var slideshowTimeout;

var paginationControl;

var isAnimating;

/\* Load Slide

================================================== \*/

var loadSlide = function(index, infinite, touch) {

if (isAnimating) {

return false;

}

isAnimating = true;

currentIndex = index;

var slide = $(slides[index]);

$this.animate({ height: slide.innerHeight() });

if (options.animation == 'fade') {

slides.css({

position: 'absolute',

opacity: 0

});

slide.css('position', 'relative');

slide.animate({ opacity:1 }, options.animationDuration, function() {

isAnimating = false;

});

} else if (options.animation == 'slide') {

if (!infinite) {

wrapper.animate({ marginLeft: -$this.width() / options.items \* index }, options.animationDuration, function() {

isAnimating = false;

});

} else {

if (index == 0) {

wrapper.animate({ marginLeft: -$this.width() / options.items \* slidesCount }, options.animationDuration, function() {

wrapper.css('marginLeft', 0);

isAnimating = false;

});

} else {

if (!touch) {

wrapper.css('marginLeft', -$this.width() / options.items \* slidesCount);

}

wrapper.animate({ marginLeft: -$this.width() / options.items \* index }, options.animationDuration, function() {

isAnimating = false;

});

}

}

}

if (paginationControl) {

paginationControl.find('a').each(function(i) {

if(i == index) {

$(this).addClass('wmuActive');

} else {

$(this).removeClass('wmuActive');

}

});

}

// Trigger Event

$this.trigger('slideLoaded', index);

};

/\* Navigation Control

================================================== \*/

if (options.navigationControl) {

var prev = $('<a class="wmuSliderPrev">' + options.previousText + '</a>');

prev.click(function(e) {

e.preventDefault();

clearTimeout(slideshowTimeout);

if (currentIndex == 0) {

loadSlide(slidesCount - 1, true);

} else {

loadSlide(currentIndex - 1);

}

});

$this.append(prev);

var next = $('<a class="wmuSliderNext">' + options.nextText + '</a>');

next.click(function(e) {

e.preventDefault();

clearTimeout(slideshowTimeout);

if (currentIndex + 1 == slidesCount) {

loadSlide(0, true);

} else {

loadSlide(currentIndex + 1);

}

});

$this.append(next);

}

/\* Pagination Control

================================================== \*/

if (options.paginationControl) {

paginationControl = $('<ul class="wmuSliderPagination"></ul>');

$.each(slides, function(i) {

paginationControl.append('<li><a href="#">' + i + '</a></li>');

paginationControl.find('a:eq(' + i + ')').click(function(e) {

e.preventDefault();

clearTimeout(slideshowTimeout);

loadSlide(i);

});

});

$this.append(paginationControl);

}

/\* Slideshow

================================================== \*/

if (options.slideshow) {

var slideshow = function() {

if (currentIndex + 1 < slidesCount) {

loadSlide(currentIndex + 1);

} else {

loadSlide(0, true);

}

slideshowTimeout = setTimeout(slideshow, options.slideshowSpeed);

}

slideshowTimeout = setTimeout(slideshow, options.slideshowSpeed);

}

/\* Resize Slider

================================================== \*/

var resize = function() {

var slide = $(slides[currentIndex]);

$this.animate({ height: slide.innerHeight() });

if (options.animation == 'slide') {

slides.css({

width: $this.width() / options.items

});

wrapper.css({

marginLeft: -$this.width() / options.items \* currentIndex,

width: $this.width() \* slides.length

});

}

};

/\* Touch

================================================== \*/

var touchSwipe = function(event, phase, direction, distance) {

clearTimeout(slideshowTimeout);

if(phase == 'move' && (direction == 'left' || direction == 'right')) {

if (direction == 'right') {

if (currentIndex == 0) {

wrapper.css('marginLeft', (-slidesCount \* $this.width() / options.items) + distance);

} else {

wrapper.css('marginLeft', (-currentIndex \* $this.width() / options.items) + distance);

}

} else if (direction == 'left') {

wrapper.css('marginLeft', (-currentIndex \* $this.width() / options.items) - distance);

}

} else if (phase == 'cancel' ) {

if (direction == 'right' && currentIndex == 0) {

wrapper.animate({ marginLeft: -slidesCount \* $this.width() / options.items }, options.animationDuration);

} else {

wrapper.animate({ marginLeft: -currentIndex \* $this.width() / options.items }, options.animationDuration);

}

} else if (phase == 'end' ) {

if (direction == 'right') {

if (currentIndex == 0) {

loadSlide(slidesCount - 1, true, true);

} else {

loadSlide(currentIndex - 1);

}

} else if (direction == 'left') {

if (currentIndex + 1 == slidesCount) {

loadSlide(0, true);

} else {

loadSlide(currentIndex + 1);

}

} else {

wrapper.animate({ marginLeft: -currentIndex \* $this.width() / options.items }, options.animationDuration);

}

}

};

if (options.touch && options.animation == 'slide') {

if (!$.isFunction($.fn.swipe)) {

$.ajax({

url: 'jquery.touchSwipe.min.js',

async: false

});

}

if ($.isFunction($.fn.swipe)) {

$this.swipe({ triggerOnTouchEnd:false, swipeStatus:touchSwipe, allowPageScroll:'vertical' });

}

}

/\* Init Slider

================================================== \*/

var init = function() {

var slide = $(slides[currentIndex]);

var img = slide.find('img');

img.load(function() {

wrapper.show();

$this.animate({ height: slide.innerHeight() });

});

if (options.animation == 'fade') {

slides.css({

position: 'absolute',

width: '100%',

opacity: 0

});

$(slides[currentIndex]).css('position', 'relative');

} else if (options.animation == 'slide') {

if (options.items > slidesCount) {

options.items = slidesCount;

}

slides.css('float', 'left');

slides.each(function(i){

var slide = $(this);

slide.attr('data-index', i);

});

for(var i = 0; i < options.items; i++) {

wrapper.append($(slides[i]).clone());

}

slides = $this.find(options.slide);

}

resize();

$this.trigger('hasLoaded');

loadSlide(currentIndex);

}

init();

/\* Bind Events

================================================== \*/

// Resize

$(window).resize(resize);

// Load Slide

$this.bind('loadSlide', function(e, i) {

clearTimeout(slideshowTimeout);

loadSlide(i);

});

});

}

})(jQuery);

11.1.4 SwipeBox.css:

.swipebox-overflow-hidden {

overflow: hidden!important;

}

#swipebox-overlay img {

border: none!important;

}

#swipebox-overlay {

width: 100%!important;

height: 100%;

position: fixed;

top: 0;

left: 0;

opacity: 0;

z-index: 9999;

overflow: hidden;

display: none;

-webkit-user-select: none;

-moz-user-select: none;

user-select: none;

}

#swipebox-slider {

height: 100%;

left: 0;

top: 0;

width: 100%;

white-space: nowrap;

position: absolute;

}

#swipebox-slider .slide {

background: url(../images/loader.gif) no-repeat center center;

height: 100%;

line-height: 1px;

text-align: center;

width: 100%;

display: inline-block;

}

#swipebox-slider .slide:before {

content: "";

display: inline-block;

height: 50%;

width: 1px;

margin-right: -1px;

}

#swipebox-slider .slide img {

display: inline-block;

max-height: 100%;

max-width: 100%;

width: auto;

height: auto;

vertical-align: middle;

}

#swipebox-action, #swipebox-caption {

position: absolute;

left: 0;

z-index: 999;

height: 50px;

width: 100%;

}

#swipebox-action {

bottom: -50px;

}

#swipebox-action.visible-bars {

bottom: 0;

}

#swipebox-action.force-visible-bars {

bottom: 0!important;

}

#swipebox-caption {

top: -50px;

text-align: center;

}

#swipebox-caption.visible-bars {

top: 0;

}

#swipebox-caption.force-visible-bars {

top: 0!important;

}

#swipebox-action #swipebox-prev, #swipebox-action #swipebox-next,

#swipebox-action #swipebox-close {

background-image: url(../images/icons.png);

background-repeat: no-repeat;

border: none!important;

text-decoration: none!important;

cursor: pointer;

position: absolute;

width: 50px;

height: 50px;

top: 0;

}

#swipebox-action #swipebox-close {

background-position: 15px 12px;

left: 40px;

}

#swipebox-action #swipebox-prev {

background-position: -32px 13px;

right: 100px;

}

#swipebox-action #swipebox-next {

background-position: -78px 13px;

right: 40px;

}

#swipebox-action #swipebox-prev.disabled,

#swipebox-action #swipebox-next.disabled {

filter: progid:DXImageTransform.Microsoft.Alpha(Opacity=30);

opacity: 0.3;

}

#swipebox-slider.rightSpring {

-moz-animation: rightSpring 0.3s;

-webkit-animation: rightSpring 0.3s;

}

#swipebox-slider.leftSpring {

-moz-animation: leftSpring 0.3s;

-webkit-animation: leftSpring 0.3s;

}

@-moz-keyframes rightSpring {

0% {

margin-left: 0px;

}

50% {

margin-left: -30px;

}

100% {

margin-left: 0px;

}

}

@-moz-keyframes leftSpring {

0% {

margin-left: 0px;

}

50% {

margin-left: 30px;

}

100% {

margin-left: 0px;

}

}

@-webkit-keyframes rightSpring {

0% {

margin-left: 0px;

}

50% {

margin-left: -30px;

}

100% {

margin-left: 0px;

}

}

@-webkit-keyframes leftSpring {

0% {

margin-left: 0px;

}

50% {

margin-left: 30px;

}

100% {

margin-left: 0px;

}

}

/\* Skin

--------------------------\*/

#swipebox-overlay {

background:rgba(13, 13, 13, 0.88);

}

#swipebox-action, #swipebox-caption {

text-shadow: 1px 1px 1px black;

background-color: #0d0d0d;

background-image: -webkit-gradient(linear, 50% 0%, 50% 100%, color-stop(0%, #0d0d0d), color-stop(100%, #000000));

background-image: -webkit-linear-gradient(#0d0d0d, #000000);

background-image: -moz-linear-gradient(#0d0d0d, #000000);

background-image: -o-linear-gradient(#0d0d0d, #000000);

background-image: linear-gradient(#0d0d0d, #000000);

-webkit-box-shadow: 0 1px 1px 1px #212121, inset 0 1px 1px 1px black;

-moz-box-shadow: 0 1px 1px 1px #212121, inset 0 1px 1px 1px black;

box-shadow: 0 1px 1px 1px #212121, inset 0 1px 1px 1px black;

filter: progid:DXImageTransform.Microsoft.Alpha(Opacity=95);

opacity: 0.95;

}

#swipebox-action {

-webkit-box-shadow: 0 -1px -1px 1px #212121, inset 0 -1px -1px 1px black;

-moz-box-shadow: 0 -1px -1px 1px #212121, inset 0 -1px -1px 1px black;

box-shadow: 0 -1px -1px 1px #212121, inset 0 -1px -1px 1px black;

}

#swipebox-caption {

color: white!important;

font-size: 15px;

line-height: 43px;

font-family: Helvetica, Arial, sans-serif;

}

@media(max-width:768px){

#swipebox-slider .slide img {

display: inline-block;

max-height: 100%;

max-width: 91%;

}

}

11.1.5 Analyze\_form.html

<!DOCTYPE html>

<html>

<head>

<title>Heart Disease Prediction </title>

<link rel = "icon" href = "au.png" type = "image/x-icon">

<link href="bootstrap.css" rel="stylesheet" type="text/css" media="all" />

<!-- jQuery (necessary for Bootstrap's JavaScript plugins) -->

<script src="jquery.min.js"></script>

<!-- Custom Theme files -->

<!--theme-style-->

<link href="style.css" rel="stylesheet" type="text/css" media="all" />

<!--//theme-style-->

<meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1">

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<meta name="keywords" content="Open Heart Responsive web template, Bootstrap Web Templates, Flat Web Templates, Android Compatible web template,

Smartphone Compatible web template, free webdesigns for Nokia, Samsung, LG, SonyEricsson, Motorola web design" />

<script type="application/x-javascript"> addEventListener("load", function() { setTimeout(hideURLbar, 0); }, false); function hideURLbar(){ window.scrollTo(0,1); } </script>

<!--fonts-->

<link href='//fonts.googleapis.com/css?family=Open+Sans+Condensed:300,300italic,700' rel='stylesheet' type='text/css'>

<!--//fonts-->

</head>

<body>

<!--header-->

<div class="banner-in">

<div class="container">

<h6>HOME / <span>ANALYZE</span></h6>

</div>

</div>

<!--header-->

<div class="header">

<div class="container">

<div class="logo">

<a href="index"><img src="logo.png" alt=" " width="150px" height="150px"/></a>

</div>

<div class="top-nav">

<span class="menu"> </span>

<ul>

<li><a href="index" class="scroll">HOME </a></li>

<li><a href="login" class="scroll">ADMIN LOGIN </a></li>

<li><a href="upload" class="scroll">UPLOAD</a></li>

<!--li><a href="{{url\_for('preview')}}" class="scroll">DATA-VIEW</a></li-->

<li class="active"><a href="analyze\_form" class="scroll">ANALYZE</a></li>

<!--li><a href="contact.html" class="scroll">RESULT</a></li-->

</ul>

<!--script-->

<script>

$("span.menu").click(function(){

$(".top-nav ul").slideToggle(500, function(){

});

});

</script>

</div>

<div class="clearfix"> </div>

</div>

</div>

<!---->

<!--content-->

<div class="content">

<div class="container">

<div class="blog">

<h3>ANALYZE</h3>

<form action="analyze\_form" method="POST" oninput="output.value = oldpeak.valueAsNumber,output2.value = ca.valueAsNumber">

<div class="form-group">

<label for="exampleFormControlInput1">Age</label>

<input type="number" class="form-control" id="age" name="age" placeholder="Age">

</div>

<label for="radio">Gender</label>

<div class="form-check">

<input class="form-check-input" type="radio" name="sex" id="sex" value="1" checked>

<label class="form-check-label" for="exampleRadios1">

Male

</label>

</div>

<div class="form-check">

<input class="form-check-input" type="radio" name="sex" id="sex" value="0">

<label class="form-check-label" for="exampleRadios2">

Female

</label>

</div>

<div class="form-group">

<label for="exampleFormControlSelect1">Chest Pain Type</label>

<select class="form-control" id="cp" name="cp">

<option value="" disabled selected>Select Chest Pain Type</option>

<option value="0">Typical Angina(0)</option>

<option value="1">Atypical Angina(1)</option>

<option value="2">Non-anginal Pain(2)</option>

<option value="3">Asymptomatic(3)</option>

</select>

</div>

<div class="form-group">

<label for="exampleFormControlInput1">Resting BP(mm Hg)</label>

<input type="number" class="form-control" id="trestbps" name="trestbps" placeholder="Resting BP">

</div>

<div class="form-group">

<label for="exampleFormControlInput1">Cholestrol(mg/dl)</label>

<input type="number" class="form-control" id="chol" name="chol" placeholder="Cholestrol">

</div>

<label for="radio">Is Fasting Blood Sugar > 120 mg/dl(FBS)</label>

<div class="form-check">

<input class="form-check-input" type="radio" id="fbs" name="fbs" value="1" checked>

<label class="form-check-label" for="exampleRadios1">

Yes

</label>

</div>

<div class="form-check">

<input class="form-check-input" type="radio" id="fbs" name="fbs" value="0">

<label class="form-check-label" for="exampleRadios2">

No

</label>

</div>

<div class="form-group">

<label for="exampleFormControlSelect1">Resting ECG</label>

<select class="form-control" id="restecg" name="restecg">

<option value="" disabled selected>Select</option>

<option value="0">Normal(0)</option>

<option value="1">ST-T(1)</option>

<option value="2">Probable(2)</option>

</select>

</div>

<div class="form-group">

<label for="exampleFormControlInput1">Max Heart Rate Achieved</label>

<input type="number" class="form-control" id="thalach" name="thalach" placeholder="HRA">

</div>

<label for="radio">Exercise Induced Angina</label>

<div class="form-check">

<input class="form-check-input" type="radio" id="exang" name="exang" value="1" checked>

<label class="form-check-label" for="exampleRadios1">

Yes

</label>

</div>

<div class="form-check">

<input class="form-check-input" type="radio" id="exang" name="exang" value="0">

<label class="form-check-label" for="exampleRadios2">

No

</label>

</div>

<div class="form-group">

<label for="formControlRange">Old Peak</label>

<div class="range">

<input type="range" id="oldpeak" name="oldpeak" min="0" max="5" value="0" step="0.1">

<div class="range-output">

<output class="output" name="output" for="range">

0.0

</output>

</div>

</div>

<div class="form-group">

<label for="exampleFormControlSelect1">Slope</label>

<select class="form-control" id="slope" name="slope">

<option value="" disabled selected>Select</option>

<option value="0">Nothing(0)</option>

<option value="1">Upsloping(1)</option>

<option value="2">Flat(2)</option>

<option value="2">Downsloping(3)</option>

</select>

</div>

<div class="form-group">

<label for="formControlRange">No. of Major Vessels</label>

<div class="range">

<input type="range" id="ca" name="ca" min="0" max="5" value="1" step="1">

<div class="range-output">

<output class="output" name="output2" for="range">

0.0

</output>

</div>

</div>

<div class="form-group">

<label for="exampleFormControlSelect1">Thalassemia</label>

<select class="form-control" id="thal" name="thal">

<option value="" disabled selected>Select</option>

<option value="1">Normal(1)</option>

<option value="2">Fixed(2)</option>

<option value="3">Reversable(3)</option>

</select>

</div>

<div class="form-group">

<label for="exampleFormControlSelect1">Select Algorithm</label>

<select class="form-control" id="role" name="model\_choice">

<option value="" disabled selected>Select Model</option>

<option value="rfmodel">Random Forest</option>

</select>

</div>

<center><input type="submit" value="predict" class="btn btn-small black waves-light btn-extend">

<input type="reset" value="Clear" class="btn btn-small black waves-light btn-extend">

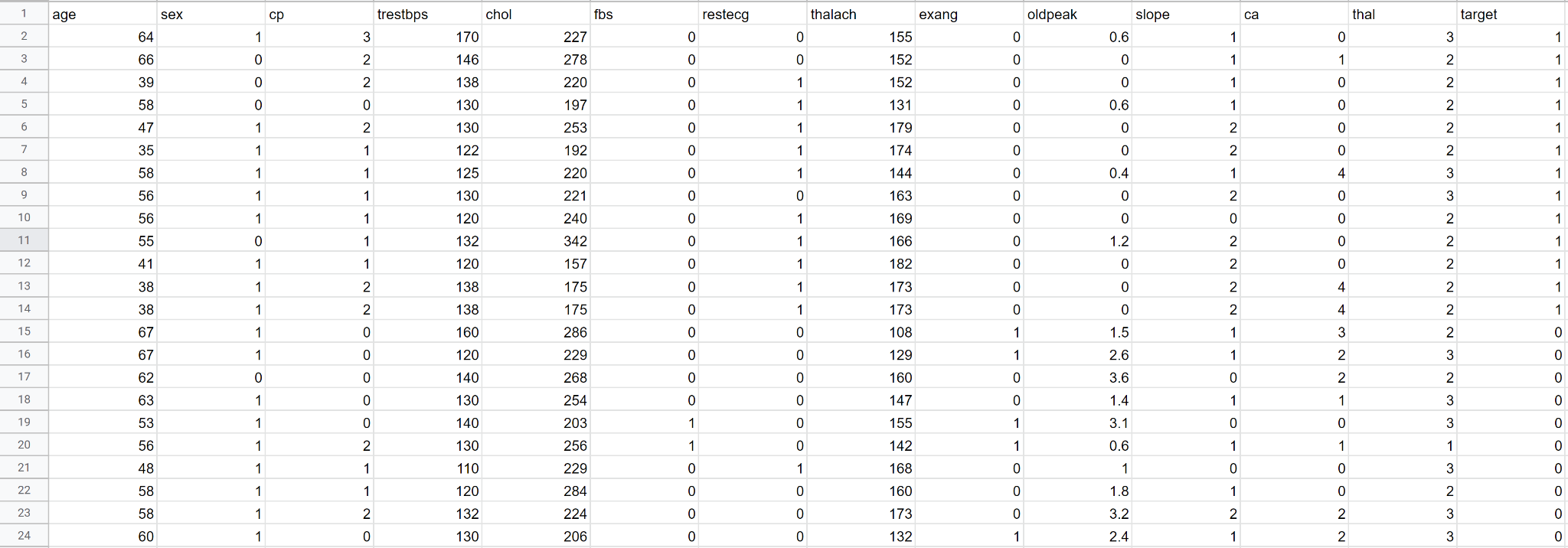
</center>

</form>

</body>

</html>

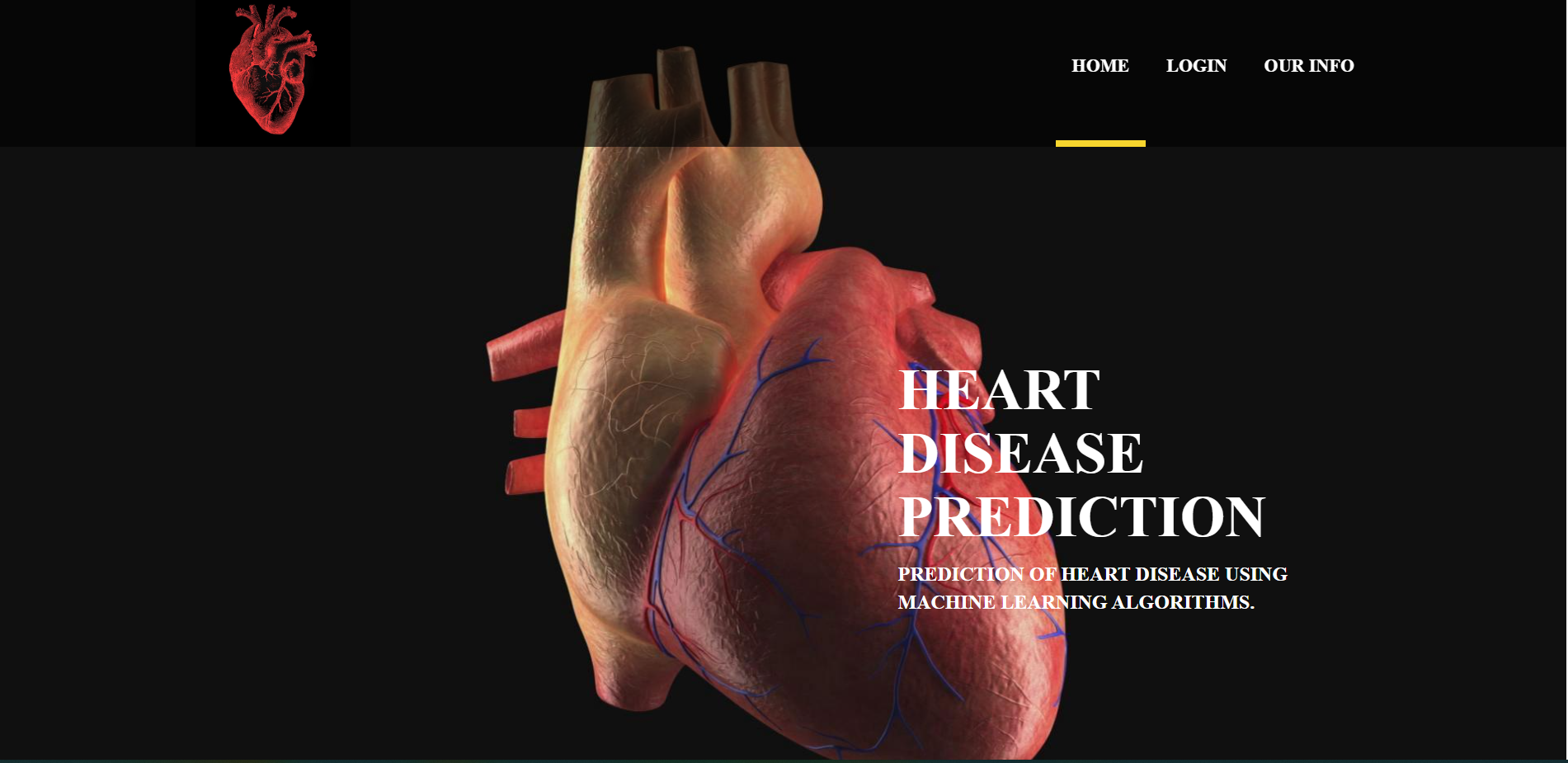
11.2 Sample Dataset:



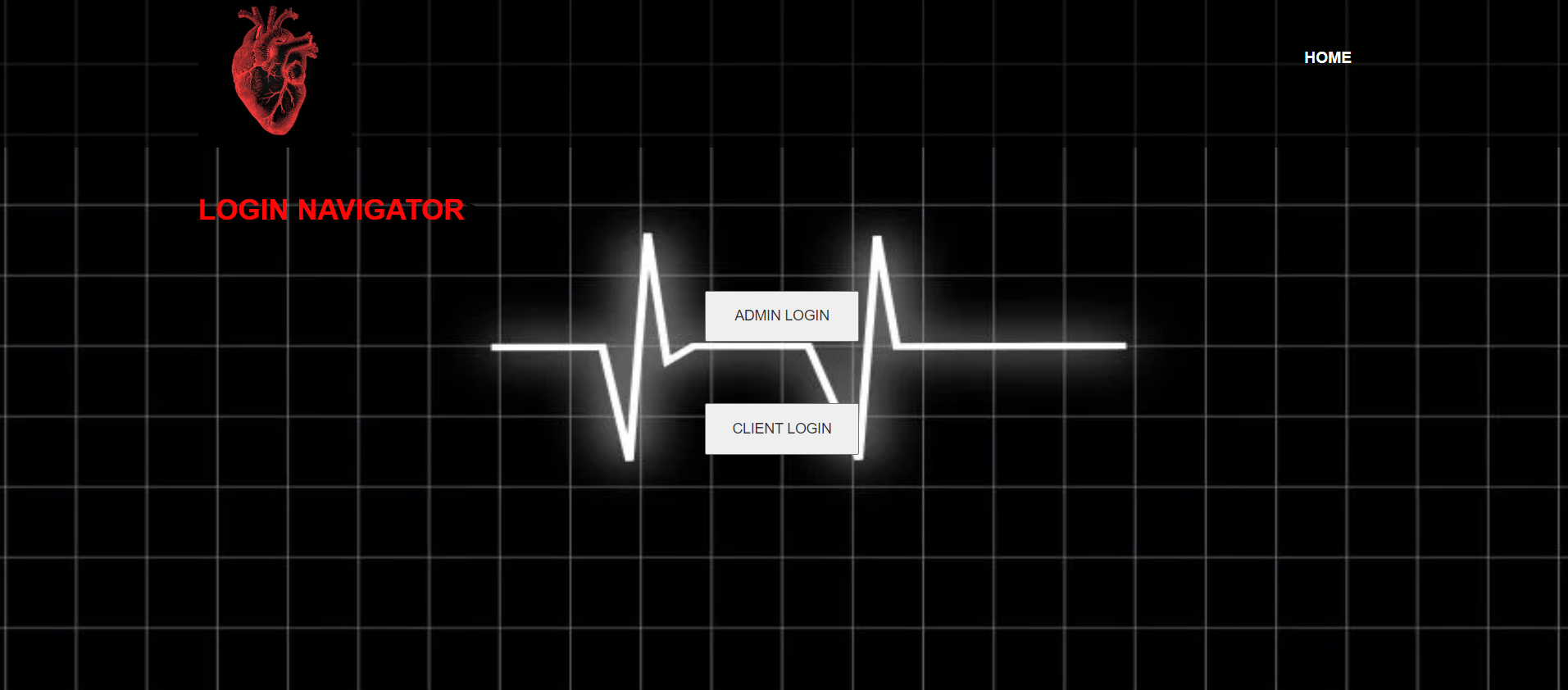
**Input dataset attributes:**

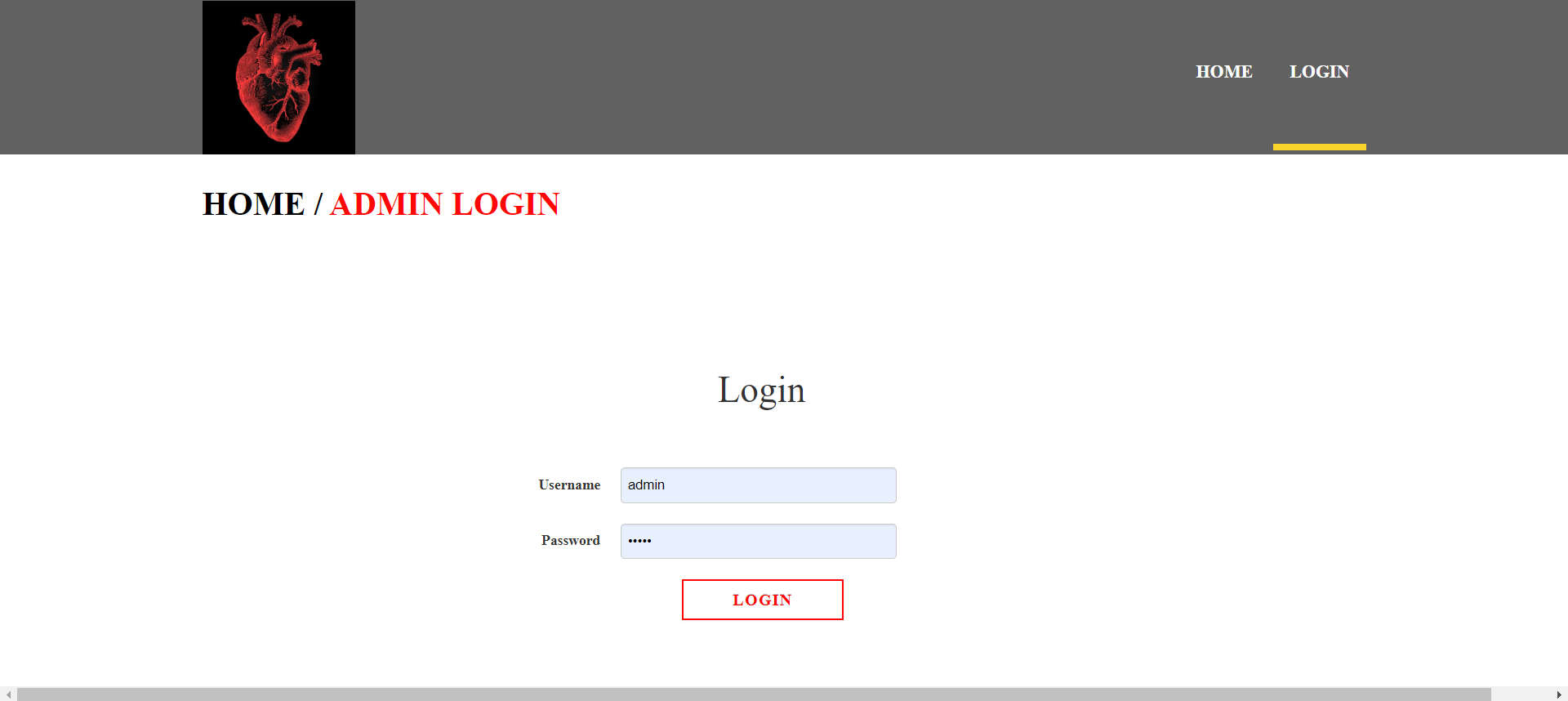
* Age in years
* sex - Gender (value 1: Male; value 0 : Female)
* cp-Chest Pain Type (value 0: Typical angina, value 1: Atypical angina, value 2: non-anginal pain; value 3: asymptomatic)
* trestbps-Resting Blood Pressure(mm Hg)
* chol-Cholesterol(mg/dl)
* fbs-Fasting Blood Sugar (value 1: > 120 mg/dl; value 0:< 120 mg/dl)
* restecg-Resting ECG(value 0:Normal, value 1:ST-T, value 2:Probable)
* thalach – Maximum Heart rate achieved
* exang – exercise induced angina (value 1: yes; value 0: no)
* OldPeak(Value:0-5)
* Slope(value 0: Nothing, value 1:Upsloping, value 2:Flat, value 3: Downsloping)
* ca - No.of Major Vessels (Value:0-5)
* Thal - Thalassemia (value 1: normal; value 2: fixed defect; value 3:reversible defect)

11.3 HOME PAGE

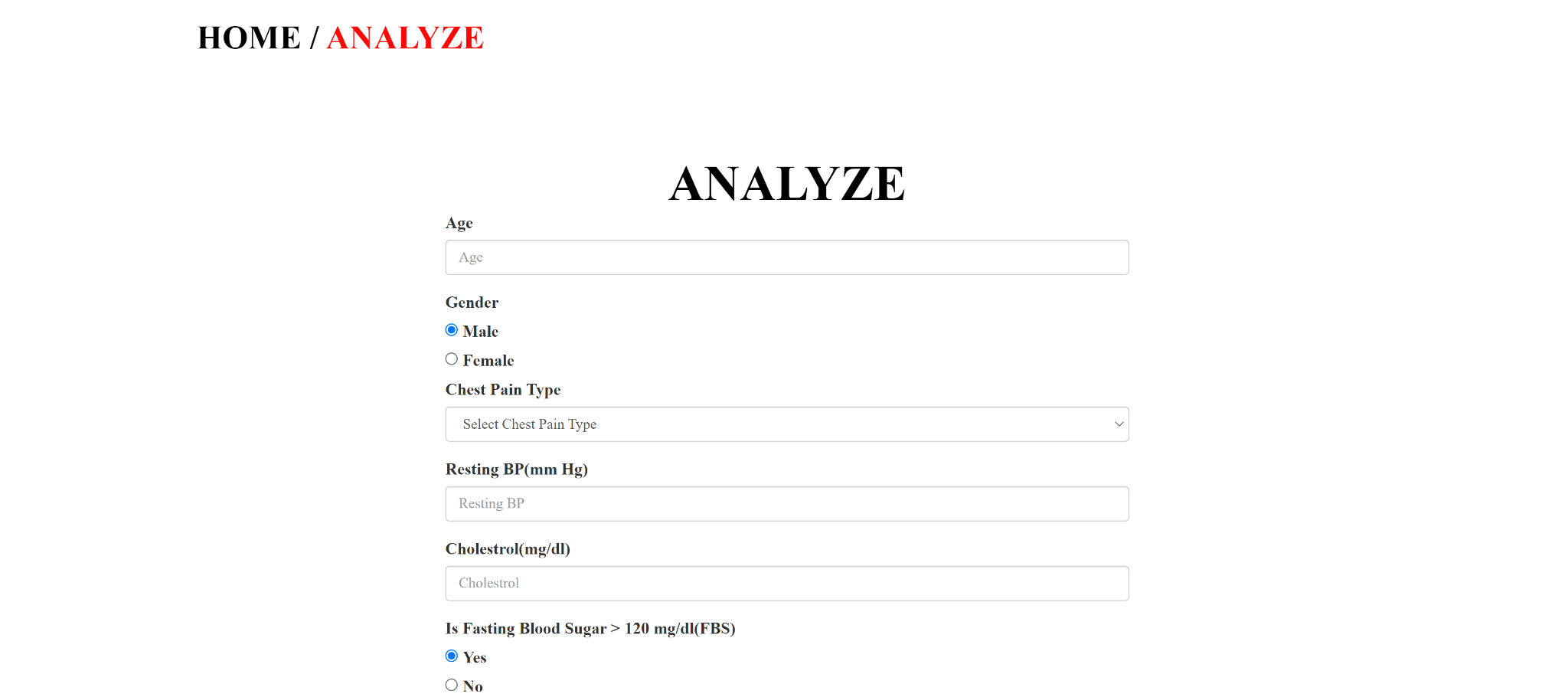


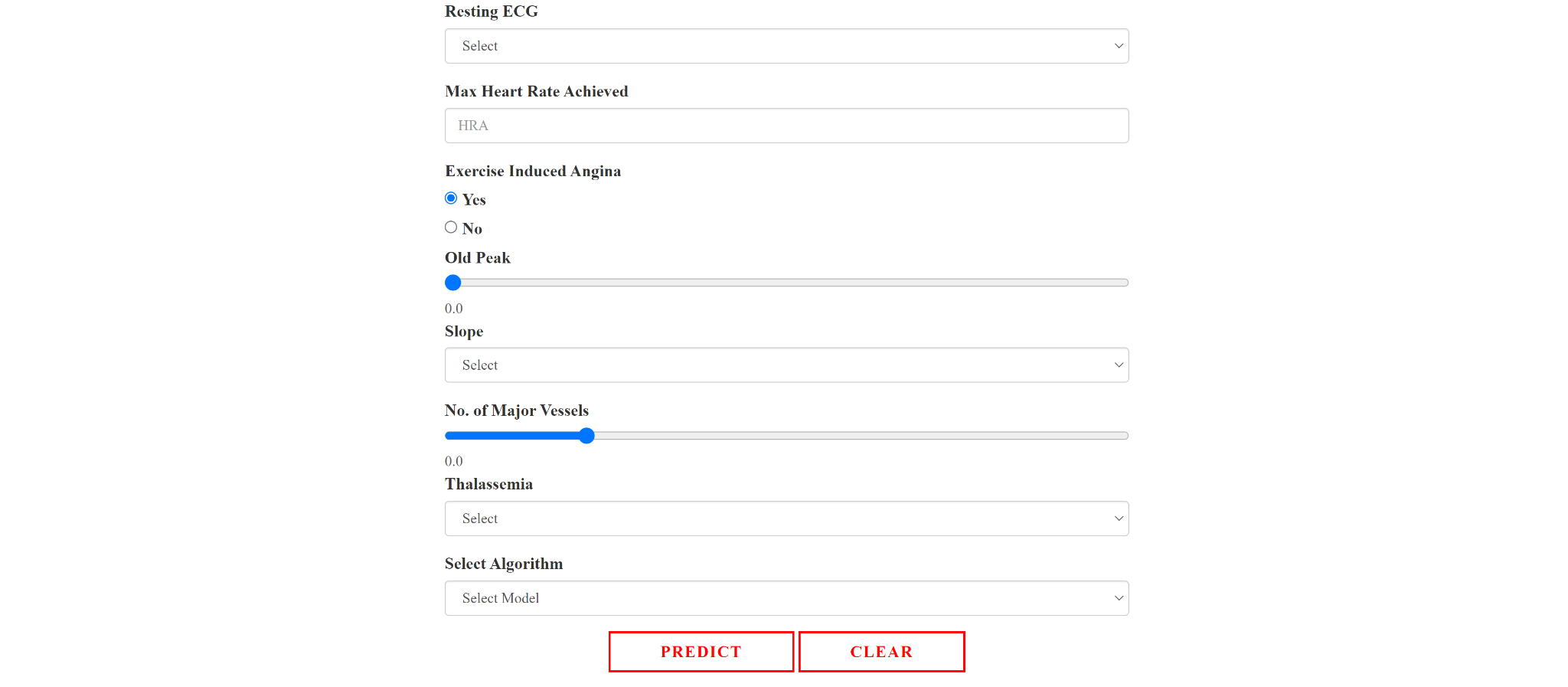
11.4 LOGIN PAGE



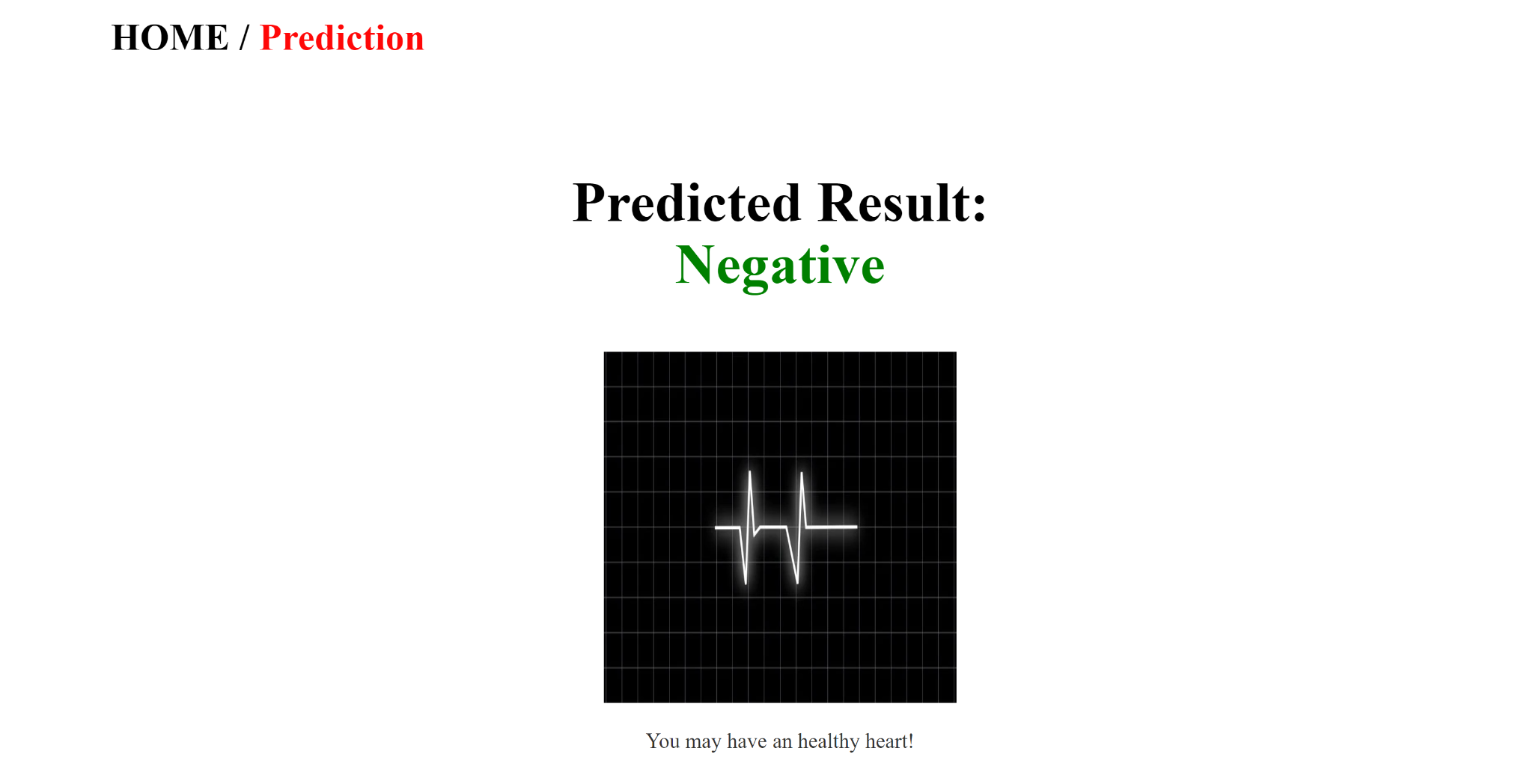


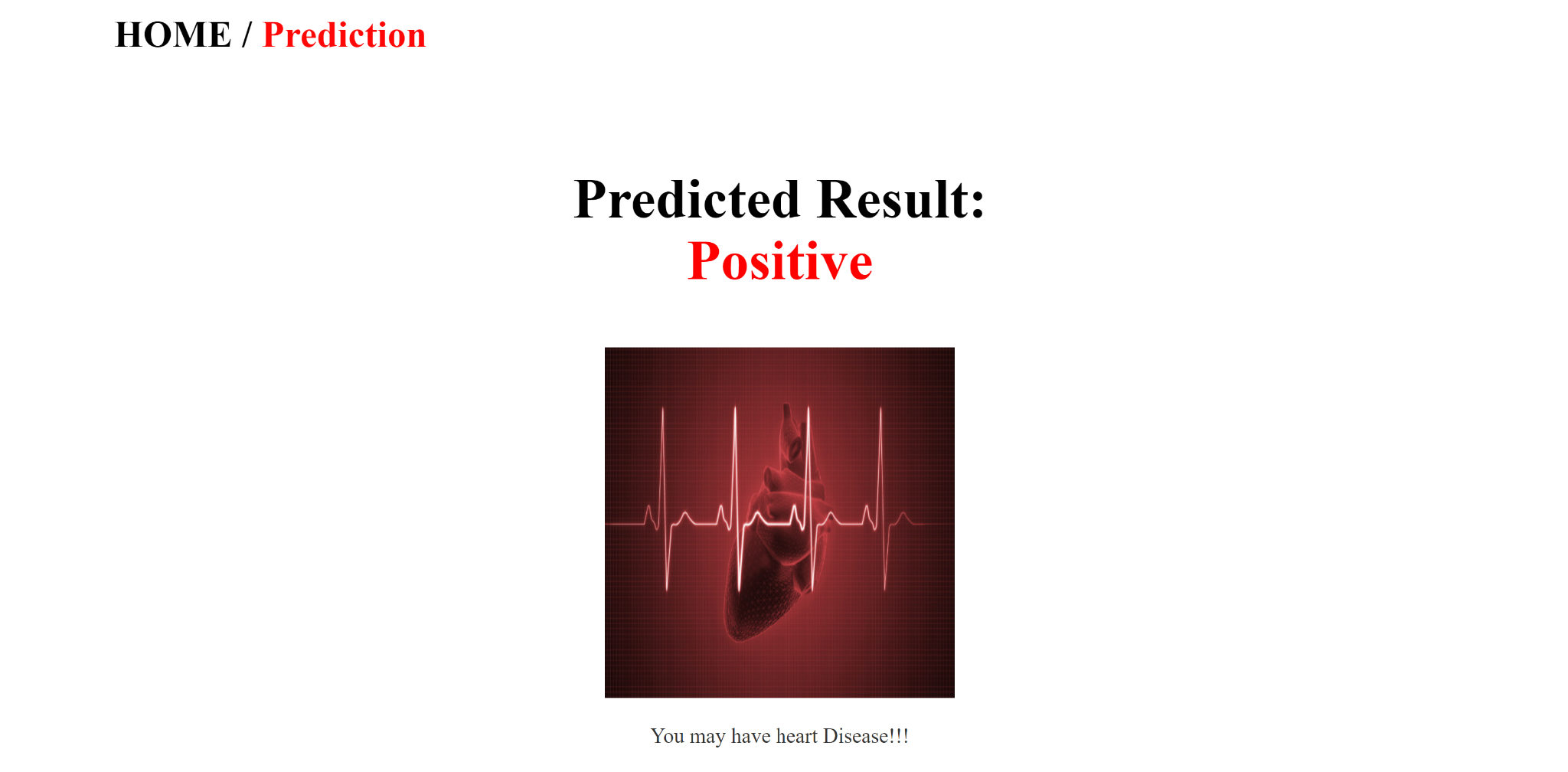
11.5 Input:

****

****

11.6 Output:

****



# 

**12. Conclusion**

Heart diseases are a major killer in India and throughout the world. The application of promising technology like machine learning to the initial prediction of heart diseases will have a profound impact on society. The early prognosis of heart disease can aid in making decisions on lifestyle changes in high-risk patients and in turn reduce the complications, which can be a great milestone in the field of medicine. The number of people facing heart disease is on a raise each year. This prompts its early diagnosis and treatment.

We have compared the accuracy score of Decision Tree, Logistic Regression, Random Forest, and Naive Bayes algorithms for predicting heart disease using the dataset. The results indicate that the Random Forest algorithm is the most efficient algorithm with an accuracy score of 90.16% for the prediction of heart disease. In the future, the work can be enhanced by using a larger dataset as compared to the one used in this analysis which will help to provide better results and help health professionals in predicting heart disease effectively and efficiently.

**13. Future Scope Of Development**

* Heart Disease has been a centipede for ages. It is very dangerous and the treatment costs lavishly.
* This could be a burden to regular people. As we know, rooting a wildling beforehand would be easier when compared to a rotten tree.
* So, this project can be further developed to make things handy to the needy. Taking the main cartilage of the project and adding features such as assigning a doctor to a patient who’s got a positive prediction.
* The prediction could further be classified and prediction can be screwy for better results.
* Better algorithms with better precision could be added for the best results. Other heart-related problems could be predicted.
* Connecting it with an IoT could help in continuous monitoring.
* The patient log could be maintained and checked frequently. The prediction could be further extended to other organs like kidney, liver and so on.