# A PROJECT REPORT

**ON**

# PREDICTION OF HEART DISEASE USING MACHINE LEARNING

SUBMITTED TO THE

SAVITRIBAI PHULE PUNE UNIVERSITY

IN PARTIAL FULFILLMENT FOR THE AWARD OF THE DEGREE

**OF**

**BACHELOR OF ENGINEERING**

**IN**

**INFORMATION TECHNOLOGY**

BY

**VIRAJ S. VARALE (407182)**

**SUNIL GURJAR (407160)**

**KOMAL B. MORE (407109)**

**APURVA S. BHUSARI (407159)**

UNDER THE GUIDANCE OF

**DR. MRS. KALPANA S. THAKARE**


Sinhgad Institutes

**DEPARTMENT OF INFORMATION TECHNOLOGY**

SINHGAD COLLEGE OF ENGINEERING

VADGAON (BK.), PUNE 411041

**2019-2020**

Sinhgad Institutes

# CERTIFICATE

This is to certify that the project report entitled

## PREDICTION OF HEART DISEASE USING MACHINE LEARNING

Submitted by

**VIRAJ S. VARALE (407182)**

**SUNIL GURJAR (407160)**

**KOMAL B. MORE (407109)**

**APURVA S. BHUSARI (407159)**

is a bonafide work carried out by them under supervision of Dr. Mrs. Kalpana S. Thakare and it is approved for fullfillment of the requirement of Savitribai Phule Pune University for the award of the Degree of Bachelor of Engineering (Information Technology)

This project report has not been earlier submitted to any other Institute or University for the award of any degree or diploma.

Prof. Mrs. Kalpana S. Thakare
Internal Guide

Prof. G. R. Pathak
Head of the Department

.....
External Examiner

Dr. S.D. Lokhande
Principal

Date:

Place: Pune

# ACKNOWLEDGMENT

We are highly indebted to my guide Dr. Mrs. Kalpana S. Thakare for her guidance and constant supervision as well as for providing necessary information regarding the project report and also for her support in completing the project report. We would like to express my special gratitude and thanks to Staff Members of department of Information Technology for giving us such attention and time.

This acknowledgment would be incomplete without expressing our thanks to Prof. G. R. Pathak, Head of the Department (Information Technology) for his support during the work.

We would like to extend our heartfelt gratitude to our Principal, Dr. S. D. Lokhande who provided a lot of valuable support, mostly being behind the veils of college bureaucracy.

We would also like to express our gratitude towards our parents and friends for their kind co-operation and encouragement which helps us in completion of this report. Our thanks and appreciations also go to my colleague in developing the project report and people who have willingly helped us out with their abilities.

(Viraj Varale)

(Sunil Gurjar)

(Komal More)

(Apurva Bhusari)

## Abstract

Heart disease describes a range of conditions that affect your heart. Diseases under the heart disease umbrella include blood vessel diseases, such as coronary artery disease; heart rhythm problems (arrhythmias) and heart defects. Coronary heart disease (CHD) is a major cause of mortality and morbidity all over the world. One in 4 deaths in India are now because of cardiovascular disease with ischemic heart disease.

Prediction of cardiovascular disease is a critical challenge in the area of clinical data analysis. Machine learning (ML) has been shown to be effective in assisting in making decisions and predictions from the large quantity of data produced by the health care industry. System have also seen ML techniques being used in recent developments in different areas of the Internet of Things (IOT). Various studies related heart disease calculates maximum accuracy up to 88.7% by using different machine learning techniques.

We proposed a system that uses signi?cant features by applying machine learning techniques resulting in improving the accuracy in the prediction of cardiovascular disease. The prediction model is introduced with different combinations of features and several known classi?cation techniques. We produce an enhanced performance level with an accuracy level of 94.4% through the prediction model for heart disease with three different algorithms random forest, Na?ve Bayes and logistic regression

We proposed a system that uses machine learning techniques to predict heart disease using three different algorithms random forest, Na?ve Bayes, logistic regression. Proposed system uses Cleveland dataset from machine learning uci repository for training and testing of machine learning model. Cleveland dataset includes important clinical records of patients with class labels. Proposed system uses real time heart rate system for better accuracy.

# CONTENTS

# Contents

# List of Figures

# List of Tables

# Chapter 1

# INTRODUCTION

Coronary heart disease (CHD) is a major cause of mortality and morbidity all over the world. According to a report of World Health Organization (WHO) cardiovascular disease (CVD) caused 17.5 million (30%) of the 58 million deaths that occurred worldwide. While the prevalence and mortality due to CHD is declining in the developed nations the same cannot be held true for developing countries. There has been an alarming increase over the past two decades in the prevalence of CHD and cardiovascular mortality in India and other south Asian countries

Heart disease now is the leading individual cause of disease burden in India, and stroke is the fifth leading cause Heart disease and stroke together contributed to 28. 1% of total deaths in India in 2016 compared with 15. 2% in 1990. Heart disease contributed 17. 8% of total deaths and stroke contributed 7. 1% of total deaths. The proportion of deaths and disability from heart disease was significantly higher in men than in women, but was similar among men and women for stroke. Deaths due to cardiovascular diseases rose from 13 lakh in 1990 to 28 lakh in 2016. The number of prevalent cases of cardiovascular diseases has increased from 2.57 crore in 1990 to 5.45 crore in 2016. The prevalence was the highest in Kerala, Punjab and Tamil Nadu, followed by Andhra Pradesh, Himachal Pradesh, Maharashtra, Goa, and West Bengal. More than half of the total cardiovascular disease deaths in India in 2016 were in people younger than 70 years. This proportion was the highest in less developed states, which is a major cause for concern with respect to the challenges posed to the health systems. Reducing premature deaths from cardiovascular diseases in the economically productive age groups requires urgent action across all states of India.

India has witnessed an alarming rise in the occurrence of heart disease, stroke, diabetes and cancers in the past 25 years. Detailed estimates of cardiovascular diseases, diabetes, chronic respiratory diseases, and cancer show that their prevalence has gone up in every Indian state between 1990 and 2016, but there is vast variation among states. The prevalence of heart disease and stroke has increased by over 50% from 1990 to 2016 in India, with an increase observed in every state. The contribution of these diseases to total deaths and disease burden in the country has almost doubled in the past 25 years.

Various cutting edge technologies like machine learning, artificial intelligence and big data can be used to cure heart disease to reduce premature deaths. We

have studied various machine learning techniques for heart disease. Different studies used different types of machine learning algorithms like decision tree, genetic algorithm, k-nearest neighbours, random forest, logistic regression and neural network, but can?t be able to get expected results. We have also seen hybrid random forest and linear model (HRFLM) for prediction of heart disease. HRFLM model achieved 88.7% accuracy. HRFLM model proposed implementation of random forest and linear model algorithm for prediction.

The proposed work suggests the prediction of heart disease using supervised machine learning algorithm using random forest, Na?ve Bayes and logistic regression algorithm and design of a health care system that provides various services to monitor the patients using wireless technology. It is an intelligent portable Patient monitoring system which is integrating patient monitoring with various medical features, wireless devices and android device. This system mainly provides a solution for prediction of heart diseases using machine learning techniques by monitoring heart rate with help of pulse sensor. It is also acts as a decision-making system which will be used to reduced time before treatment apart from the decision-making techniques. The proposed system provides a framework for measuring the heart rate of the patient using a Arduino Uno microcontroller, Wi-Fi esp01 module and pulse sensor and the measured parameters is sent to the Wi-Fi hotspot enabled Android smart-phone. Input data collected from user and heart rate value is transmitted to the server side where machine learning algorithm is implemented. Proposed system uses NGROK server, FLASK micro web framework and JSON for human readable data transmission from client to server. This model takes input values from user and give results to respective doctor and patients for further analysis.

## 1.1 Motivation

In developing nation like India, the number of allopathic doctors are very less. There is only 1 allopathic doctor for 10,000 of patient, so that it increases the burden on medical sector. We can use computer technology like machine learning to reduce this gap. We can reduce premature death by using machine learning techniques. Many software, system related to diagnosis of heart disease has got developed and are available in the existing literature. But performance, usability and accuracy of such software, system made them less impactful and not reliable. Considering these research gap, we are highly motivated to developed proposed system.

## 1.2 Problem Definition

?To design and implement a prediction system, that predict and analysis of the heart disease using machine learning algorithm like random forest, Naive Bayes and logistic regression.

# Chapter 2

# LITERATURE SURVEY

## 2.1 Literature Review

Heart disease is one of the most significant causes of mortality in the world today. Prediction of heart disease is an important challenge in the area of clinical data analysis. Many of the researcher have worked in this particular domain. Few of them research papers are discussed below:

**Senthilkumar Mohan Chandrasegar Thirumalai1 and Gautam Srivastava "Effective heart disease prediction using hybrid machine learning techniques" (Member, IEEE)" 2019 [1]**

Author have propose hybrid random forest and linear model (HRFLM) system using random forest and linear model algorithm. The prediction model is introduced with different features and several classification techniques. System produced an accuracy level of 88:7% through the prediction model for heart disease with the hybrid random forest with a linear model. This research paper shows that use random forest and linear model gives better accuracy than any other algorithm like decision tree, genetic algorithm and neural network in this particular areas. Hybrid random forest and linear model increases accuracy but in spite of that it can?t be able to detect type of heart disease.

**S. Sreejith, S. Rahul and R.C. Jisha "a real time patient monitoring system for heart disease prediction system using random forest" 2016 [2]**

The proposed work suggests portable Patient monitoring system with various sensitive parameters, wireless devices and smart phones. This system uses random forest algorithm to predict heart disease. System have ability to reduce communication gap between doctor and patient. System provides patient and doctor messaging facility. System also have facility to generate and forwards alarm messages to the related doctor and relatives of patient. The various parameters are analyzed and processed by android application at client side. The processed output is transferred to the server. Whenever an emergency alarm arises an alert message is forwarded to the doctor by the client-side application. Accuracy of this system is only 84.48% which is very less compared to

other research papers. System uses only one algorithm for prediction so that it decreases the reliability system.

**"M. A. Jabbar1, B. L. Deek- Shatulu2 and Priti Chandra3 "Intelligent heart disease prediction disease prediction system using random forest and evolutionary approach" 2017 [3]**

Several data mining techniques are used by researchers to help health care professionals to predict the heart disease. Random forest is an ensemble and most accurate machine learning algorithm for classification, suitable for medical applications. Chi square feature selection measure is used to evaluate between variables and determines whether they are correlated or not. System propose a classification model which uses random forest as classifier, chi square and genetic algorithm as feature selection measures to predict heart disease. The experimental results have shown that author?s approach to improve classification accuracy compared to other classification approaches, and the presented model can be successfully used by health care professional for predicting heart disease.

**Heart Disease Prediction System using Naive Bayes Dhanashree S. Medhekar1, Mayur P. Bote2, and Shruti D. Deshmukh3. Feb. 2019 [4]**

Author presents a classifier approach for detection of heart disease and shows how Naive Bayes can be used for classification purpose. System categories output result into five categories namely no, low, average, high and very high. Also, if unknown sample comes then the system will predict the class label of that sample. Hence two basic functions namely classification (training) and prediction (testing) will be performed. Accuracy of the system is depends on algorithm and database used. System does not use k-fold cross validation techniques to get accurate results.

**Heart Disease Prediction using Logistic Regression Algorithm using Machine Learning Reddy Prasad, Pidaparthi Anjali, S.Adil and N. Deepa March 2013 [5]**

The main theme of this paper is the prediction of heart diseases using machine learning techniques by summarizing the few current researches. In this paper the logistic regression algorithms is used and the health care data which classifies the patients whether they are having heart diseases or not according to the information in the record. System used to predict heart disease using logistic regression algorithm. Author gives accuracy of different algorithms decision tree, Na?ve Bayes, support vector machine and logistic regression. Logistic regression have highest accuracy 86.89% by comparing other three algorithms.

# Chapter 3

# SOFTWARE REQUIREMENTS SPECIFICATION

A Project Specification (or spec) is a comprehensive description of objectives and the requirements for a development project. It contains all goals, functionality, and details required for a development team to fulfill the vision of the client.

## 3.1 System Features

1. Gathering statistical data
2. Analyzing gathered data
3. Drawing conclusion from analyzed data
4. Using random forest classifier, for predicting outcome from statistical
5. Using the predictive model to predict heart rate disease
6. System predict the final answer heart disease

## 3.2 External Interface Requirements

External interface requirements specify hardware, software, or database elements with which a system or component must interface. This section provides information to ensure that the system will communicate properly with external components.

**User Interfaces**

A. Login:
To login in the system user has first register himself/herself. After successful Registration user can login into the system.

B. Patient:
From his module user can make his/her profile and see his/her heart rate monitoring data continuously. User can see previous data.

C. Patient Registration:
In this module, people get registered in this app and give its overall details related to patient, i.e. it fills in a registration form by giving the total details such as name, address, city, sex, weight, DOB, blood group, telephone numbers, e-mail address, etc. The proposed system was also given two fields? username and password to fill such that the system has to registered donor and can enter the login form with its username and password and can modify the details if needed.

D. Update Profile:
The registered doctor is able to modify patients? details. Patient can also modify his details as the login form restricts others from entering the username and password providing high security for the details given by the patient. After giving the username and password it checks for the health detail whether he is an existing patient or not and if the username and password matches, he can then able to modify his total details.

E. Prediction Result on the basis of heartbeat rate:
Prediction of heart disease is a important challenge in the area of medical data analysis. Machine learning (ML) has been shown to be effective in assisting in making decisions and predictions from the large quantity of data produced by the health care industry. Various studies give only a partial view into predicting heart disease with ML techniques. This paper includes a novel method that aims to find significant features by applying supervised machine learning techniques resulting in improving the accuracy in the prediction of heart disease.

F. Doctors:
Cloud communicating is an emerging technology that can be integrated with traditional health management used to provide better health services. Again, social media nowadays become an important medium of communication. The scalability, adaptability, reduction of cost and high-performance features of machine learning for improvising of the medical sector

## 3.3 Hardware Requirement

Since the application must run over the internet, all the hardware shall require to connect internet will be hardware interface for the system. As for e.g. Modem, WAN ? LAN, Ethernet Cross-Cable.

Table 3.1: Hardware Requirement

| Name | Details |
|---|---|
| Processor | Intel i3 |
| RAM | 4GB and above |
| Hard Drive | 100 GB and above |
| Micro controller | Arduino UNO |
| Sensor | Heart rate sensor, Wi-Fi module |

## 3.4 Software Requirement

Following are the software used for the analysis and prediction of cricket match outcome application.

Table 3.2: Software Interfaces

| Name | Details |
|---|---|
| Operating system | Android |
| Database | NoSQL |
| Tools | Android Studio , Anaconda , Arduino |
| Server | NGROK Reverse Proxy Server |
| Languages | Python , Java |

**JAVA JDK and JRE:**
Proposed system using Java Development Kit in order to execute the Java code. The Java Development Kit (JDK) is used to implement of either one of the Java SE, Java EE or Java ME platforms. The JDK includes java virtual machine and a few other resources to finish the Java Application. The Java Run-time Environment (JRE), is part of the Java Development Kit (JDK), a set of programming tools for developing Java applications. The Java Run-time Environment provides the minimum requirements to execute a Java application; it consists of the Java Virtual Machine (JVM), core classes, and supporting files.

**NOSQL**
NOSQL is not only structure query language database management system. The application is utilized by a wide range of purposes, including data warehousing, e-commerce, and logging applications. The most common utilized for NOSQL however, is for the purpose of a web database. It can be utilized by store anything from a single record of information to an entire inventory of available products for an online store. In association with a scripted language such as PHP or Perl (both offered on hosted accounts) it is used to create websites for interaction in real-time with a NOSQL database. In order to quickly display classified and searched information to a website user, it is helpful.

# Chapter 4

# SYSTEM DESIGN

## 4.1 System Architecture

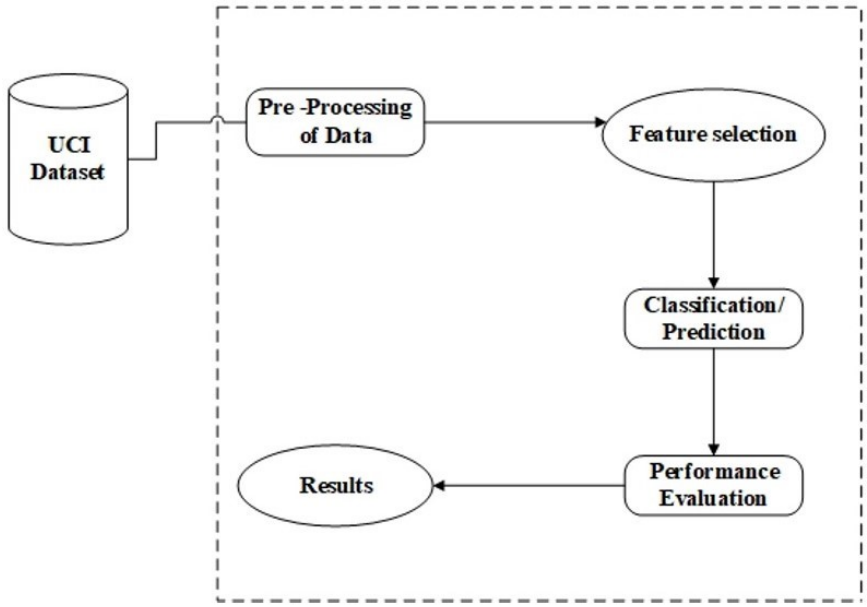



Figure 4.1: System Architecture Design

Proposed System uses Cleveland dataset from uci repository .Dataset contains total 13 features with class label. 11 in 13 features contains important clinical information.

**Total 13 features are given in Cleveland dataset**

1. Age.
2. Chest pain represented in 4 values.

3. Cholesterol level.
4. Blood sugar level.
5. ECG result.
6. Max rate of heart.
7. Angina induced by exercise.
8. Exercised induced ST depression in comparison with state of heart.
9. Fluoroscopy color vessel numbered from 0 to 3.
10. Status of heart normal-3 defect-6 reversible defect -7.
11. Fluoroscopy colored major vessel number from 0 ? 3.
12. Heart disease diagnosis represented in 5 values.
13. Sex -0 for male and 1 for female

Proposed system uses chi square method and f-measure for feature selection. This is the formula for Chi-Square:

$X^2 = \sum P(O - E)^2/E$

O = each Observed (actual) value
E = each Expected value
Calculate (O?E)2/E for each pair of observed and expected values then sum them all up.

Proposed system uses 4 different machine learning algorithm Random Forest, Na?ve Bayes, Logistic Regression and Voting classifier to predict heart disease. Voting classifier is used to ensemble machine learning algorithm to get single result from different algorithm.

Performance of the system is calculated using confusion matrix. Calculate True Positive, True Negative, False Positive, and False Negative from confusion matrix and define accuracy, recall and precision to evaluate system performance. Finally proposed system is used to predict heart disease. Proposed system uses soft voting and hard voting to calculate final result.

## 4.2 Data Flow Diagram

**Dataflow Diagram Level**

**DFD LEVEL 0**

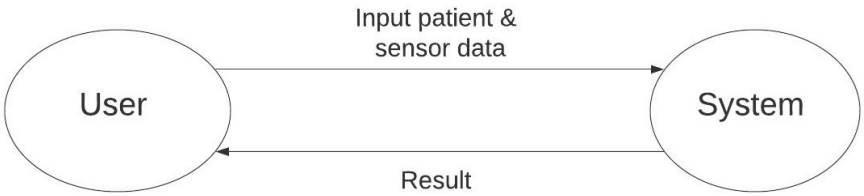



Figure 4.2: DFD Level 0

Data flow diagram level 0 contains only one process node that describe entire system relationship with external entity.

**DFD LEVEL 1**

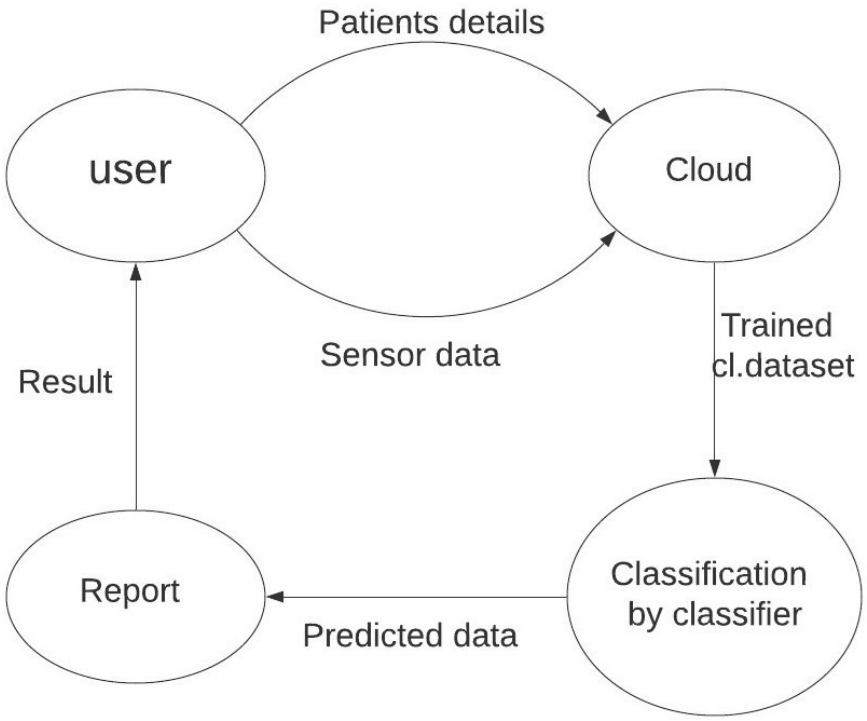



Figure 4.3: DFD Level 1

Data flow diagram level 1 shows sub-process of entire system and its direction in which data has to be flow. It is just overview of entire system and its main entity.

**DFD LEVEL 2**

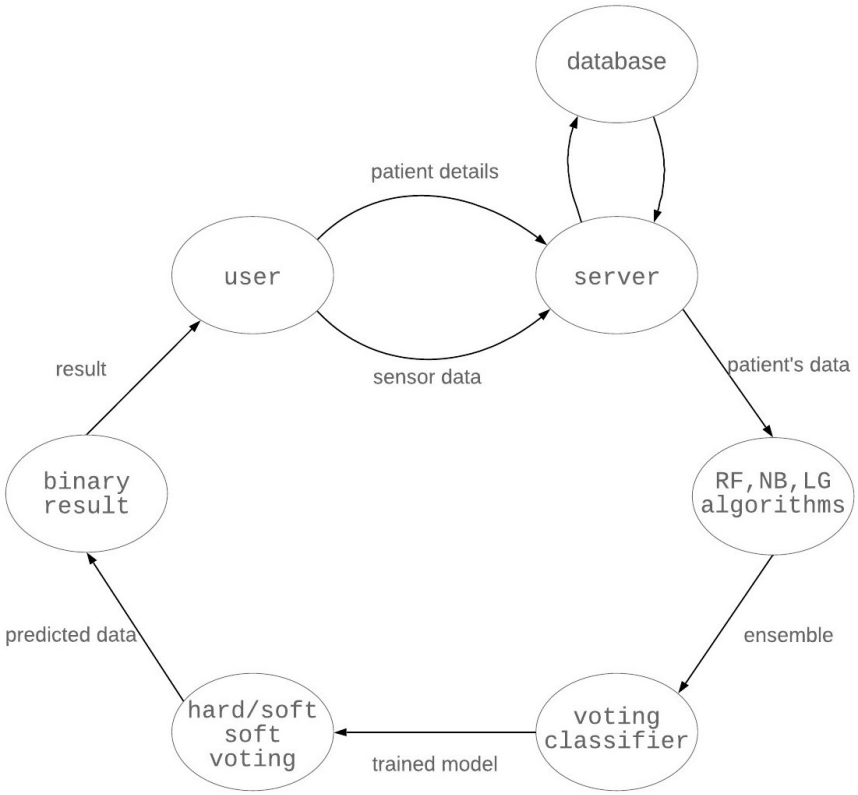



Figure 4.4: DFD Level 2

Data flow diagram level 2 shows entire system in detail view. It shows each process and its sub process in detail view.

## 4.3 UML Diagrams

**Use Case Diagram**

Use case diagram is dynamic or behavior diagram. It models the functionality of system. Use case diagram simply represents a user?s interaction with system that shows the relationship between the users.

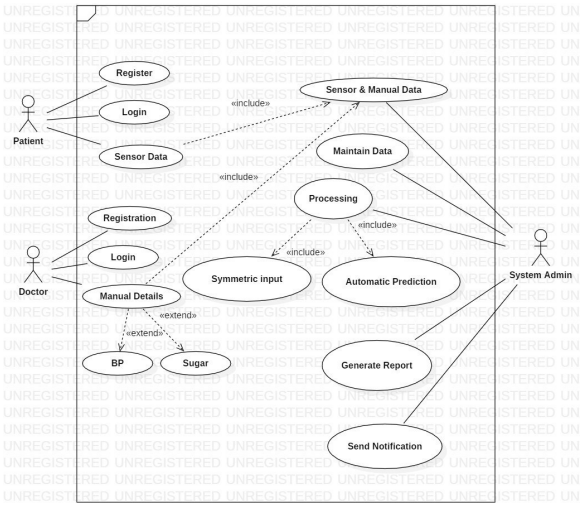



Figure 4.5: Use Case Diagram

### Sequence Diagram

Sequence diagram simply describes interaction between objects in sequential order. The order in which these interaction takes place.

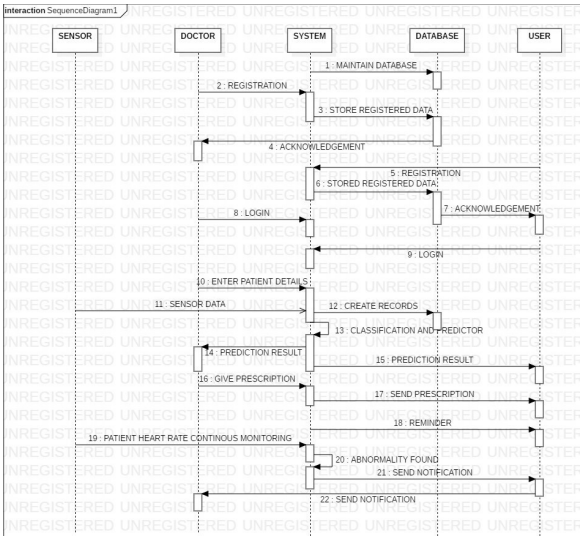



Figure 4.6: Sequence Diagram

### Activity Diagram

Activity diagram describes dynamic aspect of the system. It is important diagram in UML. Activity diagram basically similar to flowchart diagram which depicts flow of one activity to another activity.

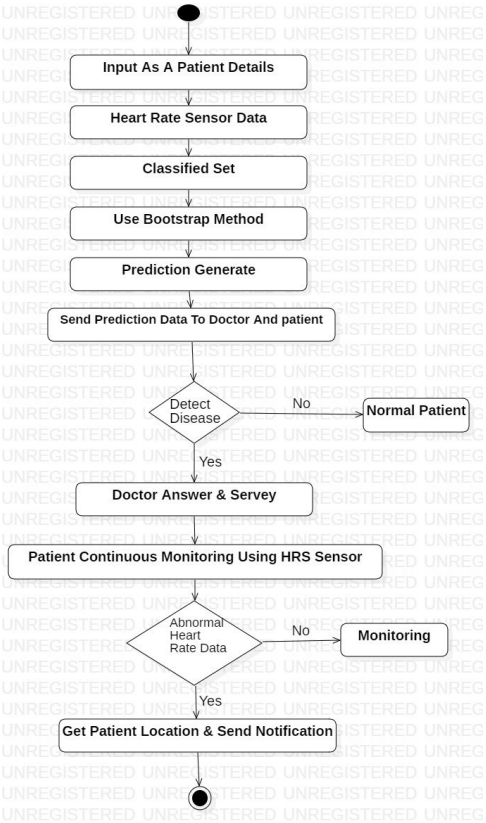



Figure 4.7: Activity Diagram

### Class Diagram

A class diagram is static structure of UML diagram. It is very important diagram in terms of implementation of codes and its main building block of object-oriented modeling. Class diagram includes connection of classes present in system.

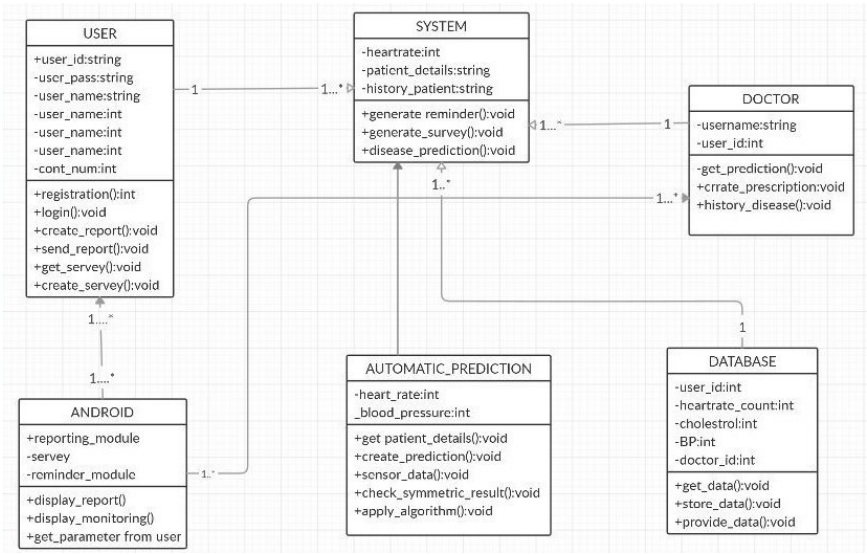



Figure 4.8: Class Diagram

# Chapter 5

# Implementation of proposed system

Block diagram (figure 5.1) shows implementation of proposed system. Proposed system of heart disease prediction implements NGROK reverse proxy server and flask web framework on server side and android application on client side. Json (Java Script object notation) is used to transmit human readable data from client side to server side. Json transmit string of all input values to the server, Server returns string key value pair to client using flask web framework. Proposed system uses http protocol?s post method to get request from client to server and return response from server to client.

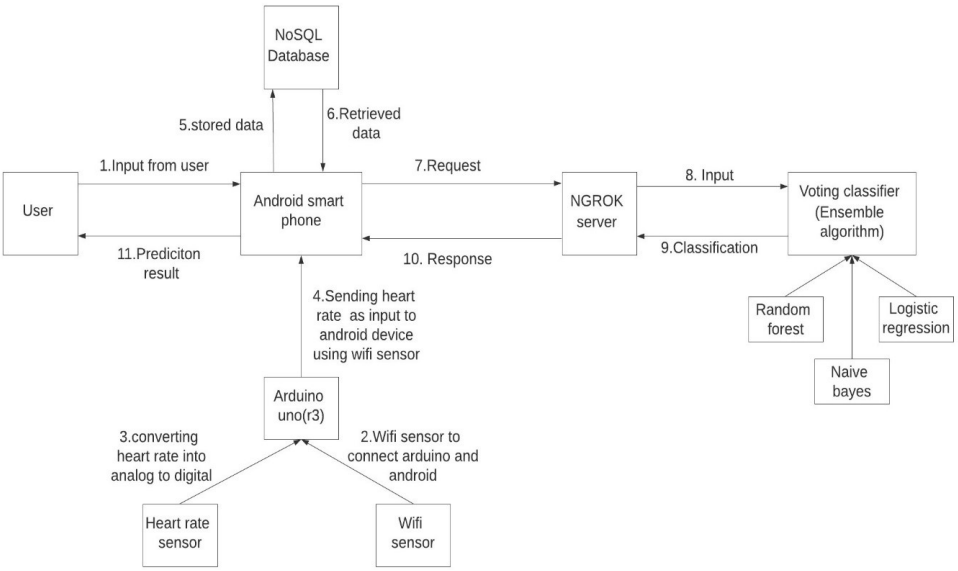



Figure 5.1: Block Diagram of Proposed System

**Above block diagram of show implementation of proposed system in sequence**

1. Proposed system gets input from user.
2. Connection of Arduino and android using Wi-Fi ESP Module.
3. Heart rate sensor to get heart rate value from patient.
4. Sending heart rate value as input from heart rate sensor to android application with the help of Arduino microcontroller..
5. Stored login values to NOSQL database .
6. Retrieved username and password values from NOSQL database. Proposed system uses auto incremented patient id as primary key.
7. Create Json (java script object notation) of input values and send this key value string to server.
8. Extract values using key associated with it. This values are used as input to the voting classifier model to predict patient has heart disease or not.
9. Voting classifier predict the output in terms of probability and classification returns output to the client.
10. Client receives response from server.
11. Finally user can see prediction result in two format
    1. probability- it gives information about chances of heart disease in percentage
    2. a Classification ? it simply predicts patient has heart disease or not.

## 5.1 Algorithms

Proposed system uses machine learning machine learning algorithm to classification of heart disease. System uses 4 different algorithm

1. Random forest algorithm
2. Logistic regression algorithm
3. Naive Bayes algorithm
4. Voting classifier algorithm

**Random forest algorithm**

Random forest is a supervised learning algorithm which is used for both classification as well as regression. But however, it is mainly used for classification problems. In proposed system, random forest algorithm is used for prediction of heart disease. In Random forest, forest is made up of trees and more trees means more robust forest. Proposed system uses 100 decision trees and each decision tree is created by using entropy and information. Similarly, random

forest algorithm creates decision trees on data samples and then gets the prediction from each of them and finally selects the best solution by means of voting. It is an ensemble method which is better than a single decision tree because it reduces the over-fitting by averaging the result. Proposed system uses sklearn libraries to import random forest algorithm.

```
/* Python Codes */
Importing random forest algorithm in python -
from sklearn.ensemble import RandomForestClassifier

Implementing random forest algorithm in python code sample-
rf_model=RandomForestClassifier
    (n_estimators=100,n_jobs=2,criterion='entropy')

Training of random forest algorithm model -
rf_model.fit(x_train,y_train)
```

### Random forest pseudo code

```
Precondition: A training set S = (x1, y1)... (Xn, yn), features F, and
    number of trees in forest B.
1. Function Random Forest(S, F)
2. H ??
3. for i?1... B do
4. S(I) ? A bootstrap sample from S
5. Hi ? Randomized Tree Learn(S (i),F)
6. H ? H? {hi}
7. End for
8. Return H
9. End function
10. Function RandomizedTreeLearn(S, F)
11. at each node:
12. F ? very small subset of F
13. Split on best feature in f
14. Return the learned tree
15. End function
```

### Decision Tree

In proposed system, random forest creates 100 decision trees. Each decision tree is created using entropy of dataset and information gain. Feature which has highest information gain will be selected as root node of decision. Similarly parent and child node is also selected using information gain. Each branch represents the outcome of the prediction , and every particular leaf node represents a class label (decision taken after computing all attributes).proposed system uses only two class label, patient have heart disease and patient does not have heart disease. The path from root to leaf represent classification rules.

**Entropy** A decision tree is a construct top-down from a root node and involves partitioning the data into subsets that contain occurrences with approximate values. Decision trees algorithm uses the entropy to calculate the homogeneity of a sample. If the sample is fully homogeneous, the entropy is zero and if the sample is equally divided, then it has entropy of one.

**Entropy Formula**

$$E(s) = P(yes)log2P(yes) - P(no)log2P(no)$$

**Information Gain** The information gain is dependent on the decrease in entropy after a data-set is split on an attribute. Constructing a decision tree is all about searching attribute that returns the highest information gain.

**Information Gain Formula**

$$I.G = Entropy(s) - Weightedavg * EntropyofeachFetaure$$

**Hyper parameters used in proposed system**

The hyper parameters in random forest are either used to increase the predictive power of the model or to make the model faster System uses following parameters of sklearns built-in random forest function to implement random forest.

**1. Increasing the predictive power** Firstly, there is the n_estimators=100 hyperparameter, which is just the number of trees the algorithm builds before taking the maximum voting or taking the averages of predictions. In general, a higher number of trees increases the performance and makes the predictions more stable, but it also slows down the computation.

**2. Increasing the model's speed** The n_jobs =2 hyperparameter tells the engine how many processors it is allowed to use. If it has a value of one, it can only use one processor. A value of ?-1? means that there is no limit.

The random_state=42 hyper parameter makes the model?s output replicable. The model will always produce the same results when it has a definite value of random state and if it has been given the same hyper parameters and the same training data. Lastly, there is the oob_score (also called oob sampling), which is a random forest cross-validation method. In this sampling, about one-third of the data is not used to train the model and can be used to evaluate its performance. These samples are called the out-of-bag samples. It's very similar to the leave-one-out-cross-validation method, but almost no additional computational burden goes along with it.

### 5.1.1 Naive Bayes algorithm

**Naive Bayes classifier calculates the probability of an event in the following steps:**

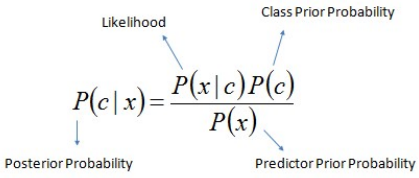
```
Step 1: Calculate the prior probability for given class labels
Step 2: Find Likelihood probability with each attribute for each class
Step 3: Put these value in Bayes Formula and calculate posterior
    probability.
```

$$P(A|B) = \frac{P(B|A) \times P(A)}{P(B)},$$

```
Where A and B are events,
P (A|B) is the conditional probability that event A occurs given that
    event B has already occurred
(P (B|A) has the same meaning but with the roles of A and B reversed) and
P (A) and P (B) are the marginal probabilities of event A and event B
    occurring respectively

Step 4: See which class has a higher probability, given the input
    belongs to the higher probability class.

Bayes theorem provides a way of calculating the posterior probability, P
    (c|x), from
P(c), P(x), and P (x|c). Naive Bayes classifier assume that the effect
    of the value of a
Predictor (x) on a given class (c) is independent of the values of other
    predictors.
This assumption is called class conditional independence.
```




$$P(c|X) = P(x_1|c) \times P(x_2|c) \times \cdots \times P(x_n|c) \times P(c)$$

```
P (c|x) is the posterior probability of class (target) given
    predictor (attribute).
P(c) is the prior probability of class.
P (x|c) is the likelihood which is the probability of predictor given
    class.
P(x) is the prior probability of predictor.
Importing Na?ve Bayes algorithm in python
from sklearn.naive_bayes import GaussianNB
Implementing Na?ve Bayes algorithm in python code sample-
nb_model=GaussianNB()
Training of Na?ve Bayes algorithm model -
nb_model.fit(x_train, y_train)
```

Naive Bayes is a probabilistic machine learning algorithm based on the Bayes Theorem, used in a wide variety of classification tasks. Proposed system uses Na?ve Bayes for classification of heart disease. Why is it called ?Naive?? The name naive is used because it assumes the features that go into the model is independent of each other. That is changing the value of one feature, does not directly influence or change the value of any of the other features used in the algorithm. Proposed system uses sklearn libraries from python to implement Na?ve Bayes algorithm.

### 5.1.2 Hyper Parameter used in proposed system

Scikit learn (python library) will help here to build a Naive Bayes model in Python. There are three types of Naive Bayes model under the scikit-learn library:
Gaussian
It is used in classification and it assumes that features follow a normal distribution. Proposed system uses Gaussian for classification of heart disease Code nb_model=GaussianNB ()

### 5.1.3 Logistic regression

Logistic regression is a classification algorithm used to assign observations to a discrete set of classes. Unlike linear regression which outputs continuous number values, logistic regression transforms its output using the logistic sigmoid function to return a probability value which can then be mapped to two or more discrete classes. Proposed system uses Logistic regression to map two discrete class. It is used for classification of heart disease.

**Importing logistic regression using sklearn library in python-**
From sklearn.linear_model import LogisticRegression

**Implementing random forest algorithm in python code sample-**
lg_model = LogisticRegression (random_state=0, max_iter = 4000)

**Training of random forest algorithm model -**
lg_model.Fit(x_train, y_train)

**Logistic regression pseudo code ?**

1. $0 \rightarrow \beta$
2. Compute $y$ by setting its elements to:

$$y_i = \begin{cases} 1 & \text{if } g_i = 1 \\ 0 & \text{if } g_i = 2 \end{cases}$$

3. Compute $p$ by setting its elements to:

$$p(x_i;\beta) = \frac{e^{\beta^T x_i}}{1+e^{\beta^T x_i}} \quad i = 1, 2, \ldots, N$$

4. Compute the $N \times (p+1)$ matrix $\tilde{\mathbf{X}}$ by multiplying the $i$th row of matrix $\mathbf{X}$ by $p(x_i;\beta)(1 - p(x_i;\beta))$, $i = 1, 2, \ldots, N$:

$$\mathbf{X} = \begin{pmatrix} x_1^T \\ x_2^T \\ \cdots \\ x_N^T \end{pmatrix} \tilde{\mathbf{X}} = \begin{pmatrix} p(x_1;\beta)(1-p(x_1;\beta))x_1^T \\ p(x_2;\beta)(1-p(x_2;\beta))x_2^T \\ \cdots \\ p(x_N;\beta)(1-p(x_N;\beta))x_N^T \end{pmatrix}$$

5. $\beta \leftarrow \beta + (\mathbf{X}^T\tilde{\mathbf{X}})^{-1}\mathbf{X}^T(\mathbf{y} - \mathbf{p})$.
6. If the stopping criteria is met, stop; otherwise go back to step 3.

### Sigmoid activation

In order to map predicted values to probabilities, we use the sigmoid function. The function maps any real value into another value between 0 and 1. In machine learning, we use sigmoid to map predictions to probabilities.

Math
$S(z) = 1/1 + e^{(?z)}$

$s(z) = outputbetween0and1(probabilityestimate)$
$z = inputtothefunction(youralgorithm?spredictione.g.mx + b)$
$e = baseofnaturallog$

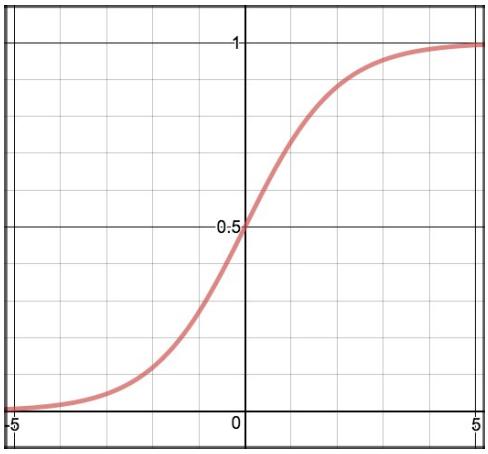



Figure 5.2: Sigmoid activation graph

**Decision boundary**

Our current prediction function returns a probability score between 0 and 1. In order to map this to a discrete class (true/false, cat/dog), we select a threshold value or tipping point above which we will classify values into class 1 and below which we classify values into class 2.
p?0.5, class=1 (patient have heart disease)
p¡0.5, class=0 (patient does not have heart disease)

**Making predictions** Using knowledge of sigmoid functions and decision boundaries, we can now write a prediction function. A prediction function in logistic regression returns the probability of our observation being positive, true, or ?patient have heart disease?. We call this class 1 and its notation is P (class=1). As the probability gets closer to 1, our model is more confident that the observation is in class 1.

**Math**
This time however we will transform the output using the sigmoid function to return a probability value between 0 and 1.
P (class=1) =1/1+e?zP (class=1) =1/1+e?z
If the model returns .4 it believes there is only a 40% chance of passing. If our decision boundary was .5, we would categorize this observation as ?patient does not have heart disease.

### 5.1.4 Voting Classifier

Proposed system uses voting classifier to train following different algorithms
1. Random forest
2. Na?ve Bayes
3. Logistic regression
A collection of several models working together on a single set is called an ensemble. The method is called Ensemble Learning. It is much more useful use all different models rather than any one.

Voting is one of the simplest way of combining the predictions from multiple machine learning algorithms. Voting classifier has two types of voting soft voting and hard voting. Proposed system uses both soft and hard voting for prediction. System uses soft voting to calculate probability of heart disease and hard voting for prediction. Voting classifier isn?t an actual classifier but a wrapper for set of different ones that are trained and valuated in parallel in order to exploit the different peculiarities of each algorithm.

Classification model-
C1-Random forest
C2-Na?ve Bayes
C3-Logistic regression Proposed system train data set using different algorithms and ensemble then to predict the final output. The final output on a prediction is taken by majority vote according to two different strategies

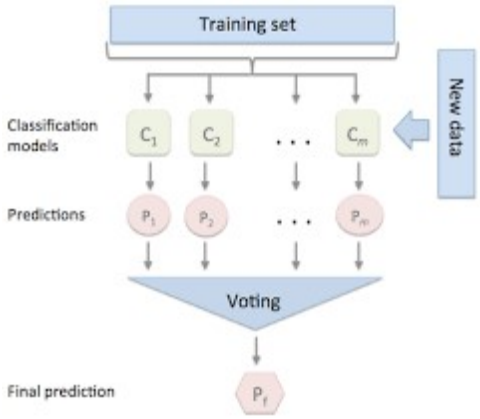



Figure 5.3: Voting Classifier

**Hard voting**

Hard voting is the simplest case of majority voting. In this case, the class that received the highest number of votes will be chosen. Here we predict the class label yv̂ia majority voting of each classifier.

Assuming that we combine three classifiers that classify a training sample as follows:
**Classifier 1(random forest) -¿ class prediction 0**
**Classifier 2(Na?ve Bayes) -¿ class prediction 0 Classifier 3(Logistic regression) -¿ class prediction 1**
**Y≙mode 0, 0, 1 = 0**

**Voting classifier hard voting code-**
$vc\_model = VotingClassifier(estimators = [('rfc', rf\_model), ('lg', lg\_model), ('nb', nb\_model)], voting = 'hard')$
$vc\_model.fit(x\_train, y\_train)$

**Soft voting**

In this case, the probability vector for each predicted class (for all classifiers) are summed up &averaged. The winning class is the one corresponding to the highest value (only recommended if the classifiers are well calibrated).

Y≙ argmax (1/Nclassifier) Summation of (p1, p2, p3).
P1=probability of random forest
P2=probability of Na?ve byes
P3= probability of Logistic regression.

**Voting classifier soft voting code-**
vc_model = VotingClassifier (estimators=[('rfc',rf_model),('lg',lg_model),('nb',nb_model)],voting='soft')
vc_model.fit(x_train,y_train)

**Hyper Parameter used in proposed system**

Scikit learn (python library) will help here to build a Voting Classifier model in Python. System uses following parameters to implement voting classifier.

Estimator
List of (STR, estimators) tuples
Invoking the fit method on the Voting Classifier will fit clones of those original estimators that will be stored in the class attribute self. Estimators
**estimators=[('rfc',rf_model),('lg',lg_model),('nb',nb_model)],voting='soft')**

**Votingstr, ?hard?, ?soft? (default=?hard?)**
If ?hard?, uses predicted class labels for majority rule voting. Else if ?soft?, predicts the class label based on the argmax of the sums of the predicted probabilities, which is recommended for an ensemble of well-calibrated classifiers.
**Voting= 'hard'**
**Voting** ='soft'

**Weights array-like, shape (n_classifiers,), optional (default='None')**
Sequence of weights (float or int) to weight the occurrences of predicted class labels (hard voting) or class probabilities before averaging (soft voting). Uses uniform weights if None.
**N_classifiers=default**

**N_jobsint or none, optional (default=none)** The number of jobs to run in parallel for fit. None means 1 unless in a joblib.parallel_backend context. -1 means using all processors.
**N_jobsint=1**

## 5.2 NOSQL Database:

A **NoSQL** originally referring to not only SQL or non-relational is a database that provides a mechanism for storage and retrieval of data. This data is modeled in means other than the tabular relations used in relational databases. Proposed system uses NOSQL database to stored and retrieved username and password to database. NOSQL database uses auto incremented patient id as primary key.

## 5.3 NGROK

Proposed system uses NGROK revers proxy server. This NGROK server hosts flask micro web framework as web service. NGROK is a multiplatform tunneling, reverse proxy software that establishes secure tunnels from a public endpoint such as internet to a locally running network service while capturing all traffic for detailed inspection and replay. NGROK allows you to take any project live without actually deploying it.

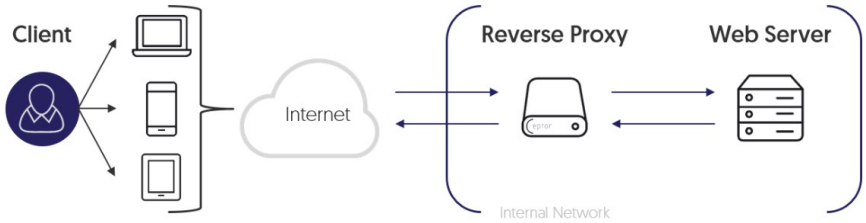



Figure 5.4: NGROK Server

Download and run a program on your machine and provide it the port of a network service, usually a web server. It connects to the NGROK cloud service which accepts traffic on a public address and relays that traffic through to the NGROK process running on your machine and then on to the local address you specified.

```
Importing NGROK server -
from flask_ngrok import run_with_ngrok
NGROK Server code in python -
app = Flask(_name_)
run_with_ngrok(app)
# Start ngrok when app is run
```

## 5.4 Flask

Proposed system uses flask micro web framework for web service. Flask web service implements post method for request and response from https protocol Flask is a lightweight WSGI web application framework. It is designed to make getting started quick and easy, with the ability to scale up to complex applications. It began as a simple wrapper and has become one of the most popular Python web application frameworks.

Flask offers suggestions, but doesn't enforce any dependencies or project layout. It is up to the developer to choose the tools and libraries they want to use. There are many extensions provided by the community that make adding new functionality easy. Web Application Framework or simply Web Framework represents a collection of libraries and modules that enables a web application developer to write applications without having to bother about low-level details such as protocols, thread management etc.

## 5.5 JSON

(JavaScript Object Notation) is lightweight data-interchange format. It is easy for humans to read and write. It is easy for machines to parse and generate. JSON is text format that is completely language independent but uses conventions that are familiar to programmers of the C- family of languages, including c, c++, c# Java, JavaScript, Perl, Python, and many others. These properties make JSON an ideal data-interchange language.

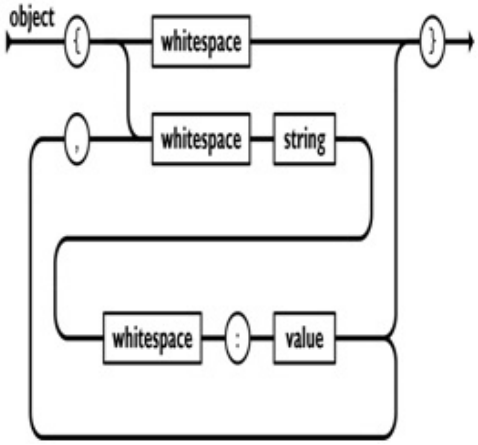



Figure 5.5: JSON Description

**JSON is built on two structures**
A collection of name/value pairs. In various languages, this is realized as an object, record, struct, dictionary, hash table, keyed list, or associative array. An ordered list of values. In most languages, this is realized as an array, vector, list, or sequence.

These are universal data structures. Virtually all modern programming languages support them in one form or another. It makes sense that a data format that is interchangeable with programming languages also be based on these structures.

## 5.6 Arduino

Arduino is an open-source prototyping platform used for building electronics projects. It consists of both a physical programmable circuit board and a software, or IDE (Integrated Development Environment) that runs on your computer, where you can write and upload the computer code to the physical board.

**Arduino Uno (R3) Description**

**Power (USB / Barrel Jack)** Every Arduino board needs a way to be connected to a power source. The Arduino UNO can be powered from a USB cable coming from your computer or a wall power supply (like this) that is terminated in a barrel jack. The USB connection is also how you will load code onto your Arduino board. More on how to program with Arduino can be found in our Installing and Programming Arduino tutorial.
**Pins (5V, 3.3V, GND, Analog, Digital, PWM, AREF)** The pins on your Arduino are the places where you connect wires to construct a circuit (probably in conjunction with a breadboard and some wire. They usually have black plastic ?headers? that allow you to just plug a wire right into the board. The Arduino has several different kinds of pins, each of which is labeled on the board and used for different functions.

- GND (3): Short for ?Ground?. There are several GND pins on the Arduino,
- 5V (4) & 3.3V (5):, the 5V pin supplies 5 volts of power, and the 3.3V pin supplies 3.3 volts of power. Most of the simple components used with the Arduino run happily off of 5 or 3.3 volts.
- Analog (6): The area of pins under the ?Analog In? label (A0 through A5 on the UNO) are Analog In pins. These pins can read the signal from an analog sensor (like a temperature sensor) and convert it into a digital value that we can read.
- Digital (7): Across from the analog pins are the digital pins (0 through 13 on the UNO). These pins can be used for both digital input (like telling if a button is pushed) and digital output (like powering an LED).
- PWM (8): You may have noticed the tilde ( ) next to some of the digital pins (3, 5, 6, 9, 10, and 11 on the UNO). These pins act as normal digital pins, but can also be used for something called Pulse-Width Modulation (PWM). We have a tutorial on PWM, but for now, think of these pins as being able to simulate analog output (like fading an LED in and out).

**AREF (9)** Stands for Analog Reference. Most of the time you can leave this pin alone. It is sometimes used to set an external reference voltage (between 0 and 5 Volts) as the upper limit for the analog input pins.
**Reset Button** The Arduino has a reset button. Pushing it will temporarily connect the reset pin to ground and restart any code that is loaded on the Arduino. This can be very useful if your code doesn?t repeat, but you want to test it multiple times.
**Power LED Indicator** Just beneath and to the right of the word ?UNO? on your circuit board, there?s a tiny LED next to the word ?ON?. This LED should light up whenever you plug your Arduino into a power source. If this light doesn?t turn on, there?s a good chance something is wrong. Time to recheck your circuit!
**TX RX LEDs** TX is short for transmit, RX is short for receive. These markings appear quite a bit in electronics to indicate the pins responsible for serial communication. There are two places on the Arduino UNO where TX and RX appear – once by digital pins 0 and 1, and a second time next to the TX and RX indicator LEDs.
**Main IC** The main IC on the Arduino is slightly different from board type to board type, but is usually from the AT mega line of IC?s from the ATMEL company. This can be important, as you may need to know the IC type (along with your board type) before loading up a new program from the Arduino software.
**Voltage Regulator** The voltage regulator does exactly what it says – it controls the amount of voltage that is let into the Arduino board. It will turn away an extra voltage that might harm the circuit.it has its limits.

### 5.6.1 WIFI ESP 01s module

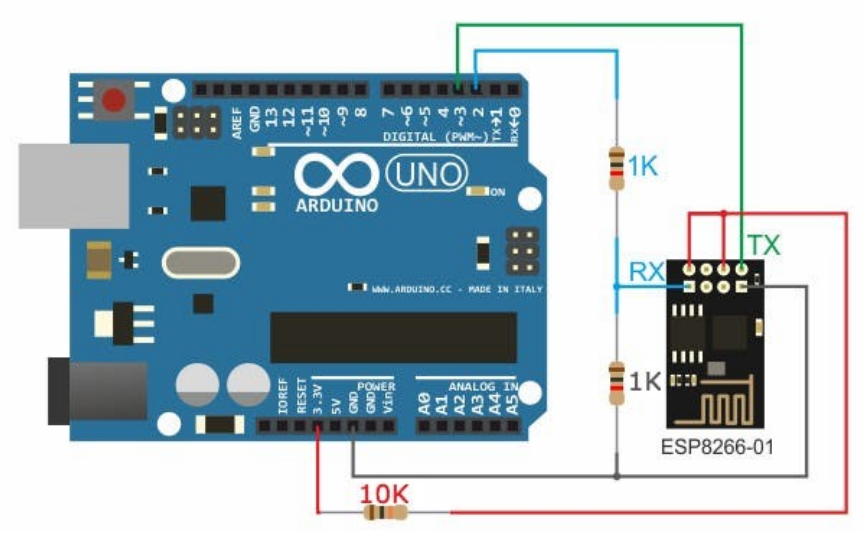



Figure 5.6: WIFI ESP 8266 and Arduino connection

Above figure shows the connection of Wi-Fi esp01 module to the Arduino Uno microcontroller.

Table 5.1: ESP8266 and Arduino Uno (r3) pin connection

| **Wi-Fi esp. 01 module pin** | **Arduino Uno pin** |
|---|---|
| VCC | 3.3 VOLT |
| GND | GND |
| RX | DIGITAL PIN 2 |
| TX | DIGITAL PIN 3 |
| CH_PD | 3.3 VOLT |
| RST | NO CONNECTION |
| GPIO 0 | NO CONNECTION |
| GPIO 2 | NO CONNECTION |

Processor: L106 32-bit RISC microprocessor core based on the Tensilica Xtensa Diamond Standard 106Micro running at 80 MHz

- Memory:

  32 KiB instruction RAM

  32 KiB instruction cache RAM

  80 KiB user-data RAM

  16 KiB ETS system-data RAM

- External QSPI flash: up to 16 MiB is supported (512 KiB to 4 MiB typically included)
- IEEE 802.11 b/g/n Wi-Fi

  Integrated TR switch, balun, LNA, power amplifier and matching network

  WEP or WPA/WPA2 authentication, or open networks
- 16 GPIO pins
- SPI
- $I^2C$ (*software implementation*)
- $I^2S$ *interfaces with DMA* (*sharing pins with GPIO*)
- UART on dedicated pins, plus a transmit-only UART can be enabled on GPIO2
- 10-bit ADC (successive approximation ADC)

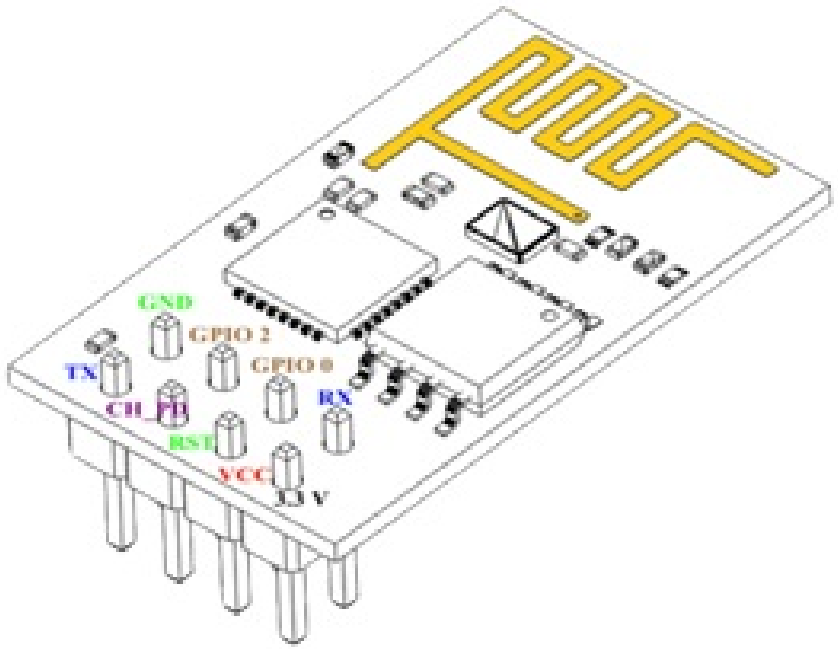



Figure 5.7: WIFI ESP 8266 pin diagram

The pinout is as follows for the common ESP-01 module

1. VCC, Voltage (+3.3 V; can handle up to 3.6 V)
2. GND, Ground (0 V)
3. RX, Receive data bit X
4. TX, Transmit data bit X
5. CH_PD, Chip power-down

6. RST, Reset
7. GPIO 0, General-purpose input/output No. 0
8. GPIO 2, General-purpose input/output No. 2

### 5.6.2 Pulse Sensor


Figure 5.8: Pulse sensor and Arduino connection

Above figure shows connection of heart rate sensor with Arduino uno (r3) microcontroller.

Table 5.2: Pulse sensor and Arduino Uno (r3) pin connection

| **Heart rate sensor pin** | **Arduino uno pin** |
|---|---|
| Signal | Analog in (A0) |
| VCC | 5Volt |
| Ground | GND |

The sensor has two sides, on one side the LED is placed along with an ambient light sensor and on the other side we have some circuitry. This circuitry is responsible for the amplification and noise cancellation work. The LED on the front side of the sensor is placed over a vein in our human body. This can either be your Finger tip or you ear tips, but it should be placed directly on top of a vein. Now the LED emits light which will fall on the vein directly. The veins will have blood flow inside them only when the heart is pumping, so if we monitor the flow of blood we can monitor the heart beats as well. If the flow of blood is detected then the ambient light sensor will pick up more light since they will be reflect ted by the blood, this minor change in received light is analyzed over time to determine our heart beats.

**Features**
Biometric Pulse Rate or Heart Rate detecting sensor
Plug and Play type sensor
Operating Voltage: +5V or +3.3V
Current Consumption: 4mA
Inbuilt Amplification and Noise cancellation circuit.
Diameter: 0.625?
Thickness: 0.125? Thick

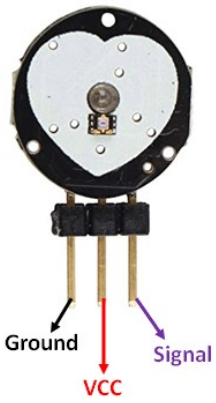



Figure 5.9: Pulse sensor pin diagram

Table 5.3: Pulse sensor pin description

| Pin number | Pin name | Wire colour | Description |
|---|---|---|---|
| 1 | Ground | Black | Connected to the ground of the system |
| 2 | VCC | Red | Connect to +5V or +3.3V supply voltage |
| 3 | Signal | Purple | Pulsating output signal. |

# Chapter 6

# Testing

## 6.1 Software Testing:

It is defined as an activity to check whether the actual results match the expected results and to ensure that the software system is defect free. Software testing also helps to identify errors, gaps or missing requirements in contrary to actual requirements.

### 6.1.1 Types of Software Testing:

**1. Integration Testing:**
Integration testing is a testing in which group of components are combined to produce output. They are sub classified as

**A. Black Box Testing:**
It is used for validation, in this we ignore internal working mechanism and focus on what is the output?

**B. White Box Testing**
It is used for verification. In this we focus on internal mechanism i.e. how the output is achieved.

**2) System Testing:**
In this, software is tested such that it works fine for different operating systems.

**3. Performance Testing:**
It is designed to test the speed and effectiveness of program.

**4. Unit Testing:**
It focuses on smallest unit of software design. In this we test an individual unit or group interrelated units. It is often done by programmer by using sample input and observing its corresponding output.

Table 6.1: Test cases

| Test Case ID | Test case | Input values | Expected result | Actual Result | Status |
|---|---|---|---|---|---|
| 1 | HDP Register | User name, Password, Confirm password | Creation and registration of account and store value into database | Creation of and registration of account and store value into database | pass |
| 2 | Login | User name, password | Successfully login into account | Successfully login into account | pass |
| 3 | Input age | age | Get input values and stored into database. | Get input values and stored into database | pass |
| 4 | Input gender | Male, Female | Get input values and stored into database | Get input values and stored into database | pass |
| 5 | Input chest pain | Typical angina, Atypical angina, Non angina pain, Asymptomatic | Get input values convert into respective weight and stored into database | Get input values convert into respective weight and stored into database | pass |
| 6 | Input blood pressure | Blood pressure | Get input values and stored into database | Get input values and stored into database | pass |
| 7 | Input Cholesterol | Cholesterol | Get input values and stored into database | Get input values and stored into database | pass |
| 8 | Input fasting blood sugar | Normal, Abnormal | Get input values convert into respective weight and stored into database | Get input values convert into respective weight and stored into database | pass |
| 9 | Input ECG | Normal, Abnormal | Get input values convert into respective weight and stored into database | Get input values convert into respective weight and stored into database | pass |
| 10 | Input Heart rate | Heart rate | Get Input values from heart rate sensor using Arduino | Get Input values from heart rate sensor using Arduino | pass |
| 11 | Input angina exercise | Yes, No | Get input values convert into respective weight and stored into database | Get input values convert into respective weight and stored into database | pass |
| 12 | Input ST depression | ST depression | Get input values convert into respective weight and stored into database | Get input values convert into respective weight and stored into database | pass |
| 13 | Input Slope | Unsloping, Flat, Downloading | Get input values convert into respective weight and stored into database | Get input values convert into respective weight and stored into database | pass |

| Test Case ID | Test case | Input values | Expected result | Actual Result | Status |
|---|---|---|---|---|---|
| 14 | Input Fluoroscopy | Fluoroscopy | Get input values convert into respective weight and stored into database | Get input values convert into respective weight and stored into database | pass |
| 15 | Input Heart Status | Normal, Fixed Defect, Reversible Defect | Get input values convert into respective weight and stored into database | Get input values convert into respective weight and stored into database | pass |
| 16 | Final | | Get all input values and send it to the server using http post method and retrieve response from server and goto result activity | Get all input values and send it to the server using http post method and retrieve response from server and goto result activity | pass |
| 17 | Result | Prediction output Prediction output in probability | Retrieve values from final activity and show it to the user | Retrieve values from final activity and show it to the user | pass |

# Chapter 7

# RESULTS AND ANALYSIS

We split Cleveland dataset into train and test part. Train part is used for training algorithms and test part is used to test the trained model. By using test part we can calculate accuracy, precision recall and error rate of created model. We divide Cleveland dataset of heart disease into 75 % of data into training and 25 % into testing. Cleveland dataset have 303 instances in which randomly 227 instances used for training algorithms and 76 instances used for testing of trained prediction model.

## 7.1 Confusion Matrix

Confusion matrix is a performance measurement for machine learning classification problem where output can be two or more classes. It is a table with 4 different combinations of predicted and actual values.

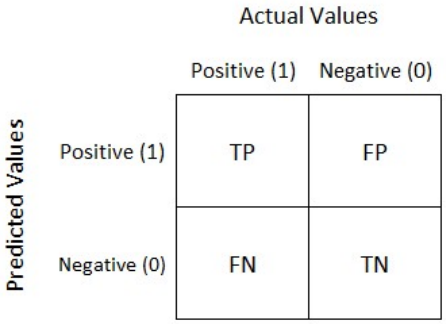



Figure 7.1: Confusion Matrix

It is extremely useful for measuring Recall, Precision, Specificity and Accuracy of machine learning classification algorithm used in proposed system

1. Random forest
2. Na?ve Bayes
3. Logistic Regression
4. Voting classifier

TP (True positive value): gives the number of positive instances correctly predicted by classification model

FN (False negative value): gives the number of positive instances wrongly predicted by classification model

FP (False positive value): gives the number of negative instances wrongly predicted as positive instance.

TN (True negative): gives the number of negative instances correctly predicted by classification model.

### 7.1.1 ACCURACY

It is the number of true positive and false positive predictions for the considered class. This is calculated as the sum of correct classifications divided by the total number of classifications.
Accuracy = (TP+TN)/N, where N is the total number of classifications.

### 7.1.2 PRECISION

It is the measure of the accuracy predicted for specific class and calculated using the formulae.
Precision = TP/ (TP+FP)

### 7.1.3 RECALL

Also called as sensitivity, corresponds to the true positive rate of the considered class and it is calculated using the formulae.
Recall = TP/ (TP+FN)

Following table shows true positive, false positive, false negative, true negative of each algorithms used in proposed system

Table 7.1: Confusion Matrix

| **Algorithm** | **TP** | **FP** | **FN** | **TN** | **Total sample** |
|---|---|---|---|---|---|
| Random forest | 32 | 6 | 2 | 36 | 76 |
| Naive Bayes | 34 | 4 | 4 | 34 | 76 |
| Logistic regression | 33 | 5 | 2 | 36 | 76 |
| Voting classifier | 36 | 2 | 2 | 36 | 76 |

Following results are calculated using confusion matrix with the help of sklearn libraries in python.

By applying confusion matrix on different algorithm we get the following results

Following table shows accuracy precision and recall and error rate of each algorithm used in proposed system

Table 7.2: Proposed system performance

| **Algorithm** | **Accuracy** | **Precision** | **Recall** | **Error rate** |
|---|---|---|---|---|
| Random forest | 90.78 | 88.39 | 88.39 | 10.52 |
| Naive Bayes | 89.47 | 89.47 | 89.47 | 10.52 |
| Logistic regression | 90.78 | 87.80 | 94.73 | 9.21 |
| Voting classifier | 94.73 | 94.73 | 94.73 | 5.26 |

### 7.1.4 Random forest bar chart

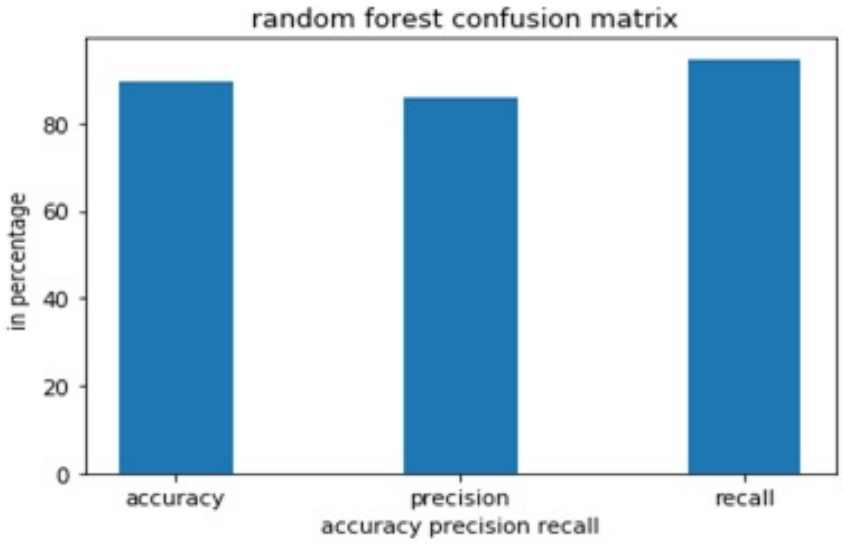



Figure 7.2: Random forest confusion matrix

### 7.1.5 Logistic regression bar chart

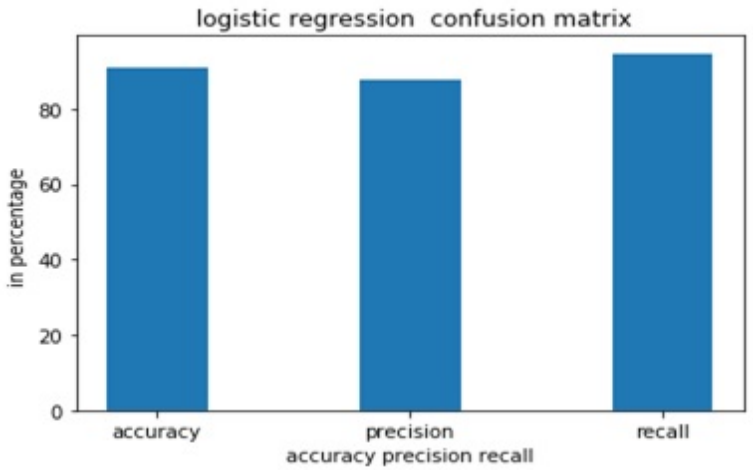



Figure 7.3: Logistic regression bar chart

### 7.1.6 Naive Bayes bar chart

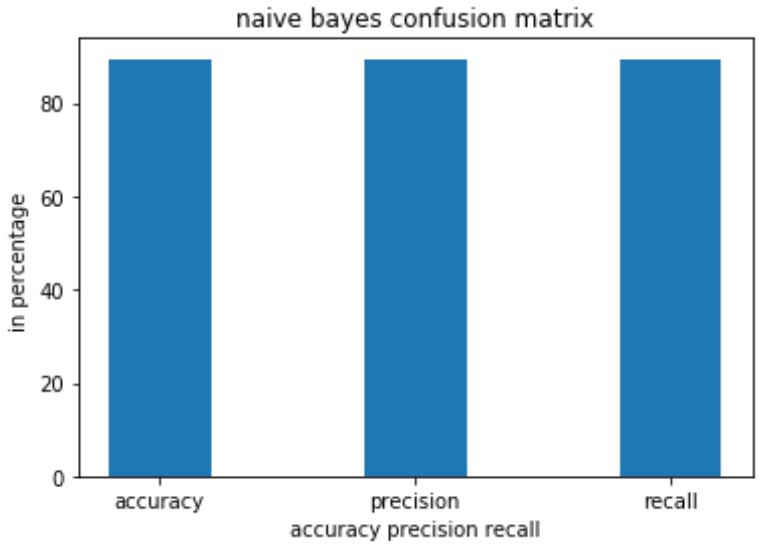



Figure 7.4: Naive Bayes confusion matrix

### 7.1.7 Voting classifier bar chart

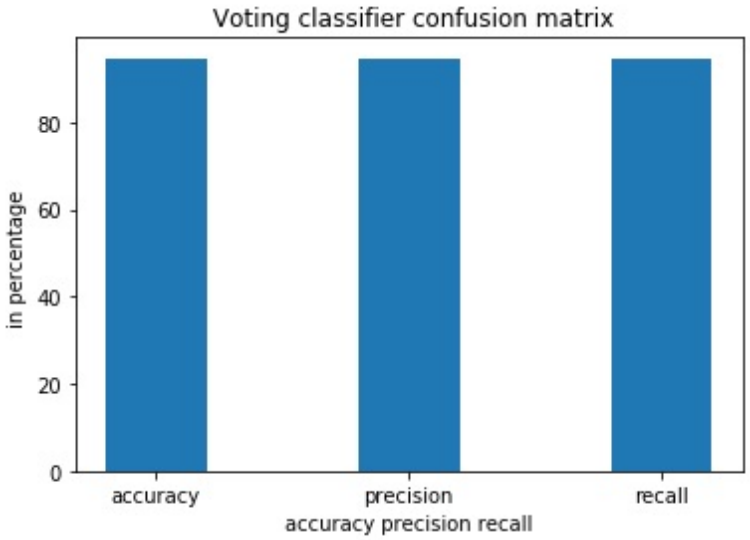



Figure 7.5: Voting classifier confusion matrix

## 7.2 Feature Selection using chi square method

Chi square test is used to calculate correlation between class labels and features. It calculates features scores. The chi square distribution is a theoretical or mathematical distribution which has wide applicability in statistical work. The term ?chi square? (pronounced with a hard ?ch?) is used because the Greek letter ? is used to de?ne this distribution. It will be seen that the elements on which this distribution is based are squared,

$X^2 = X_i(O_i - E_i)^2/E_i$

### 7.2.1 Chi square high score features

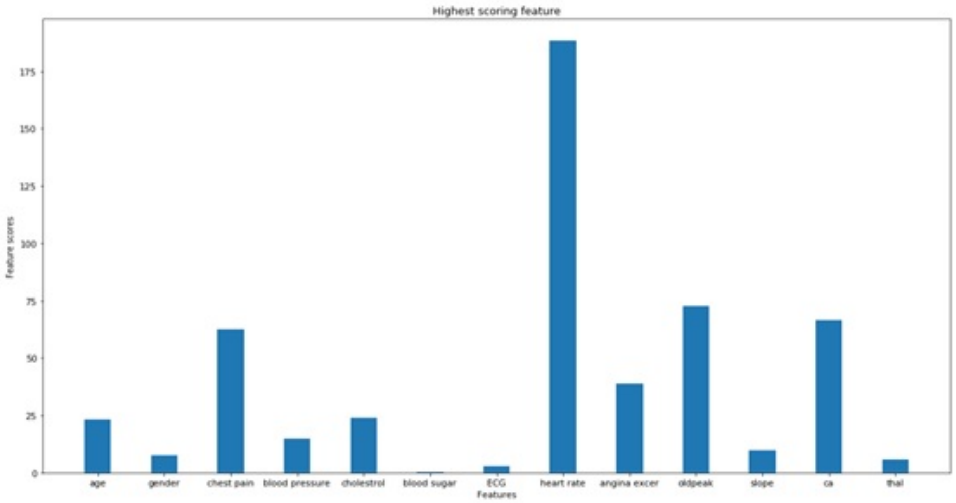



Figure 7.6: Feature selection using chi square method

### 7.2.2 Chi square score

Table 7.3: Feature score using chi square method

| **Features** | **Chi square test score** |
|---|---|
| Age | 23.29 |
| Gender | 7.58 |
| Chest pain | 62.60 |
| Blood pressure | 14.82 |
| Cholesterol | 23.94 |
| Blood sugar | 0.20 |
| ECG | 2.98 |
| Heart rate | 188.32 |
| Angina excel | 38.91 |
| Old peak | 72.64 |
| Slope | 9.80 |
| Fluoroscopy | 66.44 |
| Thal | 5.79 |

## 7.3 F-measure score

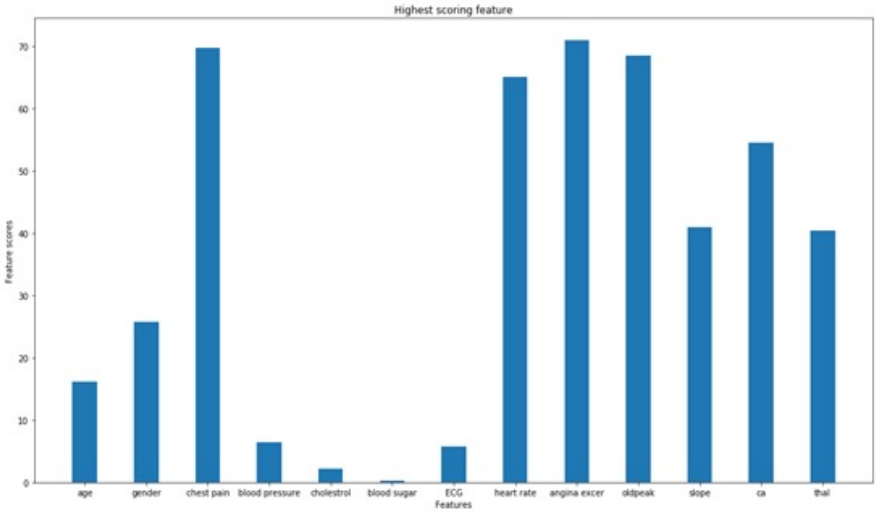



Figure 7.7: Feature selection using F measure

### 7.3.1 F-measure score

Table 7.4: Feature selection using F measure

| **Features** | **Chi square test score** |
|---|---|
| Age | 16.12 |
| Gender | 25.79 |
| Chest pain | 69.77 |
| Blood pressure | 6.46 |
| Cholesterol | 2.20 |
| Blood sugar | 0.24 |
| ECG | 5.78 |
| Heart rate | 65.12 |
| Angina excel | 70.95 |
| Old peak | 68.55 |
| Slope | 40.90 |
| Fluoroscopy | 54.56 |
| Thal | 40.41 |

## 7.4 RESULTS

### 7.4.1 Register page

By using this page admin can register their account on android app. Admin can register using username and password and details of user stored on NoSQL database.

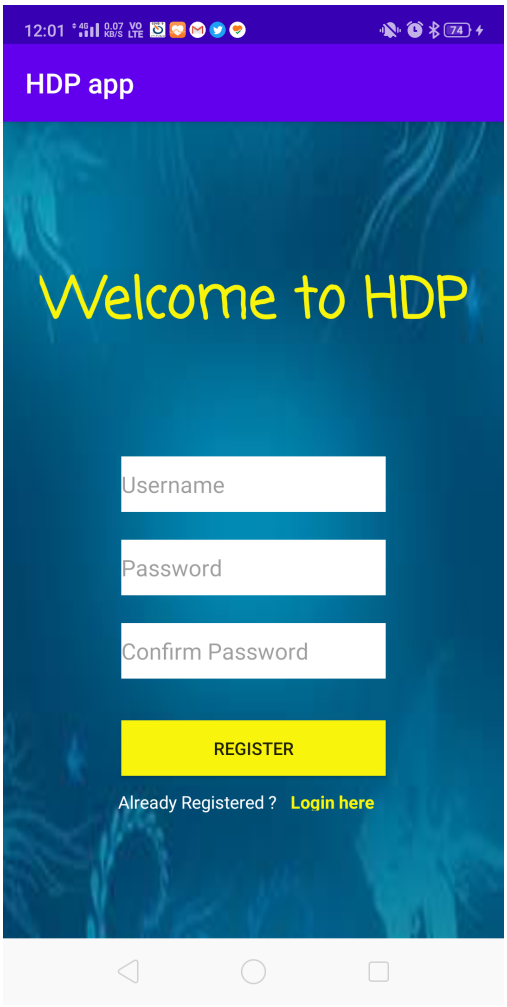



Figure 7.8: Register Activity

### 7.4.2 Login page

By using this page user can login into their account using username and password.

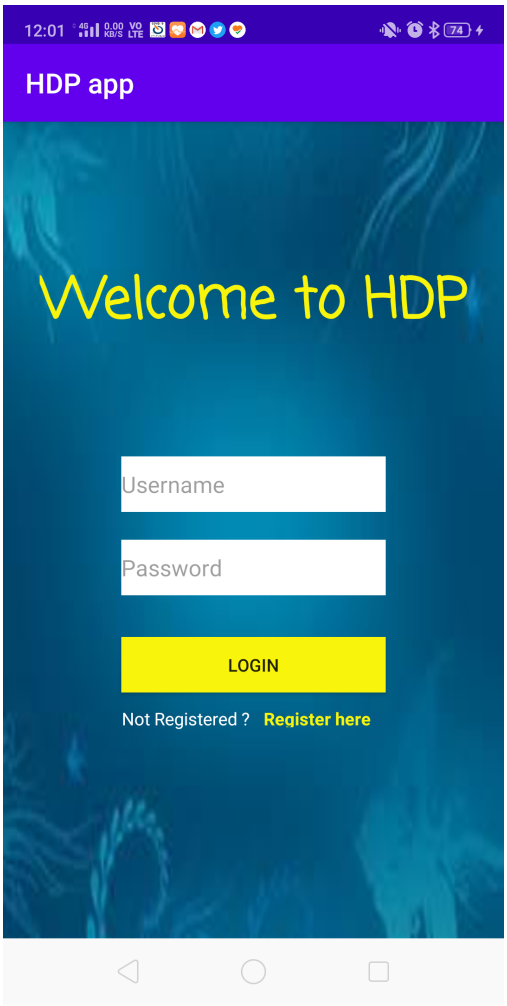



Figure 7.9: Login Activity

### 7.4.3 Home Activity page

In this page user can able to test patient report and can be able to successfully log out from account

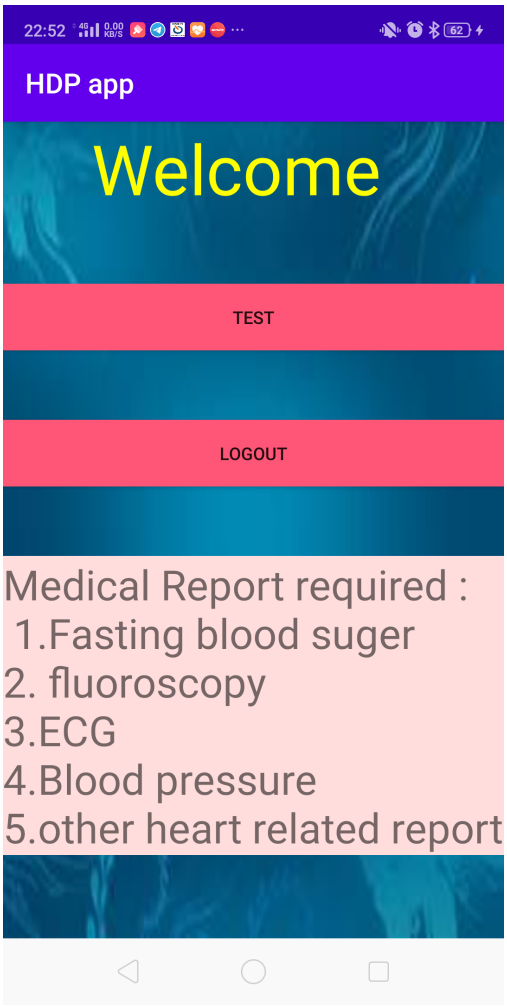



Figure 7.10: Home Activity

### 7.4.4 Input activity

Using this activity user can collect patient?s medical data for testing and prediction of heart disease. Total 13 Input activity which are used to collect patient medical data.

### 7.4.5 Input age activity

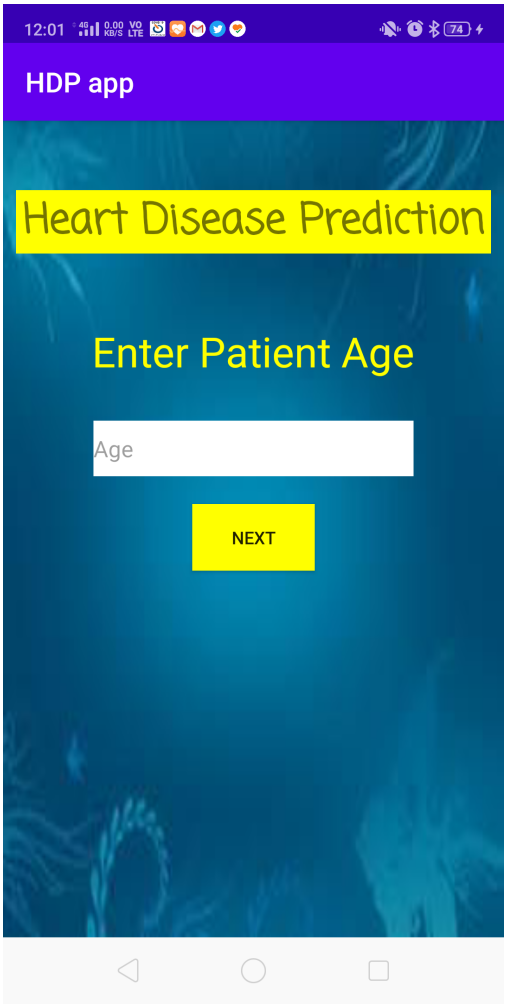



Figure 7.11: Input Age Activity

### 7.4.6 Input Gender activity

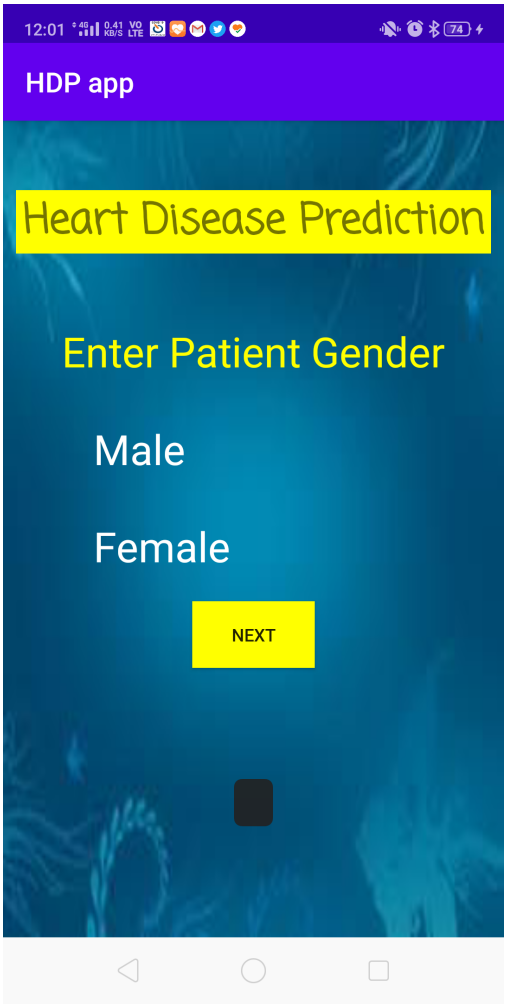



Figure 7.12: Input Gender Activity

### 7.4.7 Chest Pain Details

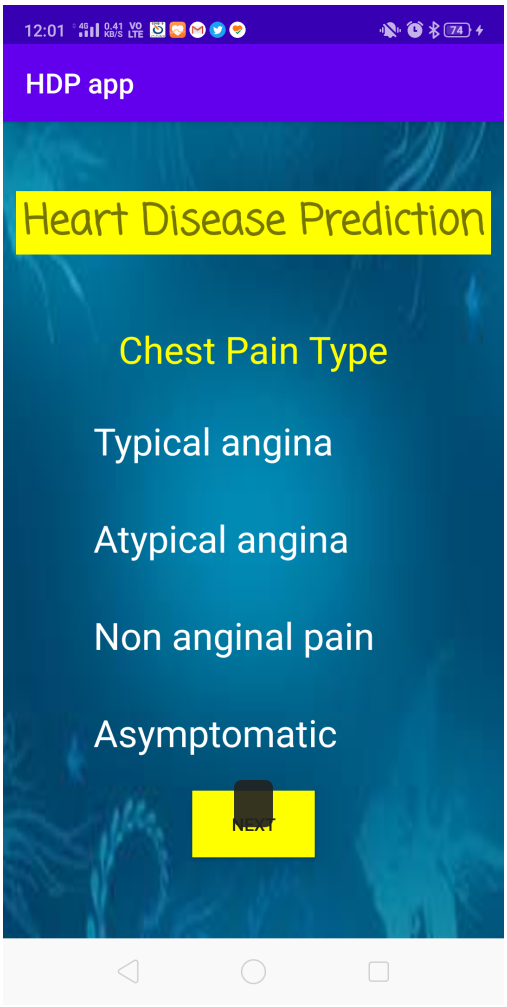



Figure 7.13: Chest Pain Details

### 7.4.8 Input blood pressure activity

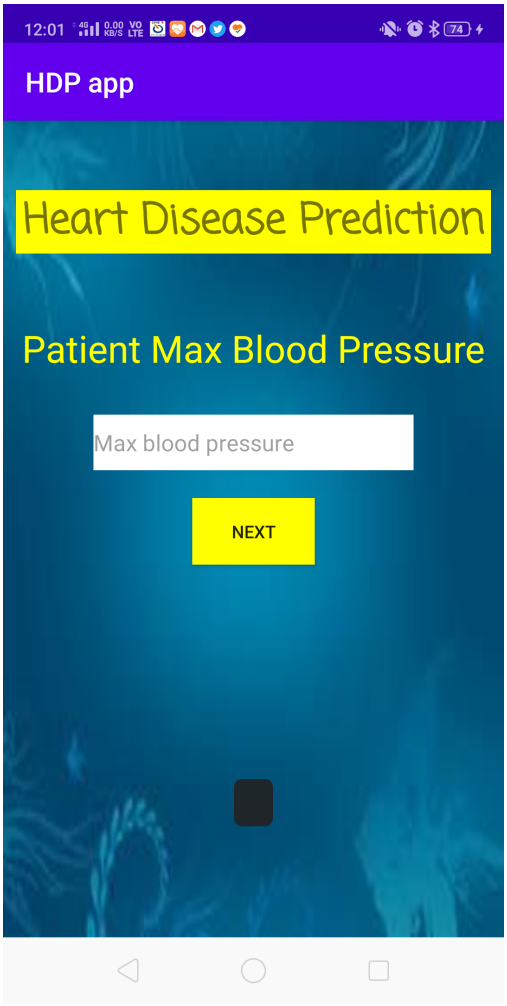



Figure 7.14: Input blood pressure activity

### 7.4.9 Input cholesterol activity


Figure 7.15: Input cholesterol activity

### 7.4.10 Input blood sugar activity

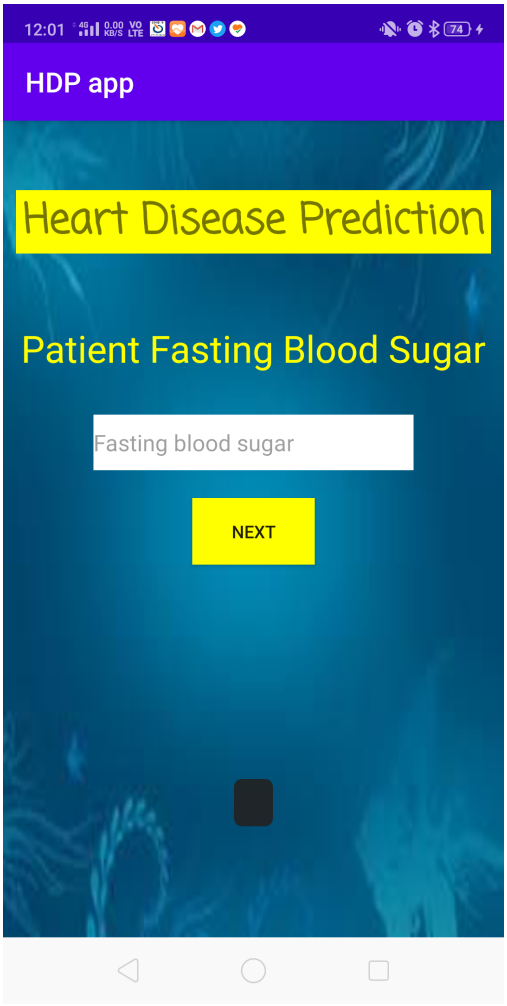



Figure 7.16: Input blood sugar activity

### 7.4.11 Input ECG activity

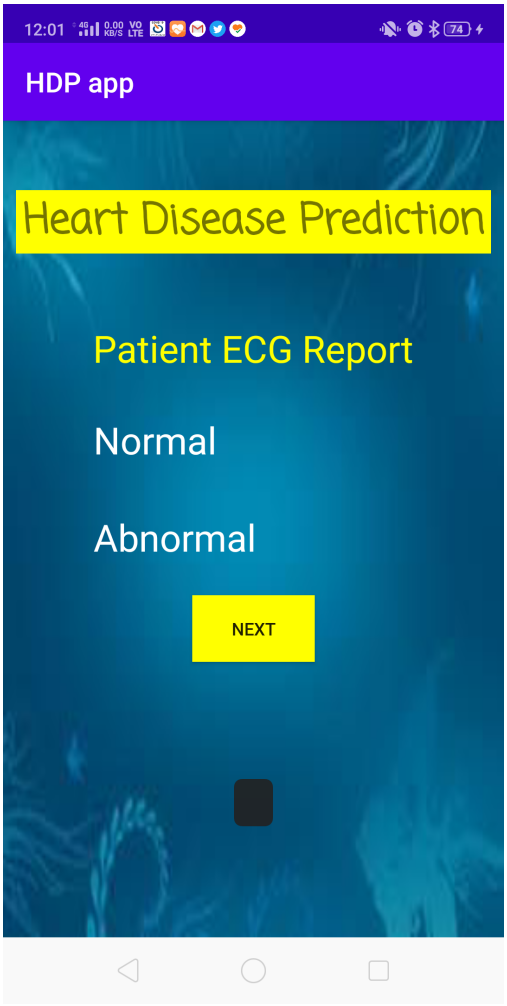



Figure 7.17: Input ECG activity

### 7.4.12 Input heart rate activity

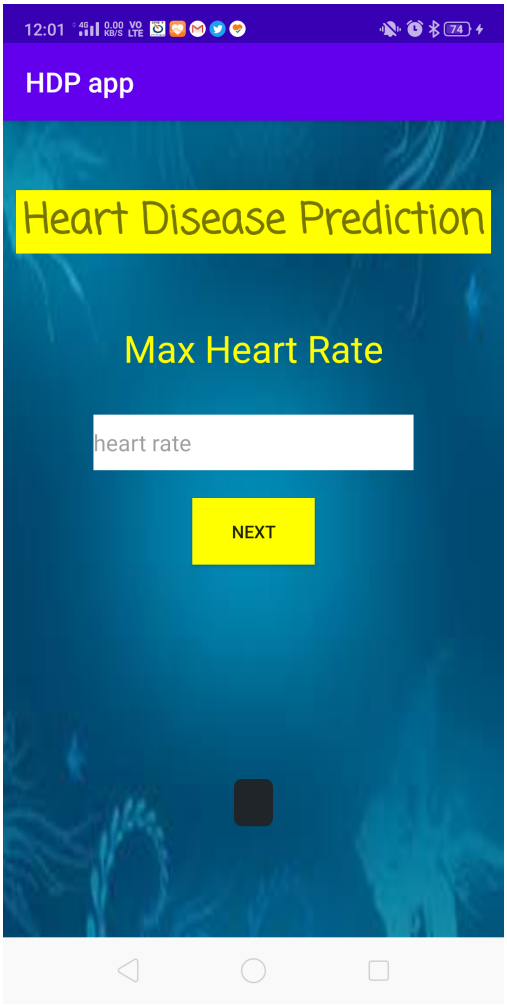



Figure 7.18: Input heart rate activity

### 7.4.13 Input angina excercise activity

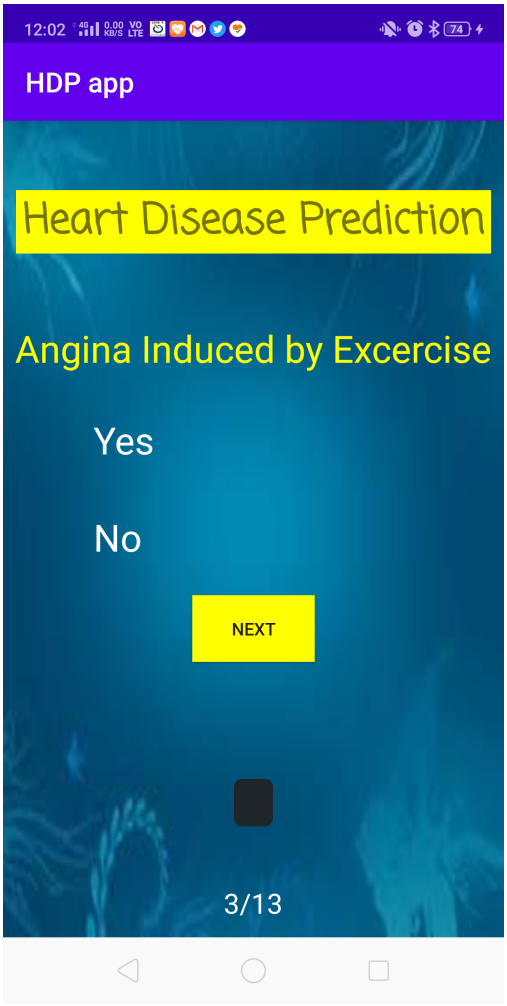



Figure 7.19: Input angina excercise activity

### 7.4.14 Input ST depression activity

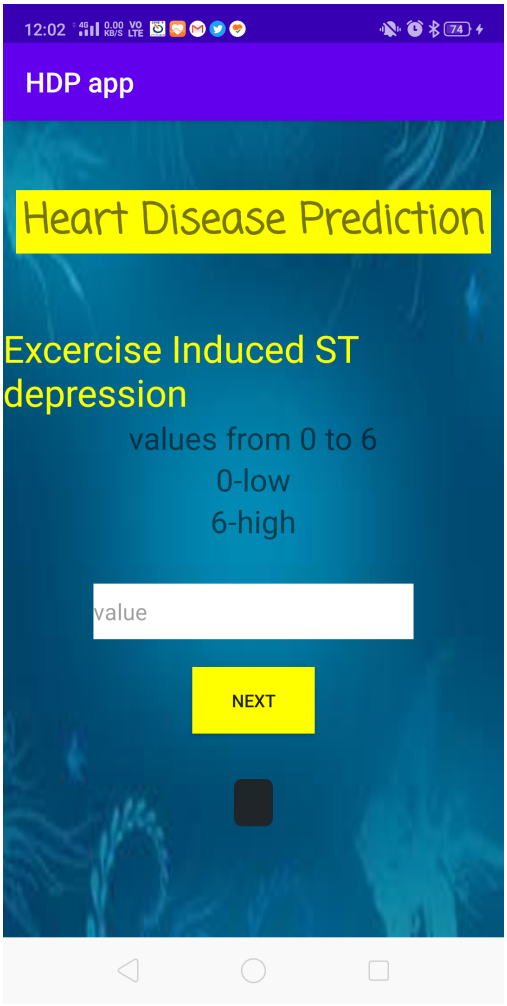



Figure 7.20: Input ST depression activity

### 7.4.15 Input Slope activity

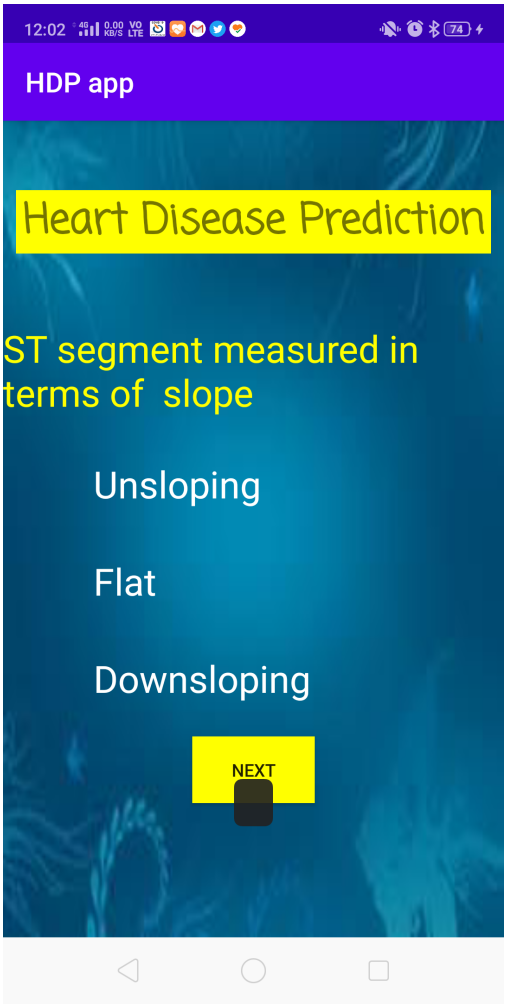



Figure 7.21: Input Slope activity

### 7.4.16 Input Fluoroscopy activity

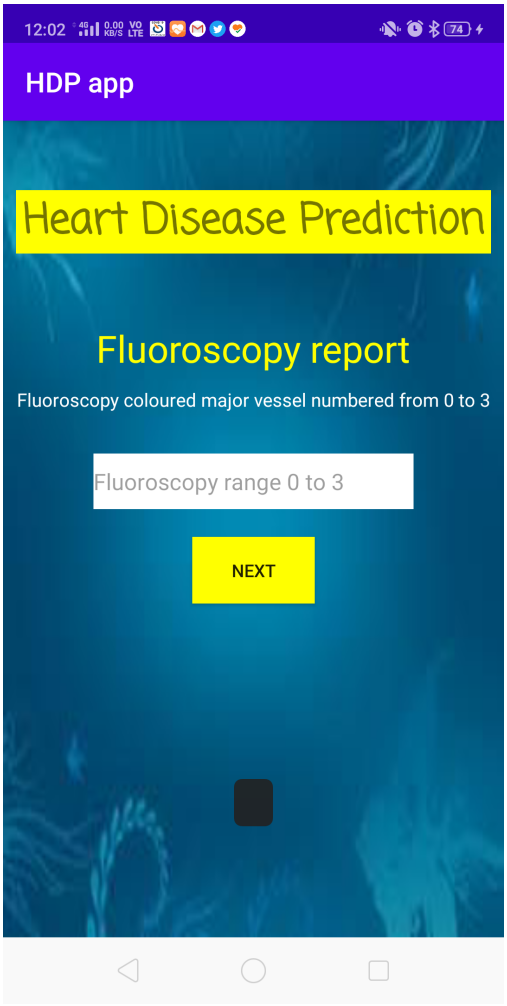



Figure 7.22: Input Fluoroscopy activity

### 7.4.17 Input heart status activity

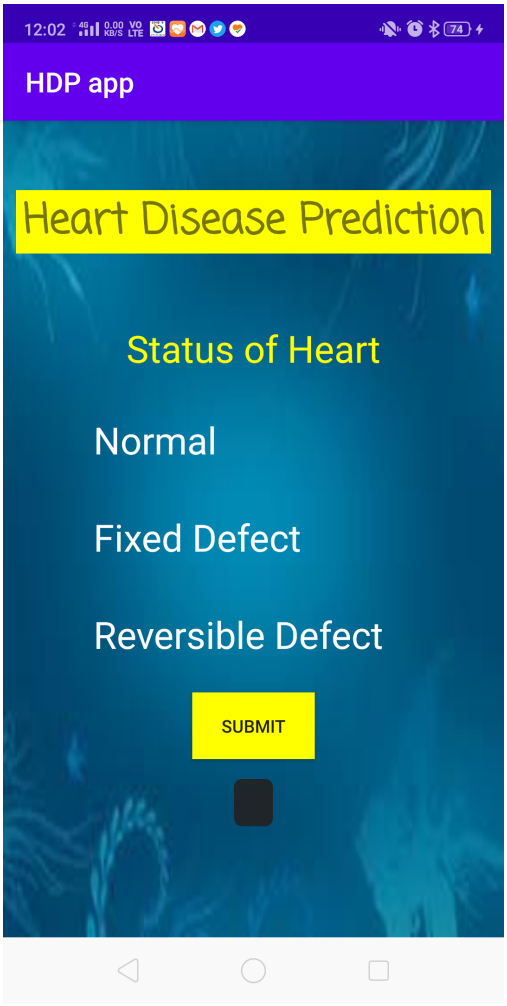



Figure 7.23: Input heart status activity

### 7.4.18 Final Activity

All collected input values from user is send to the server using using https post method and JSON for prediction of heart disease. TEST button is used to send all input values to the server, going to the next Result activity


Figure 7.24: Final activity

### 7.4.19 Result Activity

Result activity retrieved output result from server and show it to the user.

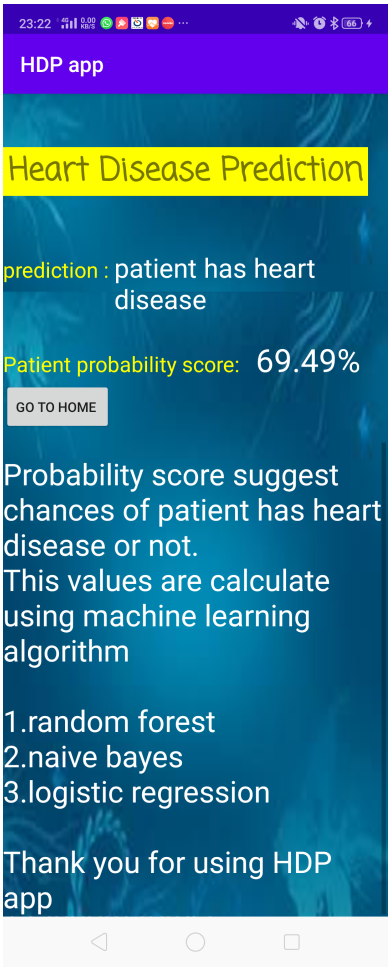



Figure 7.25: Result activity

# Chapter 8

# OTHER SPECIFICATION

## 8.1 Advantages

1. Proposed system maintain its accuracy in case of missing input values.
2. Proposed system accuracy is 94.4
3. Logistic Regression
4. Easy user interface
5. Portable

## 8.2 Limitations

1. System accuracy is depend on patient?s clinical report
2. Patient have to rely on doctor for diagnosis of heart disease as it required necessary medical report like fluoroscopy, ECG, blood pressure.
3. Proposed system does not detect type of heart disease. It predicts patient have heart disease or not.

# Chapter 9

# CONCLUSION

Finding the processing of raw healthcare data of heart information will help in the long term saving of human lives and early detection of abnormalities in heart conditions. Proposed system used machine learning techniques to process raw data and provide a new health care system to predict heart disease. Diagnosis of heart disease is challenging and very important in the medical field. The premature deaths can be reduced if the heart disease is detected at the early stages and preventative measures are taken. This proposed system used Cleveland dataset to predict heart disease. This system combines the characteristics of 4 different machine learning algorithms Random Forest (RF), Naive Bayes (NB) and Logistic Regression (LG) with the help of Voting Classifier (VC). Proposed system proved to be quite accurate in the prediction of heart disease. The future course of this research can be performed with diverse mixtures of machine learning techniques to better prediction techniques. Furthermore, new feature selection methods can be developed to get a broader perception of the significant features to increase the performance of heart disease prediction.

# Chapter 10

# REFRENCES

1. Senthilkumar Mohan Chandrasegar Thirumalai1 and Gautam Srivastava "Effective heart disease prediction using hybrid machine learning techniques" IEEE volume 7, page no.81542-81554, 2019.

2. S. Sreejith, S. Rahul and R.C. Jisha "a real time patient monitoring system for heart disease prediction system using random forest" Springer International Publishing Switzerland , page no 485-500, 2016.

3. M. A. Jabbar1, B. L. Deek- Shatulu2 and Priti Chandra3 "Intelligent heart disease prediction disease prediction system using random forest and evolutionary approach" Springer Nature Singapore Pte Ltd. page no 613-624, 2017.

4. Dhanashree S. Medhekarl, Mayur P. Bote and Shruti D. Deshmukh "Heart disease prediction system using Naive Bayes" International Journal of Engineering and Advanced Technology (IJEAT) page no. 659-662, Feb. 2019.

5. Reddy Prasad, Pidaparthi Anjali, S. Adil and N. Deepa "heart disease prediction using logistic regression algorithm using machine learning" International journal of enhanced research in science technology & engineering page no. 1-5, March 2013.