HEART DISEASE PREDICTION USING MACHINE LEARNING ALGORITHMS

A Project report submitted in partial fulfilment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE ENGINEERING

Submitted by

GUNTURU DEEPTHI	-	317126510140
CHERUKURI SHIVANI	-	317126510133
KORUPROLU NAGAVINITH	-	317126510148
KESUBOYINA HANUDEEP	-	317126510147

Under the guidance of

Mrs. G.PRANITHA

Assistant Professor

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

ANIL NEERUKONDA INSTITUTE OF TECHNOLOGY AND SCIENCES

(UGC AUTONOMOUS)

(*Permanently Affiliated to AU, Approved by AICTE and Accredited by NBA & NAAC with 'A' Grade*)

Sangivalasa, Bheemili mandal, Visakhapatnam dist.(A.P)

2017-2021

ACKNOWLEDGEMENT

We would like to express our deep gratitude to our project guide **G PRANITHA** Assistant Professor, Department of Computer Science and Engineering, ANITS, for her guidance with unsurpassed knowledge and immense encouragement. We are grateful to **DR R SIVARANJANI**, Head of the Department, Computer Science and Engineering, for providing us with the required facilities for the completion of the project work.

We are very much thankful to the **Principal and Management, ANITS, Sangivalasa,** for their encouragement and cooperation to carry out this work.

We express our thanks to Project Coordinator **Dr.V.Usha Bala**, for her continuous support and encouragement. We thank all **teaching faculty** of Department of CSE, whose suggestions during reviews helped us in accomplishment of our project. We would like to thank **S.Sajahan** of the Department of CSE, ANITS for providing great assistance in accomplishment of our project.

We would like to thank our parents, friends, and classmates for their encouragement throughout our project period. At last but not the least, we thank everyone for supporting us directly or indirectly in completing this project successfully.

PROJECT STUDENTS

317126510140	G. Deepthi
317126510133	Ch. Shivani
317126510148	K. Vinith
317126510147	K. Hanudeep Kumar

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

ANIL NEERUKONDA INSTITUTE OF TECHNOLOGY AND SCIENCES
(UGC AUTONOMOUS)
(*Affiliated to AU, Approved by AICTE and Accredited by NBA & NAAC with 'A' Grade*)
Sangivalasa, Bheemili mandal, Visakhapatnam dist.(A.P)

PRAGNANAM BRAHMA
ANITS

CERTIFICATE

This is to certify that the project report entitled "**HEART DISEASE PREDICTION USING MACHINE LEARNING ALGORITHMS"** submitted by **G.Deepthi (317126510140), Ch.Shivani (317126510133), K.Naga Vinith (317126510148), K.Hanudeep (317126510147)** in partial fulfilment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science Engineering** of Anil Neerukonda Institute of technology and sciences (A), Visakhapatnam is a record of bonafide work carried out under my guidance and supervision.

Gprani

Project Guide

G.PRANITHA
ASSISTANT PROFESSOR
Department of CSE
ANITS

Head of the Department

Dr.R.SIVARANJANI
Department of CSE
ANITS

DECLARATION

We, **GUNTURU DEEPTHI, CHERUKURI SHIVANI**, **KORUPROLU NAGA VINITH, KESUBOYINA HANUDEEP** of final semester B.Tech., in the department of Computer Science and Engineering from ANITS, Visakhapatnam, hereby declare that the project work entitled **HEART DISEASE PREDICTION USING MACHINE LEARNING ALGORITHMS** is carried out by us and submitted in partial fulfilment of the requirements for the award of **Bachelor of Technology in Computer Science Engineering** , under Anil Neerukonda Institute of Technology & Sciences (A) during the academic year 2017-2021 and has not been submitted to any other university for the award of any kind of degree.

Gunturu Deepthi	-	317126510140
Cherukuri Shivani	-	317126510133
Koruprolu Nagavinith	-	317126510148
Kesuboyina Hanudeep	-	317126510147

ABSTRACT

Machine Learning is used across many ranges around the world. The healthcare industry is no exclusion. Machine Learning can play an essential role in predicting 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 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 and we propose an ensemble classifier which perform hybrid classification by taking strong and weak classifiers since it can have multiple number of samples for training and validating the data so we perform the analysis of existing classifier and proposed classifier like Ada-boost and XG-boost which can give the better accuracy and predictive analysis..

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

CONTENTS

LIST OF FIGURES

LIST OF SYMBOLS

TP	Number of people with heart diseases.
TN	Number of people with heart diseases and no heart diseases.
FP	Number of people with no heart diseases.
FN	Number of people with no heart diseases and with heart diseases.
f(x)	Output between the 0 and 1 value.
e	Base of the natural logarithm.
x	Input to the function.

LIST OF TABLES

LIST OF ABBREVATIONS

ML	Machine Learning
AI	Artificial Intelligence
NN	Neural Networks
SVM	Support Vector Machine
XG	Extreme Gradient

CHAPTER 1

INTRODUCTION

According to the World Health Organization, every year 12 million deaths occur worldwide due to Heart Disease. Heart disease is one of the biggest causes of morbidity and mortality among the population of the world. Prediction of cardiovascular disease is regarded as one of the most important subjects in the section of data analysis. The load of cardiovascular disease is rapidly increasing all over the world from the past few years. Many researches have been conducted in attempt to pinpoint the most influential factors of heart disease as well as accurately predict the overall risk. Heart Disease is even highlighted as a silent killer which leads to the death of the person without obvious symptoms. The early diagnosis of heart disease plays a vital role in making decisions on lifestyle changes in high-risk patients and in turn reduces the complications.

Machine learning proves to be effective in assisting in making decisions and predictions from the large quantity of data produced by the health care industry. This project aims to predict future Heart Disease by analyzing data of patients which classifies whether they have heart disease or not using machine-learning algorithm. Machine Learning techniques can be a boon in this regard. Even though heart disease can occur in different forms, there is a common set of core risk factors that influence whether someone will ultimately be at risk for heart disease or not. By collecting the data from various sources, classifying them under suitable headings & finally analysing to extract the desired data we can say that this technique can be very well adapted to do the prediction of heart disease.

1.1 MOTIVATION FOR THE WORK

The main motivation of doing this research is to present a heart disease prediction model for the prediction of occurrence of heart disease. Further, this research work is aimed towards identifying the best classification algorithm for identifying the possibility of heart disease in a patient. This work is justified by performing a comparative study and analysis using three classification algorithms namely Naïve Bayes, Decision Tree, and Random Forest are used at different levels of evaluations. Although these are commonly used machine learning algorithms, the heart disease prediction is a vital task involving highest possible accuracy. Hence, the three algorithms are evaluated at numerous levels and types of evaluation strategies. This will provide researchers and medical practitioners to establish a better.

1.2 PROBLEM STATEMENT

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

CHAPTER 2

LITERATURE SURVEY

With growing development in the field of medical science alongside machine learning various experiments and researches has been carried out in these recent years releasing the relevant significant papers.

[1] Purushottam ,et ,al proposed a paper "Efficient Heart Disease Prediction System" using hill climbing and decision tree algorithms .They used 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 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%.

[2] Santhana Krishnan. J ,et ,al proposed a paper "Prediction of Heart Disease Using Machine Learning Algorithms" using decision tree and Naive Bayes algorithm for prediction of heart disease. In decision tree algorithm the tree is built using certain conditions which gives True or False decisions. The algorithms like SVM, KNN are results based on vertical or horizontal split conditions depends on dependent variables. But decision tree for a tree like structure having root node, leaves and branches base on the decision made in each of tree Decision tree also help in the understating the importance of the attributes in the dataset. They have also used Cleveland data set. Dataset splits in 70% training and 30% testing by using some methods. This algorithm gives 91% accuracy. The second algorithm is Naive Bayes, which is used for classification. It can handle complicated, nonlinear, dependent data so it is found suitable for heart disease dataset as this dataset is also complicated, dependent and nonlinear in nature. This algorithm gives an 87% accuracy.

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

Some analysis has been led to think about the execution of prescient data mining strategy on the same dataset, and the result decided that Decision Tree has highest accuracy than Bayesian classifier.

[4] Aditi Gavhane et al proposed a paper “Prediction of Heart Disease Using Machine Learning”, in which training and testing of dataset is 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 layers between these two input and output layers. Through hidden layers each input node is connected to output layer. This connection is assigned with some random weights. The other input is called bias which is assigned with weight based on requirement the connection between the nodes can be feedforwarded or feedback.

[5] 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 neighbour, Decision tree and Naïve Bayes. Other unique characterization-based strategies utilized are packing calculation, Part thickness, consecutive negligible streamlining and neural systems, straight Kernel selfarranging guide and SVM (Bolster Vector Machine).

[6] 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 the heart disease among people different neural systems and data mining techniques are used.

[7] Abhay Kishore et alproposed “Heart Attack Prediction Using Deep Learning” in which heart attack prediction system by using Deep learning techniques and to predict the probable aspects of heart related infections of the patient Recurrent Neural System is used. This model uses deep learning and data mining to give the best precise model and least blunders. This paper acts as strong reference model for another type of heart attack prediction models

[8] Senthil Kumar Mohan et al, proposed “Effective Heart Disease Prediction Using Hybrid Machine Learning Techniques” in which their main objective is to improve

exactness in cardiovascular problems. The algorithms used are KNN, LR, SVM, NN to produce an improved exhibition level with a precision level of 88.7% through the prediction model for heart disease with hybrid random forest with linear model(HRFLM).

[9] Anjan N. Repaka et al, proposed a model stated the performance of prediction for two classification models, which is analyzed and compared to previous work. The experimental results show that accuracy is improved in finding the percentage of risk prediction of our proposed method in comparison with other models.

[10] Aakash Chauhan et al, proposed “Heart Disease Prediction using Evolutionary Rule Learning”. Data is directly retrieved from electronic records that reduce the manual tasks. The amount of services are decreased and shown major number of rules helps within the best prediction of heart disease. Frequent pattern growth association mining is performed on patient’s dataset to generate strong association.

CHAPTER 3 METHODOLOGY

3.1 EXISTING SYSTEM

Heart disease is even being highlighted as a silent killer which leads to the death of a person without obvious symptoms. The nature of the disease is the cause of growing anxiety about the disease & its consequences. Hence continued efforts are being done to predict the possibility of this deadly disease in prior. So that various tools & techniques are regularly being experimented with to suit the present-day health needs. Machine Learning techniques can be a boon in this regard. Even though heart disease can occur in different forms, there is a common set of core risk factors that influence whether someone will ultimately be at risk for heart disease or not. By collecting the data from various sources, classifying them under suitable headings & finally analysing to extract the desired data we can conclude. This technique can be very well adapted to the do the prediction of heart disease. As the well-known quote says “Prevention is better than cure”, early prediction & its control can be helpful to prevent & decrease the death rates due to heart disease.

3.2 PROPOSED SYSTEM

The working of the system starts with the collection of data and selecting the important attributes. Then the required data is preprocessed into the required format. The data is then divided into two parts training and testing data. The algorithms are applied and the model is trained using the training data. The accuracy of the system is obtained by testing the system using the testing data. This system is implemented using the following modules.

1.) Collection of Dataset

2.) Selection of attributes

3.) Data Pre-Processing

4.) Balancing of Data

5.) Disease Prediction

3.2.1 Collection of dataset

Initially, we collect a dataset for our heart disease prediction system. After the collection of the dataset, we split the dataset into training data and testing data. The training dataset is used for prediction model learning and testing data is used for evaluating the prediction model. For this project, 70% of training data is used and 30% of data is used for testing. The dataset used for this project is Heart Disease UCI. The dataset consists of 76 attributes; out of which, 14 attributes are used for the system.

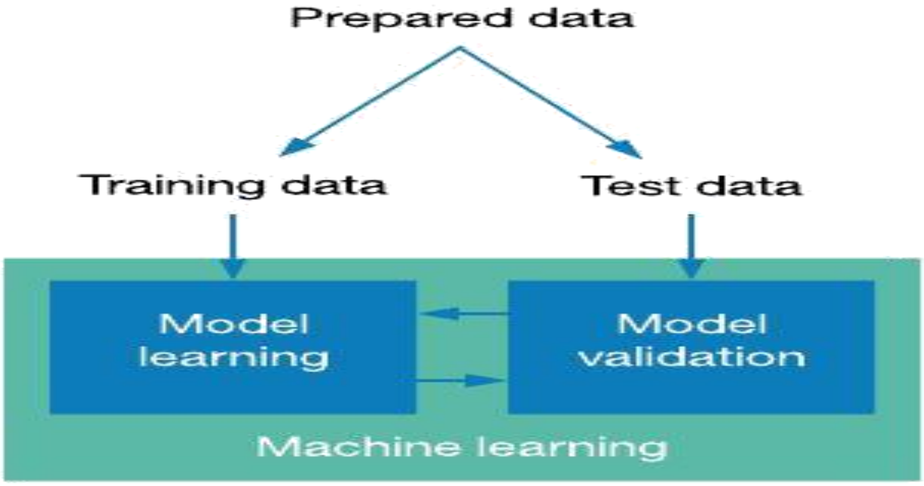


Figure: Collection of Data

3.2.2 Selection of attributes

Attribute or Feature selection includes the selection of appropriate attributes for the prediction system. This is used to increase the efficiency of the system. Various attributes of the patient like gender, chest pain type, fasting blood pressure, serum cholesterol, exang, etc are selected for the prediction. The Correlation matrix is used for attribute selection for this model.

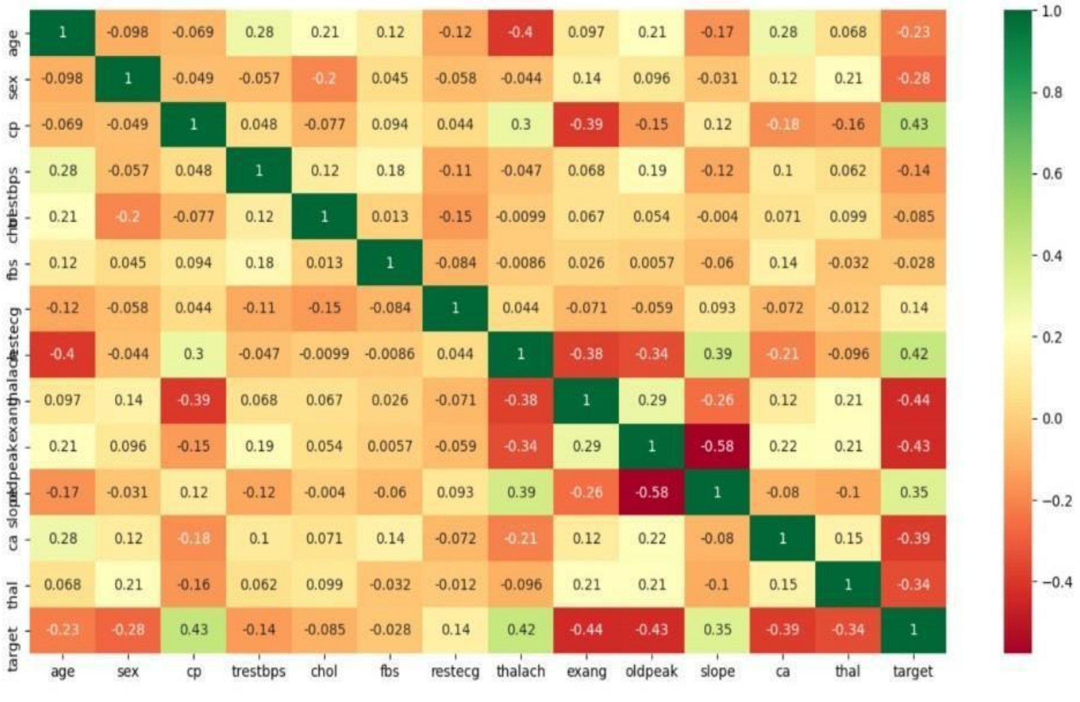


Figure: Correlation matrix

3.2.3 Pre-processing of Data

Data pre-processing is an important step for the creation of a machine learning model. Initially, data may not be clean or in the required format for the model which can cause misleading outcomes. In pre-processing of data, we transform data into our required format. It is used to deal with noises, duplicates, and missing values of the dataset. Data pre-processing has the activities like importing datasets, splitting datasets, attribute scaling, etc. Preprocessing of data is required for improving the accuracy of the model.

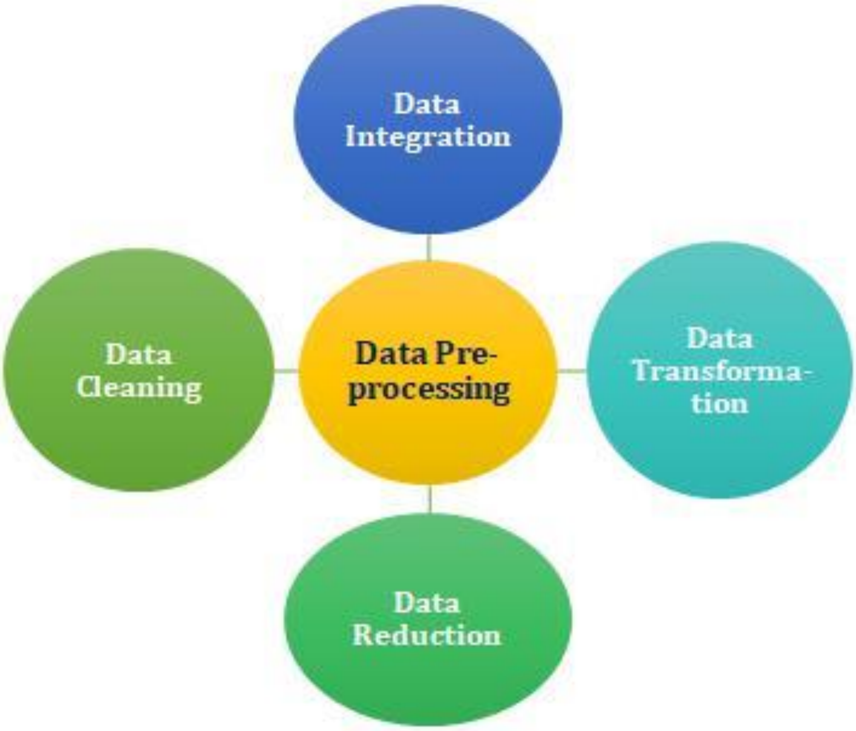


Figure: Data Pre-processing

3.2.4 Balancing of Data

Imbalanced datasets can be balanced in two ways. They are Under Sampling and Over Sampling

(a) Under Sampling:

In Under Sampling, dataset balance is done by the reduction of the size of the ample class. This process is considered when the amount of data is adequate. (b) Over Sampling:

In Over Sampling, dataset balance is done by increasing the size of the scarce samples. This process is considered when the amount of data is inadequate.

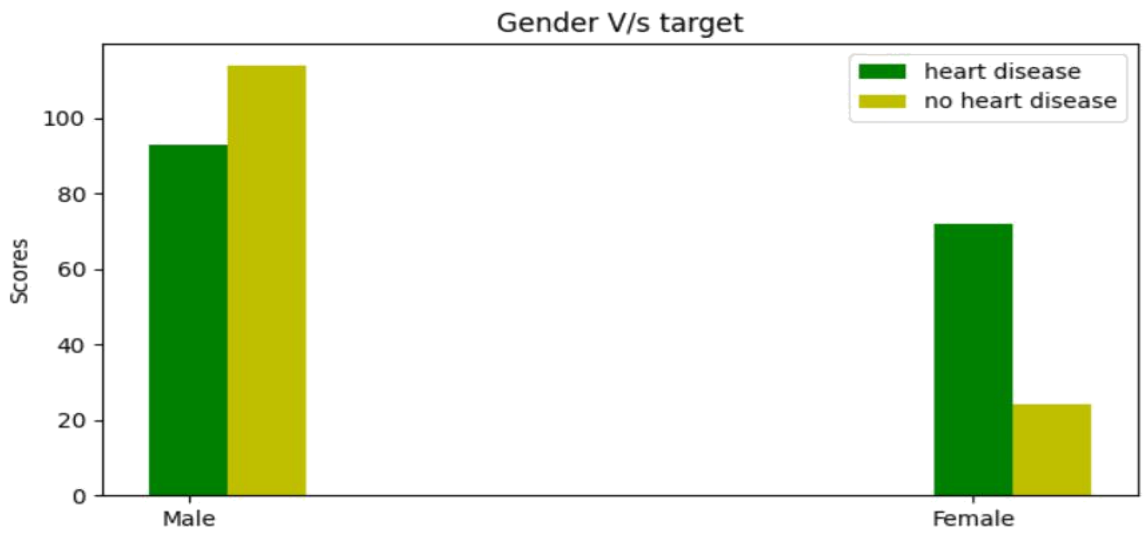


Figure: Data Balancing

3.2.5 Prediction of Disease

Various machine learning algorithms like SVM, Naive Bayes, Decision Tree, Random Tree, Logistic Regression, Ada-boost, Xg-boost are used for classification. Comparative analysis is performed among algorithms and the algorithm that gives the highest accuracy is used for heart disease prediction.

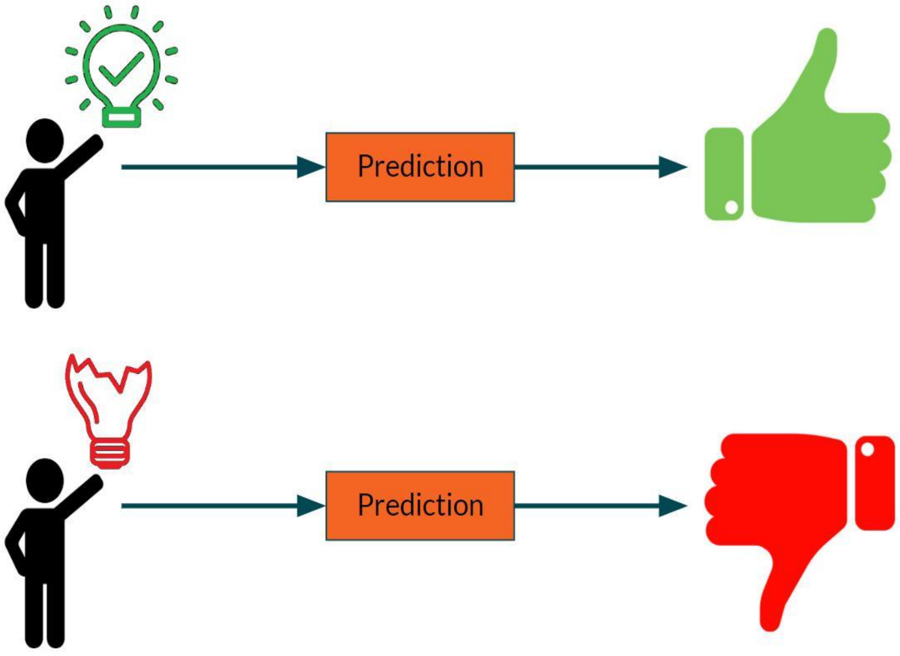


Figure: Prediction of Disease

CHAPTER 4

WORKING OF SYSTEM

4.1 SYSTEM ARCHITECTURE

The system architecture gives an overview of the working of the system.

The working of this system is described as follows:

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

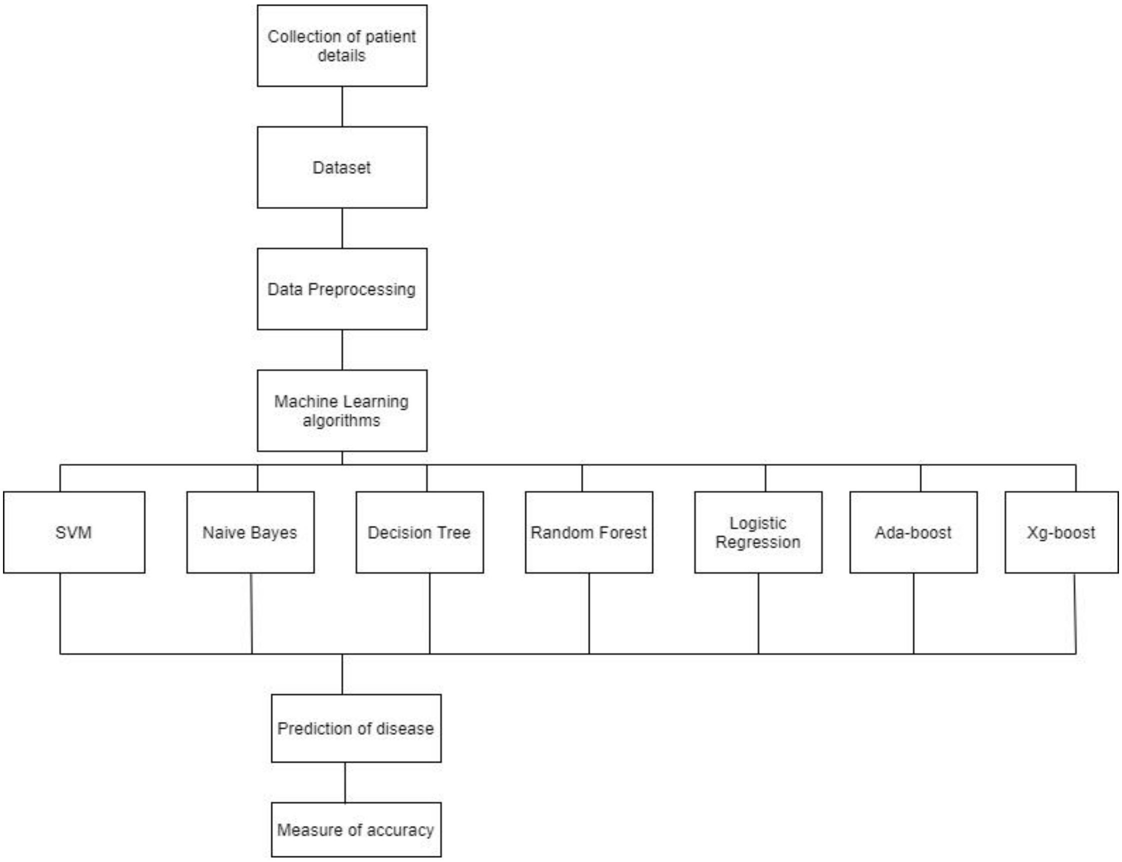


Figure:. SYSTEM ARCHITECTURE

4.2 MACHINE LEARNING

In machine learning, classification refers to a predictive modeling problem where a class label is predicted for a given example of input data.

- **Supervised Learning**

Supervised learning is the type of machine learning in which machines are trained using well "labelled" training data, and on the basis of that data, machines predict the output. The labelled data means some input data is already tagged with the correct output.

In supervised learning, the training data provided to the machines work as the supervisor that teaches the machines to predict the output correctly. It applies the same concept as a student learns in the supervision of the teacher.

Supervised learning is a process of providing input data as well as correct output data to the machine learning model. The aim of a supervised learning algorithm is to find a mapping function to map the input variable(x) with the output variable(y).

- **Unsupervised learning**

Unsupervised learning cannot be directly applied to a regression or classification problem because unlike supervised learning, we have the input data but no corresponding output data. The goal of unsupervised learning is to find the underlying structure of dataset, group that data according to similarities, and represent that dataset in a compressed format.

- Unsupervised learning is helpful for finding useful insights from the data.
- Unsupervised learning is much similar to how a human learns to think by their own experiences, which makes it closer to the real AI.
- Unsupervised learning works on unlabeled and uncategorized data which make unsupervised learning more important.
- In real-world, we do not always have input data with the corresponding output so to solve such cases, we need unsupervised learning.

- **Reinforcement learning**

Reinforcement learning is an area of Machine Learning. It is about taking suitable action to maximize reward in a particular situation. It is employed by various software and machines to find the best possible behaviour or path it should take in a specific situation. Reinforcement learning differs from supervised learning in a way that in supervised learning the training data has the answer key with it so the model is trained with the correct answer itself whereas in reinforcement learning, there is no answer but the reinforcement agent decides what to do to perform the given task. In the absence of a training dataset, it is bound to learn from its experience.

4.3 ALGORITHMS

4.3.1 SUPPORT VECTOR MACHINE (SVM):

Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems. However, primarily, it is used for Classification problems in Machine Learning.

The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane.SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called support vectors, and hence the algorithm is termed as Support Vector Machine.

Support vector machines (SVMs) are powerful yet flexible supervised machine learning algorithms which are used both for classification and regression. But generally, they are used in classification problems. In the 1960s, SVMs were first introduced but later they got refined in 1990. SVMs have their unique way of implementation as compared to other machine learning algorithms. Lately, they are extremely popular because of their ability to handle multiple continuous and categorical variables.

The followings are important concepts in SVM -

Support Vectors - Data Points that are closest to the hyperplane are called support vectors. Separating line will be defined with the help of these data points.

Hyperplane - As we can see in the above diagram, it is a decision plane or space which is divided between a set of objects having different classes.

Margin - It may be defined as the gap between two lines on the closest data points of different classes. It can be calculated as the perpendicular distance from the line to the support vectors. Large margin is considered as a good margin and small margin is considered as a bad margin.

Types of SVM:

SVM can be of two types:

- **Linear SVM:** Linear SVM is used for linearly separable data, which means if a dataset can be classified into two classes by using a single straight line, then such data is termed as linearly separable data, and classifier is used called as Linear SVM classifier.
- **Non-linear SVM:** Non-Linear SVM is used for non-linearly separated data, which means if a dataset cannot be classified by using a straight line, then such data is termed as non-linear data and classifier used is called as Non-linear SVM classifier.

The objective of the support vector machine algorithm is to find a hyperplane in an N-dimensional space (N - the number of features) that distinctly classifies the data points.

The advantages of support vector machines are:

- Effective in high dimensional spaces.
- Still effective in cases where the number of dimensions is greater than the number of samples.
- Uses a subset of training points in the decision function (called support vectors), so it is also memory efficient.
- Versatile: different kernel functions can be specified for the decision function. Common kernels are provided, but it is also possible to specify custom kernels.

The disadvantages of support vector machines include:

- If the number of features is much greater than the number of samples, avoid over-fitting in choosing Kernel functions and regularization term is crucial.

SVMs do not directly provide probability estimates, these are calculated using an expensive five-fold cross-validation.

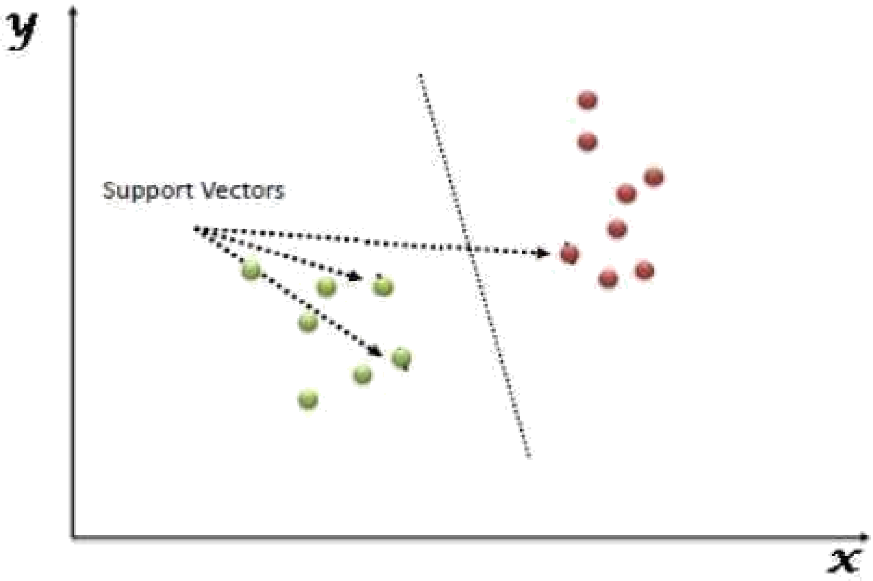


Figure: Support Vector Machine

4.3.2 NAIVE BAYES ALGORITHM:

Naive Bayes algorithm is a supervised learning algorithm, which is based on Bayes theorem and used for solving classification problems.It is mainly used in text classification that includes a high-dimensional training dataset.

Naive Bayes Classifier is one of the simple 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 Naïve Bayes Algorithm are spam filtration, Sentimental analysis, and classifying articles.

It is a classification technique based on Bayes' Theorem with an assumption of independence among predictors. In simple terms, a Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature.

The Naive Bayes model is easy to build and particularly useful for very large data sets. Along with simplicity, Naive Bayes is known to outperform even highly sophisticated classification methods.

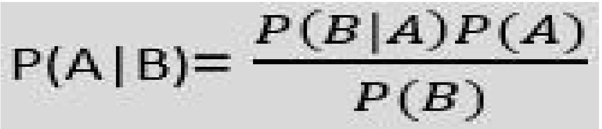
The Naive Bayes algorithm is comprised of two words Naive and Bayes, Which can be described as:

- **Naive:** It is called Naive because it assumes that the occurrence of a certain feature is independent of the occurrence of other features. Such as if the fruit is identified on the basis of color, shape, and taste, then red, spherical, and sweet fruit is recognized as an apple. Hence each feature individually contributes to identify that it is an apple without depending on each other.
- **Bayes:** It is called Bayes because it depends on the principle of Bayes' Theorem.

Bayes’s theorem:

Bayes' theorem is also known as Bayes' Rule or Bayes' law, which is used to determine the probability of a hypothesis with prior knowledge. It depends on the conditional probability.

The formula for Bayes' theorem is given as:



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

Where,

P(A|B) is Posterior probability:Probability of hypothesis A on the observed event B.

P(B|A) is Likelihood probability: Probability of the evidence given that the probability of a hypothesis is true.

P(A) is Prior Probability: Probability of hypothesis before observing the evidence.

P(B) is Marginal Probability: Probability of Evidence.

Types of Naive Bayes model:

There are three types of Naive Bayes Model, which are given below:

- **Gaussian:** The Gaussian model assumes that features follow a normal distribution. This means if predictors take continuous values instead of discrete, then the model assumes that these values are sampled from the Gaussian distribution.
- **Multinomial:** The Multinomial Naïve Bayes classifier is used when the data is multinomial distributed. It is primarily used for document classification problems, it means a particular document belongs to which category such as

Sports, Politics, education, etc. The classifier uses the frequency of words for the predictors.

- **Bernoulli:** The Bernoulli classifier works similar to the Multinomial classifier, but the predictor variables are the independent Booleans variables. Such as if a particular word is present or not in a document. This model is also famous for document classification tasks.

4.3.3 DECISION TREE ALGORITHM

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

Decision nodes are used to make any decision and have multiple branches, whereas Leaf nodes are the output of those decisions and do not contain any further branches. The decisions or the test are performed on the basis of features of the given dataset. It is a graphical representation for getting all the possible solutions to a problem/decision based on given conditions. It is called a Decision Tree because, similar to a tree, it starts with the root node, which expands on further branches and constructs a tree-like structure. In order to build a tree, we use the CART algorithm, which stands for Classification and Regression Tree algorithm. A Decision Tree simply asks a question, and based on the answer (Yes/No), it further split the tree into subtrees.

The Decision Tree Algorithm belongs to the family of supervised machine learning algorithms. It can be used for both a classification problem as well as for a regression problem.

The goal of this algorithm is to create a model that predicts the value of a target variable, for which the decision tree uses the tree representation to solve the problem in which the leaf node corresponds to a class label and attributes are represented on the internal node of the tree.

There are various algorithms in Machine learning, so choosing the best algorithm for the given dataset and problem is the main point to remember while creating a machine learning model. Below are the two reasons for using the Decision Tree:

- Decision Trees usually mimic human thinking ability while making a decision, so it is easy to understand.
- The logic behind the decision tree can be easily understood because it shows a tree-like structure.

In Decision Tree the major challenge is to identify the attribute for the root node in each level. This process is known as attribute selection. We have two popular attribute selection measures:

1. Information Gain:

When we use a node in a Decision Tree to partition the training instances into smaller subsets, the entropy changes. Information gain is a measure of this change in entropy.

Entropy is the measure of uncertainty of a random variable, it characterizes the impurity of an arbitrary collection of examples.
The higher the entropy the more the information content.

2. Gini Index:

Gini Index is a metric to measure how often a randomly chosen element would be incorrectly identified. It means an attribute with lower Gini index should be preferred. Sklearn supports “Gini” criteria for Gini Index and by default, it takes “gini” value.

The most notable types of Decision Tree algorithms are:-

1. **IDichotomiser 3 (ID3):**

   This algorithm uses Information Gain to decide which attribute is to be used to classify the current subset of the data. For each level of the tree, information gain is calculated for the remaining data recursively.

2. **C4.5:** This algorithm is the successor of the ID3 algorithm. This algorithm uses either Information gain or Gain ratio to decide upon the classifying attribute. It is a direct improvement from the ID3 algorithm as it can handle both continuous and missing attribute values.
3. **Classification and Regression Tree (CART):** It is a dynamic learning algorithm which can produce a regression tree as well as a classification tree depending upon the dependent variable.

Working:

In a Decision Tree, for predicting the class of the given dataset, the algorithm starts from the root node of the tree. This algorithm compares the values of the root attribute with the record (real dataset) attribute and, based on the comparison, follows the branch and jumps to the next node.

For the next node, the algorithm again compares the attribute value with the other sub-nodes and moves further. It continues the process until it reaches the leaf node of the tree. The complete process can be better understood using the below algorithm:

- Step-1: Begin the tree with the root node, says S, which contains the complete dataset.
- Step-2: Find the best attribute in the dataset using Attribute Selection Measure (ASM).
- Step-3: Divide the S into subsets that contains possible values for the best attributes.
- Step-4: Generate the Decision Tree node, which contains the best attribute.
- Step-5: Recursively make new decision trees using the subsets of the dataset created in step -3. Continue this process until a stage is reached where you cannot further classify the nodes and call the final node as a leaf node.

4.3.4 RANDOM FOREST ALGORITHM

Random Forest is a supervised learning algorithm. It is an extension of machine learning classifiers which include the bagging to improve the performance of Decision Tree. It combines tree predictors, and trees are dependent on a random vector which is independently sampled. The distribution of all trees are the same. Random Forests splits nodes using the best among of a predictor subset that are randomly chosen from the node itself, instead of splitting nodes based on the variables. The time complexity of the worst case of learning with Random Forests is O(M(dnlogn)) , where M is the number of growing trees, n is the number of instances, and d is the data dimension.

It can be used both for classification and regression. It is also the most flexible and easy to use algorithm. A forest consists of trees. It is said that the more trees it has, the more robust a forest is. Random Forests create Decision Trees on randomly selected data samples, get predictions from each tree and select the best solution by means of voting. It also provides a pretty good indicator of the feature importance.

Random Forests have a variety of applications, such as recommendation engines, image classification and feature selection. It can be used to classify loyal loan applicants, identify fraudulent activity and predict diseases. It lies at the base of the Boruta algorithm, which selects important features in a dataset.

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 to 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, and it predicts the final output.

The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting.

Assumptions:

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.

Algorithm Steps:

It works in four steps:

- Select random samples from a given dataset.
- Construct a Decision Tree for each sample and get a prediction result from each Decision Tree.
- Perform a vote for each predicted result.
- Select the prediction result with the most votes as the final prediction.

Advantages:

- 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.

Disadvantages:

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

4.3.5 LOGISTIC REGRESSION ALGORITHM

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 the 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, a mouse is obese or not based on itsweight, 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.

Advantages:

Logistic Regression is one of the simplest machine learning algorithms and is easy to implement yet provides great training efficiency in some cases. Also due to these reasons, training a model with this algorithm doesn't require high computation power.

The predicted parameters (trained weights) give inference about the importance of each feature. The direction of association i.e. positive or negative is also given. So we can use Logistic Regression to find out the relationship between the features.

This algorithm allows models to be updated easily to reflect new data, unlike Decision Tree or Support Vector Machine. The update can be done using stochastic gradient descent.

Logistic Regression outputs well-calibrated probabilities along with classification results. This is an advantage over models that only give the final classification as results. If a training example has a 95% probability for a class, and another has a 55% probability for the same class, we get an inference about which training examples are more accurate for the formulated problem.

Disadvantages:

Logistic Regression is a statistical analysis model that attempts to predict precise probabilistic outcomes based on independent features. On high dimensional datasets, this may lead to the model being over-fit on the training set, which means overstating the accuracy of predictions on the training set and thus the model may not be able to predict accurate results on the test set. This usually happens in the case when the model is trained on little training data with lots of features. So on high dimensional datasets, Regularization techniques should be considered to avoid over-fitting (but this makes the model complex). Very high regularization factors may even lead to the model being under-fit on the training data.

Non linear problems can't be solved with logistic regression since it has a linear decision surface. Linearly separable data is rarely found in real world scenarios. So the transformation of non linear features is required which can be done by increasing the number of features such that the data becomes linearly separable in higher dimensions.

Non-Linearly Separable Data:

It is difficult to capture complex relationships using logistic regression. More powerful and complex algorithms such as Neural Networks can easily outperform this algorithm

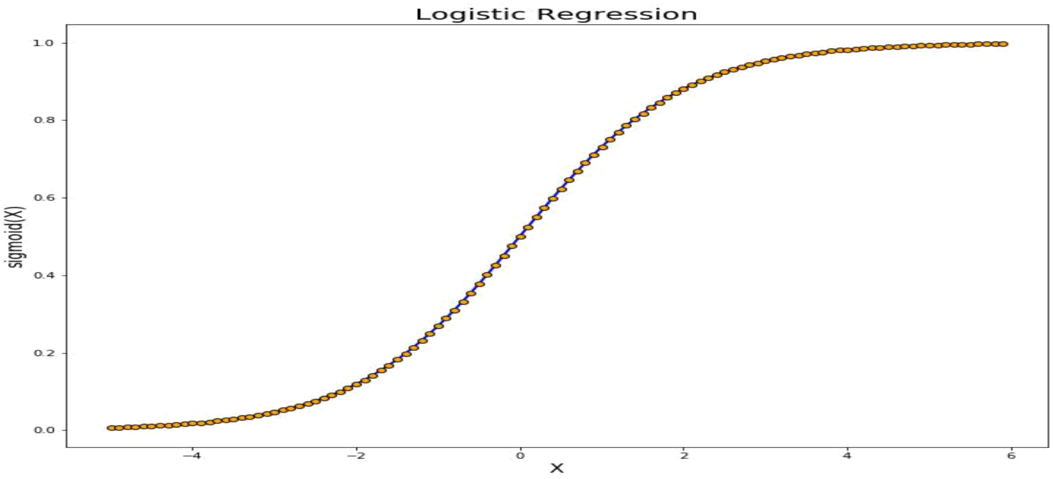


Figure: Logistic Regression

4.3.6 ADABOOST ALGORITHM

Adaboost was the first really successful boosting algorithm developed for the purpose of binary classification. Adaboost is short for Adaptive Boosting and is a very popular boosting technique which combines multiple “weak classifiers” into a single “strong classifier”

Algorithm:

1. Initially, Adaboost selects a training subset randomly.
2. It iteratively trains the Adaboost machine learning model by selecting the training set based on the accurate prediction of the last training.
3. It assigns the higher weight to wrong classified observations so that in the next iteration these observations will get the high probability for classification.
4. Also, it assigns the weight to the trained classifier in each iteration according to the accuracy of the classifier. The more accurate classifier will get high weight.
5. This process iterates until the complete training data fits without any error or until reached to the specified maximum number of estimators.
6. To classify, perform a "vote" across all of the learning algorithms you built

Advantages:

Adaboost has many advantages due to its ease of use and less parameter tweaking when compared with the SVM algorithms. Plus Adaboost can be used with SVM though theoretically, overfitting is not a feature of Adaboost applications, perhaps because the parameters are not optimized jointly and the learning process is slowed due to estimation stage-wise. This link is useful to understand mathematics. The flexible Adaboost can also be used for accuracy improvement of weak classifiers and cases in image/text classification.

Disadvantages:

Adaboost uses a progressively learning boosting technique. Hence high-quality data is needed in examples of Adaboost vs Random Forest. It is also very sensitive to outliers and noise in data requiring the elimination of these factors before using the data. It is also much slower than the XG-boost algorithm.

4.3.7 XGBOOST ALGORITHM

XG-boost is an implementation of Gradient Boosted decision trees. It is a type of Software library that was designed basically to improve speed and model performance. In this algorithm, decision trees are created in sequential form. Weights play an important role in XG-boost. Weights are assigned to all the independent variables which are then fed into the decision tree which predicts results. Weight of variables predicted wrong by the tree is increased and these the variables are then fed to the second decision tree. These individual classifiers/predictors then assemble to give a strong and more precise model. It can work on regression, classification, ranking, and user-defined predict.

Regularization: XG-boost has in-built L1 (Lasso Regression) and L2 (Ridge Regression) regularization which prevents the model from overfitting. That is why,

XG-boost is also called regularized form of GBM (Gradient Boosting Machine). While using Scikit Learn libarary, we pass two hyper-parameters (alpha and lambda) to XG-boost related to regularization. alpha is used for L1 regularization and lambda is used for L2 regularization.

2. Parallel Processing: XG-boost utilizes the power of parallel processing and that is why it is much faster than GBM. It uses multiple CPU cores to execute the model. While using Scikit Learn libarary, nthread hyper-parameter is used for parallel processing. nthread represents number of CPU cores to be used. If you want to use all the available cores, don't mention any value for nthread and the algorithm will detect automatically.

3. Handling Missing Values: XG-boost has an in-built capability to handle missing values. When XG-boost encounters a missing value at a node, it tries both the left and right hand split and learns the way leading to higher loss for each node. It then does the same when working on the testing data.

4. Cross Validation: XG-boost allows user to run a cross-validation at each iteration of the boosting process and thus it is easy to get the exact optimum number of boosting iterations in a single run. This is unlike GBM where we have to run a grid-search and only a limited values can be tested.

5. Effective Tree Pruning: A GBM would stop splitting a node when it encounters a negative loss in the split. Thus it is more of a greedy algorithm. XG-boost on the other hand make splits upto the max_depth specified and then start pruning the tree backwards and remove splits beyond which there is no positive gain.

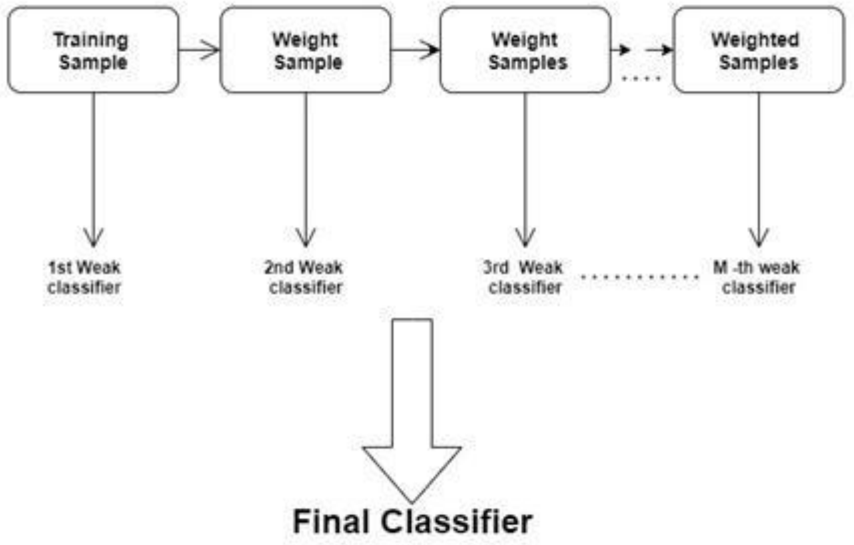


Figure : Xgboost

CHAPTER 5
EXPERIMENTAL ANALYSIS

5.1 SYSTEM CONFIGURATION

5.1.1 Hardware requirements:

Processer	:	Any Update Processer
Ram	:	Min 4GB
Hard Disk	:	Min 100GB

5.1.2 Software requirements:

Operating System	:	Windows family
Technology	:	Python3.7
IDE	:	Jupiter notebook

5.2 SAMPLE CODE

```
#import the libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.model_selection import train_test_split, StratifiedKFold, cross_val_score
from sklearn.pipeline import make_pipeline, Pipeline from sklearn.model_selection
import GridSearchCV

from sklearn.svm import SVC
from sklearn.naive_bayes import MultinomialNB
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.externals import joblib
from sklearn.metrics import make_scorer, f1_score, recall_score, precision_score
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score
from sklearn.metrics import log_loss
import warnings
warnings.simplefilter(action = 'ignore', category= FutureWarning)



from sklearn.ensemble import BaggingClassifier
from sklearn.ensemble import AdaBoostClassifier
import numpy as np
from flask import Flask,request,jsonify, render_template
import pickle
app=Flask(__name__,template_folder='template')
app._static_folder = 'static'
model1=pickle.load(open('model1.pkl','rb'))
model2=pickle.load(open('model2.pkl','rb'))
@app.route('/home')
def homepage():
    return render_template('index.html')
@app.route('/precautions')
def precautions():
    return render_template('precautions.html')
@app.route('/advancedpage')
def advancedpage():
    return render_template('index.html')
@app.route('/quick',methods=['POST'])
def quick():
        def bmi(height,weight):
                bmi=int(weight)/((int(height)/100)**2)
                return bmi
        int_features1 = [float(x) for x in request.form.values()]
```

```
age=int_features1[1]
cigs=int_features1[3]
height=int_features1[8]
weight=int_features1[9]
hrv=int_features1[10]
int_features1.pop(8)
int_features1.pop(9)
bmi=round(bmi(height,weight),2)
int_features1.insert(8,bmi)

if int(int_features1[0])==1.0:
        sex="Male"
else:
        sex="Female"
if int(int_features1[2])==1.0:
        smoking="Yes"
else:
        smoking="No"
if int(int_features1[4])==1.0:
        stroke="Yes"
else:
        stroke="No"

if int(int_features1[5])==1.0:
        hyp="Yes"
else:
        hyp="No"
if int(int_features1[7])==1.0:
        dia="Yes"
else:
        dia="No"
if int(int_features1[6])==1.0:
        bpmeds="Yes"
```

```
    else:
        bpmeds="No"



    final_feature1=[np.array(int_features1)]
    prediction1= model1.predict(final_feature1)
    result=prediction1[0]

    if result==0:
        result="No need to worry"
    else:
        result="You are detected with heart problems. You need to consult a doctor immediately"
    return render_template('quick_report.html',prediction_text1= result,gender=sex,age=age,smoking=smoking,cigs=cigs,stroke=stroke,hyp=hyp,dia=dia,bpmeds=bpmeds,bmi=bmi,hrv=hrv)


@app.route('/quickpage')
def quickpage():
    return render_template('index1.html')


@app.route('/customersupport')
def customersupport():
    return render_template('customercare.html')
@app.route('/Doctorconsult')
def Doctorconsult():
    return render_template('Doctorconsult.html')


@app.route('/')
def home():
    return render_template('Home.html')


@app.route('/advanced',methods=['POST'])
def advanced():
```

```
int_features2 = [int(x) for x in request.form.values()]
final2_feature=[np.array(int_features2)] prediction2=
model2.predict(final2_feature) result=prediction2[0]



age=int_features2[0]
trestbps=int_features2[3]
chol=int_features2[4]
oldspeak=int_features2[7]
thalach=int_features2[7]
ca=int_features2[10]

if int(int_features2[1])==1:
        sex="Male"
else:
        sex="Female"

if int(int_features2[2])==1:
        cp="Typical angina"
  elif int(int_features2[2])==2:
        cp="Atypical angina"
  elif int(int_features2[2])==3:
        cp="Non-angina pain"
else:
        cp="Asymtomatic"



if int(int_features2[5])==1:
        fbs="Yes"
else:
        fbs="No"

if int(int_features2[6])==1:
        restecg="ST-T wave abnormality"
    elif int(int_features2[6])==2:
        restecg="showing probable or definite left ventricular hypertrophy by
Estes"
    else:
        restecg="Normal"

    if int(int_features2[8])==1:
        exang="Yes"
    else:
        exang="No"

    if int(int_features2[9])==1:
        slope="upsloping"
    elif int(int_features2[9])==2:
        slope="flat"
    else:
        slope="downsloping"

    if int(int_features2[11])==3:
        thal="Normal"
    elif int(int_features2[11])==6:
        thal="Fixed defect"
    else:
        thal=" reversable defect"

    if result==0:
        result="No need to worry"
    else:
        result="You are detected with heart problems. You need to consult
a doctor immediately"
    return render_template('advance_report.html',prediction_text2=
result,age=age,sex=sex,cp=cp,trestbps=trestbps,chol=chol,fbs=fbs,restecg=restecg,old
peak=oldspeak,exang=exang,slope=slope,ca=ca,thal=thal)
```

```
if __name__=="__main__":
app.run(debug=True)

#read the csv dataset
data = pd.read_csv("heart.csv", encoding='ANSI')
data.columns
data.head()

#Total number of rows and columns
data.shape

# Plot a line graph for Age V/s heart
disease plt.subplots(figsize =(8,5))
classifiers = ['<=40', '41-50', '51-60','61 and Above']
heart_disease = [13, 53, 64, 35] no_heart_disease =
[6, 23, 65, 44]

l1 = plt.plot(classifiers, heart_disease , color='g', marker='o', linestyle ='dashed',
markerfacecolor='y', markersize=10)
l2 = plt.plot(classifiers, no_heart_disease, color='r',marker='o', linestyle ='dashed',
markerfacecolor='y', markersize=10 )

plt.xlabel('Age')
plt.ylabel('Number of patients')
plt.title('Age V/s Heart disease')
plt.legend((l1[0], l2[0]), ('heart_disease', 'no_heart_disease'))
plt.show()

# Plot a bar graph for Gender V/s target
N = 2
ind = np.arange(N)
width = 0.1
fig, ax = plt.subplots(figsize =(8,4))
```

```
heart_disease = [93, 72]
rects1 = ax.bar(ind, heart_disease, width, color='g')
no_heart_disease = [114, 24]
rects2 = ax.bar(ind+width, no_heart_disease, width, color='y')

ax.set_ylabel('Scores')
ax.set_title('Gender V/s target')
ax.set_xticks(ind)
ax.set_xticklabels(('Male','Female'))
ax.legend((rects1[0], rects2[0]), ('heart disease', 'no heart disease'))

plt.show()

#Pie charts for thal:Thalassemla
# Having heart disease
labels= 'Normal', 'Fixed defect', 'Reversable defect'
sizes=[6, 130, 28]
colors=['red', 'orange', 'green']

plt.pie(sizes, labels=labels, colors=colors, autopct='%.1f%%',
shadow=True, startangle=140)

plt.axis('equal')
plt.title('Thalassemla blood disorder status of patients having heart disease')
plt.show()

# Not having heart disease
labels= 'Normal', 'Fixed defect', 'Reversable defect'
sizes=[12, 36, 89]
colors=['red', 'orange', 'green']
```

```
plt.pie(sizes, labels=labels, colors=colors, autopct='%.1f%%',
shadow=True, startangle=140)


plt.axis('equal')
plt.title('Thalassemla blood disorder status of patients who do not have heart disease')
plt.show()


## Feature selection
#get correlation of each feature in dataset


corrmat = data.corr()
top_corr_features = corrmat.index
plt.figure(figsize=(13,13))


#plot heat map
g=sns.heatmap(data[top_corr_features].corr(),annot=True,cmap="RdYlGn")


data=data.drop(['sex', 'fbs', 'restecg', 'slope', 'chol', 'age', 'trestbps'], axis=1)


target=data['target']
data = data.drop(['target'],axis=1)
data.head()


# We split the data into training and testing set:
x_train, x_test, y_train, y_test = train_test_split(data, target, test_size=0.3,
random_state=10)


## Base Learners
clfs = []
kfolds = StratifiedKFold(n_splits=5, shuffle=True, random_state=1)
np.random.seed(1)


#Support Vector Machine(SVM)
```

```
pipeline_svm = make_pipeline(SVC(probability=True, kernel="linear",
class_weight="balanced"))


grid_svm = GridSearchCV(pipeline_svm,
param_grid = {'svc__C': [0.01, 0.1, 1]},
                    cv = kfolds,
                    verbose=1,
n_jobs=-1)


grid_svm.fit(x_train, y_train)
grid_svm.score(x_test, y_test)
print("\nBest Model: %f using %s" % (grid_svm.best_score_,
grid_svm.best_params_))
print('\n')
print('SVM LogLoss {score}'.format(score=log_loss(y_test,
grid_svm.predict_proba(x_test))))
clfs.append(grid_svm)


# save best model to current working directory
joblib.dump(grid_svm, "heart_disease.pkl")


# load from file and predict using the best configs found in the CV step
model_grid_svm = joblib.load("heart_disease.pkl" )


# get predictions from best model above
y_preds = model_grid_svm.predict(x_test)
print('SVM accuracy score: ',accuracy_score(y_test, y_preds))
print('\n')


import pylab as plt
labels=[0,1]
cmx=confusion_matrix(y_test,y_preds, labels)
print(cmx)
fig = plt.figure()
```

```
ax = fig.add_subplot(111)
cax = ax.matshow(cmx)
plt.title('Confusion matrix of the classifier')
fig.colorbar(cax)
ax.set_xticklabels([''] + labels)
ax.set_yticklabels([''] + labels)
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.show()

print('\n')
print(classification_report(y_test, y_preds))

# Multinomial Naive Bayes(NB)
classifierNB=MultinomialNB()
classifierNB.fit(x_train,y_train)
classifierNB.score(x_test, y_test)

print('MultinomialNBLogLoss {score}'.format(score=log_loss(y_test,
classifierNB.predict_proba(x_test))))
clfs.append(classifierNB)

# save best model to current working directory
joblib.dump(classifierNB, "heart_disease.pkl")

# load from file and predict using the best configs found in the CV step
model_classifierNB = joblib.load("heart_disease.pkl" )

# get predictions from best model above
y_preds = model_classifierNB.predict(x_test)
print('MultinomialNB accuracy score: ',accuracy_score(y_test, y_preds))
print('\n')
```

```
import pylab as plt
labels=[0,1]
cmx=confusion_matrix(y_test,y_preds, labels)
print(cmx)
fig = plt.figure()
ax = fig.add_subplot(111)
cax = ax.matshow(cmx)
plt.title('Confusion matrix of the classifier')
fig.colorbar(cax)
ax.set_xticklabels([''] + labels)
ax.set_yticklabels([''] + labels)
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.show()

print('\n')
print(classification_report(y_test, y_preds))

# Logistic Regression(LR)
classifierLR=LogisticRegression()

classifierLR.fit(x_train,y_train)
classifierLR.score(x_test, y_test)

print('LogisticRegressionLogLoss {score}'.format(score=log_loss(y_test,
classifierLR.predict_proba(x_test))))
clfs.append(classifierLR)

# save best model to current working directory
joblib.dump(classifierLR, "heart_disease.pkl")

# load from file and predict using the best configs found in the CV step
model_classifierLR = joblib.load("heart_disease.pkl" )
```

```
# get predictions from best model above
y_preds = model_classifierLR.predict(x_test)
print('Logistic Regression accuracy score: ',accuracy_score(y_test, y_preds))
print('\n')

import pylab as plt
labels=[0,1]
cmx=confusion_matrix(y_test,y_preds, labels)
print(cmx)
fig = plt.figure()
ax = fig.add_subplot(111)
cax = ax.matshow(cmx)
plt.title('Confusion matrix of the classifier')
fig.colorbar(cax)
ax.set_xticklabels([''] + labels)
ax.set_yticklabels([''] + labels)
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.show()

print('\n')
print(classification_report(y_test, y_preds))

# Decision Tree (DT)
classifierDT=DecisionTreeClassifier(criterion="gini", random_state=50,
max_depth=3, min_samples_leaf=5)
classifierDT.fit(x_train,y_train)
classifierDT.score(x_test, y_test)

print('Decision Tree LogLoss {score}'.format(score=log_loss(y_test,
classifierDT.predict_proba(x_test))))
clfs.append(classifierDT)
```

```
# save best model to current working directory
joblib.dump(classifierDT, "heart_disease.pkl")

# load from file and predict using the best configs found in the CV step
model_classifierDT = joblib.load("heart_disease.pkl" )

# get predictions from best model above
y_preds = model_classifierDT.predict(x_test)
print('Decision Tree accuracy score: ',accuracy_score(y_test, y_preds))
print('\n')

import pylab as plt
labels=[0,1]
cmx=confusion_matrix(y_test,y_preds, labels)
print(cmx)
fig = plt.figure()
ax = fig.add_subplot(111)
cax = ax.matshow(cmx)
plt.title('Confusion matrix of the classifier')
fig.colorbar(cax)
ax.set_xticklabels(["] + labels)
ax.set_yticklabels(["] + labels)
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.show()

print('\n')
print(classification_report(y_test, y_preds))

# Random Forest(RF)
classifierRF=RandomForestClassifier()
classifierRF.fit(x_train,y_train)
classifierRF.score(x_test, y_test)
print('RandomForestLogLoss {score}'.format(score=log_loss(y_test,
classifierRF.predict_proba(x_test))))
clfs.append(classifierRF)

# save best model to current working directory
joblib.dump(classifierRF, "heart_disease.pkl")

# load from file and predict using the best configs found in the CV step
model_classifierRF = joblib.load("heart_disease.pkl" )

# get predictions from best model above
y_preds = model_classifierRF.predict(x_test)
print('Random Forest accuracy score: ',accuracy_score(y_test, y_preds))
print('\n')

import pylab as plt
labels=[0,1]
cmx=confusion_matrix(y_test,y_preds, labels)
print(cmx)
fig = plt.figure()
ax = fig.add_subplot(111)
cax = ax.matshow(cmx)
plt.title('Confusion matrix of the classifier')
fig.colorbar(cax)
ax.set_xticklabels([''] + labels)
ax.set_yticklabels([''] + labels)
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.show()

print('\n')
print(classification_report(y_test, y_preds))
print('\n')
print('Accuracy of svm: {}'.format(grid_svm.score(x_test, y_test)))

print('Accuracy of naive bayes: {}'.format(classifierNB.score(x_test, y_test)))

print('Accuracy of logistic regression: {}'.format(classifierLR.score(x_test, y_test)))

print('Accuracy of decision tree: {}'.format(classifierDT.score(x_test, y_test)))

print('Accuracy of random forest: {}'.format(classifierRF.score(x_test, y_test)))

//#//

#import the libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.model_selection import train_test_split, StratifiedKFold, cross_val_score
from sklearn.pipeline import make_pipeline, Pipeline from sklearn.model_selection
import GridSearchCV

from sklearn.svm import SVC
from sklearn.naive_bayes import MultinomialNB
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier

from sklearn.externals import joblib
from sklearn.metrics import make_scorer, f1_score, recall_score, precision_score
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score
from sklearn.metrics import log_loss

import warnings

warnings.simplefilter(action = 'ignore', category= FutureWarning)

from sklearn.ensemble import BaggingClassifier

from sklearn.ensemble import AdaBoostClassifier

#read the csv dataset

data = pd.read_csv("heart.csv", encoding='ANSI')

data.columns

data.head()


#Total number of rows and columns

data.shape


# Plot a line graph for Age V/s heart

disease plt.subplots(figsize =(8,5))

classifiers = ['<=40', '41-50', '51-60','61 and Above']

heart_disease = [13, 53, 64, 35] no_heart_disease =

[6, 23, 65, 44]


l1 = plt.plot(classifiers, heart_disease , color='g', marker='o', linestyle ='dashed',
markerfacecolor='y', markersize=10)

l2 = plt.plot(classifiers, no_heart_disease, color='r',marker='o', linestyle ='dashed',
markerfacecolor='y', markersize=10 )


plt.xlabel('Age')

plt.ylabel('Number of patients')

plt.title('Age V/s Heart disease')

plt.legend((l1[0], l2[0]), ('heart_disease', 'no_heart_disease'))

plt.show()


# Plot a bar graph for Gender V/s target

N = 2

ind = np.arange(N)
```

```
width = 0.1

fig, ax = plt.subplots(figsize =(8,4))

heart_disease = [93, 72]

rects1 = ax.bar(ind, heart_disease, width, color='g')

no_heart_disease = [114, 24]

rects2 = ax.bar(ind+width, no_heart_disease, width, color='y')


ax.set_ylabel('Scores')

ax.set_title('Gender V/s target')

ax.set_xticks(ind)

ax.set_xticklabels(('Male','Female'))

ax.legend((rects1[0], rects2[0]), ('heart disease', 'no heart disease'))


plt.show()


#Pie charts for thal:Thalassemla

# Having heart disease

labels= 'Normal', 'Fixed defect', 'Reversable defect'

sizes=[6, 130, 28]

colors=['red', 'orange', 'green']


plt.pie(sizes, labels=labels, colors=colors, autopct='%.1f%%',
shadow=True, startangle=140)


plt.axis('equal')

plt.title('Thalassemla blood disorder status of patients having heart disease')

plt.show()


# Not having heart disease

labels= 'Normal', 'Fixed defect', 'Reversable defect'

sizes=[12, 36, 89]

colors=['red', 'orange', 'green']
```

```
plt.pie(sizes, labels=labels, colors=colors, autopct='%.1f%%',
shadow=True, startangle=140)


plt.axis('equal')
plt.title('Thalassemla blood disorder status of patients who do not have heart disease')
plt.show()


## Feature selection
#get correlation of each feature in dataset


corrmat = data.corr()
top_corr_features = corrmat.index
plt.figure(figsize=(13,13))


#plot heat map
g=sns.heatmap(data[top_corr_features].corr(),annot=True,cmap="RdYlGn")


data=data.drop(['sex', 'fbs', 'restecg', 'slope', 'chol', 'age', 'trestbps'], axis=1)


target=data['target']
data = data.drop(['target'],axis=1)
data.head()


# We split the data into training and testing set:
x_train, x_test, y_train, y_test = train_test_split(data, target, test_size=0.3,
random_state=10)


## Base Learners
clfs = []
kfolds = StratifiedKFold(n_splits=5, shuffle=True, random_state=1)
np.random.seed(1)


#Support Vector Machine(SVM)
```

```
pipeline_svm = make_pipeline(SVC(probability=True, kernel="linear",
class_weight="balanced"))


grid_svm = GridSearchCV(pipeline_svm,

param_grid = {'svc__C': [0.01, 0.1, 1]},

               cv = kfolds,

               verbose=1,

n_jobs=-1)


grid_svm.fit(x_train, y_train)

grid_svm.score(x_test, y_test)

print("\nBest Model: %f using %s" % (grid_svm.best_score_,
grid_svm.best_params_))

print('\n')

print('SVM LogLoss {score}'.format(score=log_loss(y_test,
grid_svm.predict_proba(x_test))))

clfs.append(grid_svm)


# save best model to current working directory

joblib.dump(grid_svm, "heart_disease.pkl")


# load from file and predict using the best configs found in the CV step

model_grid_svm = joblib.load("heart_disease.pkl" )


# get predictions from best model above

y_preds = model_grid_svm.predict(x_test)

print('SVM accuracy score: ',accuracy_score(y_test, y_preds))

print('\n')


import pylab as plt

labels=[0,1]

cmx=confusion_matrix(y_test,y_preds, labels)

print(cmx)

fig = plt.figure()
```

```
ax = fig.add_subplot(111)
cax = ax.matshow(cmx)
plt.title('Confusion matrix of the classifier')
fig.colorbar(cax)
ax.set_xticklabels(["] + labels)
ax.set_yticklabels(["] + labels)
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.show()

print('\n')
print(classification_report(y_test, y_preds))

# Multinomial Naive Bayes(NB)
classifierNB=MultinomialNB()
classifierNB.fit(x_train,y_train)
classifierNB.score(x_test, y_test)

print('MultinomialNBLogLoss {score}'.format(score=log_loss(y_test,
classifierNB.predict_proba(x_test))))
clfs.append(classifierNB)

# save best model to current working directory
joblib.dump(classifierNB, "heart_disease.pkl")

# load from file and predict using the best configs found in the CV step
model_classifierNB = joblib.load("heart_disease.pkl" )

# get predictions from best model above
y_preds = model_classifierNB.predict(x_test)
print('MultinomialNB accuracy score: ',accuracy_score(y_test, y_preds))
print('\n')
```

```
import pylab as plt
labels=[0,1]
cmx=confusion_matrix(y_test,y_preds, labels)
print(cmx)
fig = plt.figure()
ax = fig.add_subplot(111)
cax = ax.matshow(cmx)
plt.title('Confusion matrix of the classifier')
fig.colorbar(cax)
ax.set_xticklabels([''] + labels)
ax.set_yticklabels([''] + labels)
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.show()

print('\n')
print(classification_report(y_test, y_preds))

# Logistic Regression(LR)
classifierLR=LogisticRegression()

classifierLR.fit(x_train,y_train)
classifierLR.score(x_test, y_test)

print('LogisticRegressionLogLoss {score}'.format(score=log_loss(y_test,
classifierLR.predict_proba(x_test))))
clfs.append(classifierLR)

# save best model to current working directory
joblib.dump(classifierLR, "heart_disease.pkl")

# load from file and predict using the best configs found in the CV step
model_classifierLR = joblib.load("heart_disease.pkl" )
```

```
# get predictions from best model above
y_preds = model_classifierLR.predict(x_test)
print('Logistic Regression accuracy score: ',accuracy_score(y_test, y_preds))
print('\n')

import pylab as plt
labels=[0,1]
cmx=confusion_matrix(y_test,y_preds, labels)
print(cmx)
fig = plt.figure()
ax = fig.add_subplot(111)
cax = ax.matshow(cmx)
plt.title('Confusion matrix of the classifier')
fig.colorbar(cax)
ax.set_xticklabels(["] + labels)
ax.set_yticklabels(["] + labels)
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.show()

print('\n')
print(classification_report(y_test, y_preds))

# Decision Tree (DT)
classifierDT=DecisionTreeClassifier(criterion="gini", random_state=50,
max_depth=3, min_samples_leaf=5)
classifierDT.fit(x_train,y_train)
classifierDT.score(x_test, y_test)
print('Decision Tree LogLoss {score}'.format(score=log_loss(y_test,
classifierDT.predict_proba(x_test))))
clfs.append(classifierDT)
```

```
# save best model to current working directory
joblib.dump(classifierDT, "heart_disease.pkl")

# load from file and predict using the best configs found in the CV step
model_classifierDT = joblib.load("heart_disease.pkl" )

# get predictions from best model above
y_preds = model_classifierDT.predict(x_test)
print('Decision Tree accuracy score: ',accuracy_score(y_test, y_preds))
print('\n')

import pylab as plt
labels=[0,1]
cmx=confusion_matrix(y_test,y_preds, labels)
print(cmx)
fig = plt.figure()
ax = fig.add_subplot(111)
cax = ax.matshow(cmx)
plt.title('Confusion matrix of the classifier')
fig.colorbar(cax)
ax.set_xticklabels([''] + labels)
ax.set_yticklabels([''] + labels)
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.show()

print('\n')
print(classification_report(y_test, y_preds))

# Random Forest(RF)
classifierRF=RandomForestClassifier()
classifierRF.fit(x_train,y_train)
classifierRF.score(x_test, y_test)
```

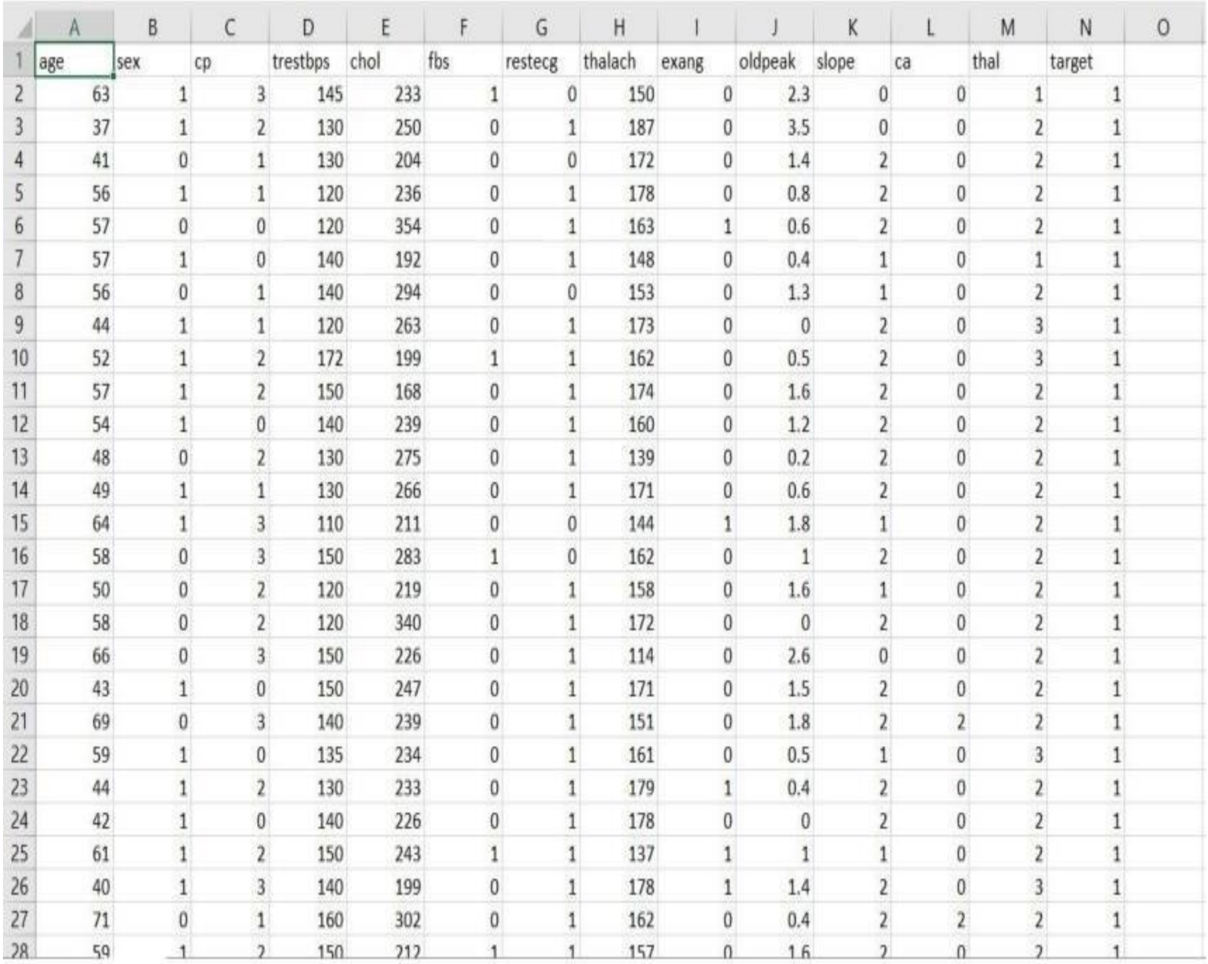
```
print('RandomForestLogLoss {score}'.format(score=log_loss(y_test, classifierRF.predict_proba(x_test))))
clfs.append(classifierRF)

# save best model to current working directory
joblib.dump(classifierRF, "heart_disease.pkl")

# load from file and predict using the best configs found in the CV step
model_classifierRF = joblib.load("heart_disease.pkl" )

# get predictions from best model above
y_preds = model_classifierRF.predict(x_test)
print('Random Forest accuracy score: ',accuracy_score(y_test, y_preds))
print('\n')
import pylab as plt
labels=[0,1]
cmx=confusion_matrix(y_test,y_preds, labels)
print(cmx)
fig = plt.figure()
ax = fig.add_subplot(111)
cax = ax.matshow(cmx)
plt.title('Confusion matrix of the classifier')
fig.colorbar(cax)
ax.set_xticklabels([''] + labels)
ax.set_yticklabels([''] + labels)
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.show()
print('\n')
print(classification_report(y_test, y_preds))
print('\n')
print('Accuracy of svm: {}'.format(grid_svm.score(x_test, y_test)))
print('Accuracy of naive bayes: {}'.format(classifierNB.score(x_test, y_test)))
print('Accuracy of logistic regression: {}'.format(classifierLR.score(x_test, y_test)))
print('Accuracy of decision tree: {}'.format(classifierDT.score(x_test, y_test)))
print('Accuracy of random forest: {}'.format(classifierRF.score(x_test, y_test)))
//#//
import numpy as np
from flask import Flask,request,jsonify, render_template
import pickle
app=Flask(__name__,template_folder='template')
app._static_folder = 'static'
model1=pickle.load(open('model1.pkl','rb'))
model2=pickle.load(open('model2.pkl','rb'))
@app.route('/home')
def homepage():
    return render_template('index.html')
@app.route('/precautions')
def precautions():
    return render_template('precautions.html')
@app.route('/advancedpage')
def advancedpage():
    return render_template('index.html')
@app.route('/quick',methods=['POST'])
def quick():
        def bmi(height,weight):
                bmi=int(weight)/((int(height)/100)**2)
                return bmi
        int_features1 = [float(x) for x in request.form.values()]

        age=int_features1[1]
        cigs=int_features1[3]
        height=int_features1[8]
        weight=int_features1[9]
        hrv=int_features1[10]
        int_features1.pop(8)
```

```
int_features1.pop(9)
bmi=round(bmi(height,weight),2)
int_features1.insert(8,bmi)

if int(int_features1[0])==1.0:
        sex="Male"
else:
        sex="Female"
if int(int_features1[2])==1.0:
        smoking="Yes"
else:
        smoking="No"
if int(int_features1[4])==1.0:
        stroke="Yes"
else:
        stroke="No"

if int(int_features1[5])==1.0:
        hyp="Yes"
else:
        hyp="No"
if int(int_features1[7])==1.0:
        dia="Yes"
else:
        dia="No"
if int(int_features1[6])==1.0:
        bpmeds="Yes"
else:
        bpmeds="No"
final_feature1=[np.array(int_features1)]
prediction1= model1.predict(final_feature1)
result=prediction1[0]
```

```
            if result==0:
                        result="No need to worry"
            else:
                        result="You are detected with heart problems. You need to consult
a doctor immediately"
            return render_template('quick_report.html',prediction_text1=
result,gender=sex,age=age,smoking=smoking,cigs=cigs,stroke=stroke,hyp=hyp,dia=di
a,bpmeds=bpmeds,bmi=bmi,hrv=hrv)


@app.route('/quickpage')
def quickpage():
    return render_template('index1.html')


@app.route('/customersupport')
def customersupport():
    return render_template('customercare.html')
@app.route('/Doctorconsult')
def Doctorconsult():
    return render_template('Doctorconsult.html')
@app.route('/')
def home():
    return render_template('Home.html')


@app.route('/advanced',methods=['POST'])
def advanced():
            int_features2 = [int(x) for x in request.form.values()]
            final2_feature=[np.array(int_features2)] prediction2=
            model2.predict(final2_feature) result=prediction2[0]



            age=int_features2[0]
            trestbps=int_features2[3]
            chol=int_features2[4]
            oldspeak=int_features2[7]
    thalach=int_features2[7]
    ca=int_features2[10]

    if int(int_features2[1])==1:
        sex="Male"
    else:
        sex="Female"

    if int(int_features2[2])==1:
        cp="Typical angina"
    elif int(int_features2[2])==2:
        cp="Atypical angina"
    elif int(int_features2[2])==3:
        cp="Non-angina pain"
    else:
        cp="Asymtomatic"



    if int(int_features2[5])==1:
        fbs="Yes"
    else:
        fbs="No"

    if int(int_features2[6])==1:
        restecg="ST-T wave abnormality"
    elif int(int_features2[6])==2:
        restecg="showing probable or definite left ventricular hypertrophy by Estes"
    else:
        restecg="Normal"

    if int(int_features2[8])==1:
        exang="Yes"
    else:
        exang="No"

    if int(int_features2[9])==1:
        slope="upsloping"
    elif int(int_features2[9])==2:
        slope="flat"
    else:
        slope="downsloping"

    if int(int_features2[11])==3:
        thal="Normal"
    elif int(int_features2[11])==6:
        thal="Fixed defect"
    else:
        thal=" reversable defect"

    if result==0:
        result="No need to worry"
    else:
        result="You are detected with heart problems. You need to consult a doctor immediately"
    return render_template('advance_report.html',prediction_text2= result,age=age,sex=sex,cp=cp,trestbps=trestbps,chol=chol,fbs=fbs,restecg=restecg,oldpeak=oldspeak,exang=exang,slope=slope,ca=ca,thal=thal)
if __name__=="__main__":
app.run(debug=True)
```

5.3 DATASET DETAILS

- Of the 76 attributes available in the dataset,14 attributes are considered for the prediction of the output.
- Heart Disease UCI : https://archive.ics.uci.edu/ml/datasets/Heart+Disease



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target	
2	63	1	3	145	233	1	0	150	0	2.3	0	0	1	1	
3	37	1	2	130	250	0	1	187	0	3.5	0	0	2	1	
4	41	0	1	130	204	0	0	172	0	1.4	2	0	2	1	
5	56	1	1	120	236	0	1	178	0	0.8	2	0	2	1	
6	57	0	0	120	354	0	1	163	1	0.6	2	0	2	1	
7	57	1	0	140	192	0	1	148	0	0.4	1	0	1	1	
8	56	0	1	140	294	0	0	153	0	1.3	1	0	2	1	
9	44	1	1	120	263	0	1	173	0	0	2	0	3	1	
10	52	1	2	172	199	1	1	162	0	0.5	2	0	3	1	
11	57	1	2	150	168	0	1	174	0	1.6	2	0	2	1	
12	54	1	0	140	239	0	1	160	0	1.2	2	0	2	1	
13	48	0	2	130	275	0	1	139	0	0.2	2	0	2	1	
14	49	1	1	130	266	0	1	171	0	0.6	2	0	2	1	
15	64	1	3	110	211	0	0	144	1	1.8	1	0	2	1	
16	58	0	3	150	283	1	0	162	0	1	2	0	2	1	
17	50	0	2	120	219	0	1	158	0	1.6	1	0	2	1	
18	58	0	2	120	340	0	1	172	0	0	2	0	2	1	
19	66	0	3	150	226	0	1	114	0	2.6	0	0	2	1	
20	43	1	0	150	247	0	1	171	0	1.5	2	0	2	1	
21	69	0	3	140	239	0	1	151	0	1.8	2	2	2	1	
22	59	1	0	135	234	0	1	161	0	0.5	1	0	3	1	
23	44	1	2	130	233	0	1	179	1	0.4	2	0	2	1	
24	42	1	0	140	226	0	1	178	0	0	2	0	2	1	
25	61	1	2	150	243	1	1	137	1	1	1	0	2	1	
26	40	1	3	140	199	0	1	178	1	1.4	2	0	3	1	
27	71	0	1	160	302	0	1	162	0	0.4	2	2	2	1	
28	59	1	2	150	212	1	1	157	0	1.6	2	0	2	1	

Figure: Dataset Attributes

Input dataset attributes

- Gender (value 1: Male; value 0 : Female)

- Chest Pain Type (value 1: typical type 1 angina, value 2: typical type angina, value 3: non-angina pain; value 4: asymptomatic)

- Fasting Blood Sugar (value 1: > 120 mg/dl; value 0:< 120 mg/dl)

- Exang – exercise induced angina (value 1: yes; value 0: no)

- CA – number of major vessels colored by fluoroscopy (value 0 – 3)

- Thal (value 3: normal; value 6: fixed defect; value 7:reversible defect)

- Trest Blood Pressure (mm Hg on admission to the hospital)

- Serum Cholesterol (mg/dl)

- Thalach – maximum heart rate achieved

- Age in Year

- Height in cms

- Weight in Kgs.

- Cholestrol

- Restecg

S. No.	Attribute	Description	Type
1	Age	Patient’s age (29 to 77)	Numerical
2	Sex	Gender of patient(male-0 female-1)	Nominal
3	Cp	Chest pain type	Nominal
4	Trestbps	Resting blood pressure(in mm Hg on admission to hospital ,values from 94 to 200)	Numerical
5	Chol	Serum cholesterol in mg/dl, values from 126 to 564)	Numerical
6	Fbs	Fasting blood sugar>120 mg/dl, true-1 false-0)	Nominal
7	Resting	Resting electrocardiographics result (0 to 1)	Nominal
8	Thali	Maximum heart rate achieved(71 to 202)	Numerical
9	Exang	Exercise included agina(1-yes 0-no)	Nominal
10	Oldpeak	ST depression introduced by exercise relative to rest (0 to .2)	Numerical
11	Slope	The slop of the peak exercise ST segment (0 to 1)	Nominal
12	Ca	Number of major vessels (0-3)	Numerical
13	Thal	3-normal	Nominal
14	Targets	1 or 0	Nominal

TABLE2: Attributes of the dataset

5.4 PERFORMANCE ANALYSIS

In this project, various machine learning algorithms like SVM, Naive Bayes, Decision Tree, Random Forest, Logistic Regression, Adaboost, XG-boost are used to predict heart disease. Heart Disease UCI dataset, has a total of 76 attributes, out of those only 14 attributes are considered for the prediction of heart disease. Various attributes of the patient like gender, chest pain type, fasting blood pressure, serum cholesterol, exang, etc are considered for this project. The accuracy for individual algorithms has to measure and whichever algorithm is giving the best accuracy,that is considered for the heart disease prediction. For evaluating the experiment, various evaluation metrics like accuracy, confusion matrix, precision, recall, and f1-score are considered.

Accuracy- Accuracy is the ratio of the number of correct predictions to the total number of inputs in the dataset. It is expressed as:

Accuracy = (TP + TN) /(TP+FP+FN+TN)

Confusion Matrix- It gives us a matrix as output and gives the total performance of the system.

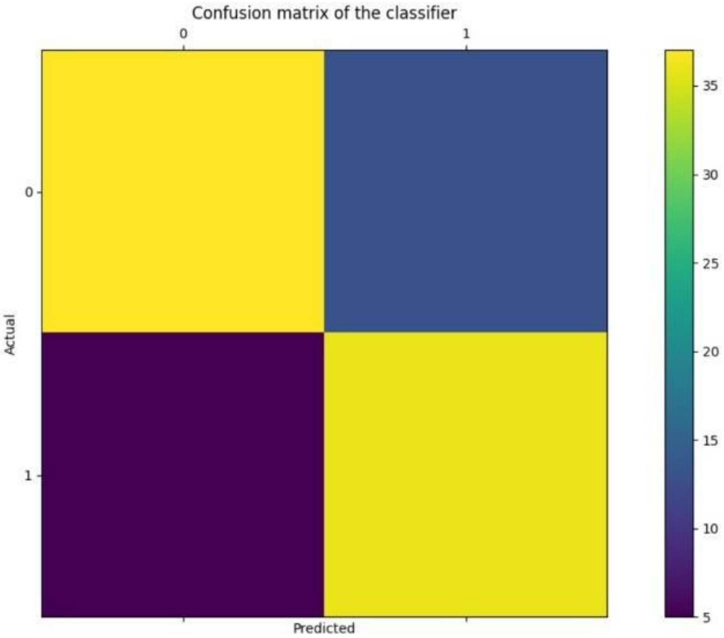


Figure: Confusion Matrix

Where

TP: True positive

FP: False Positive

FN: False Negative

TN: True Negative

Correlation Matrix: The correlation matrix in machine learning is used for feature selection. It represents dependency between various attributes.

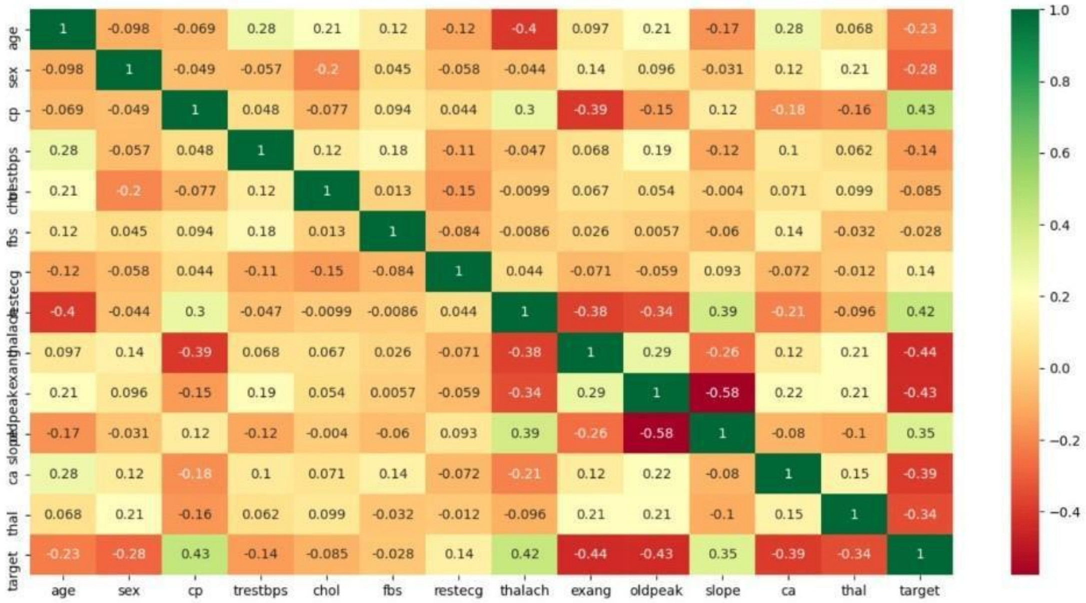


Fig: Correlation matrix

Precision- It is the ratio of correct positive results to the total number of positive results predicted by the system.

It is expressed as:

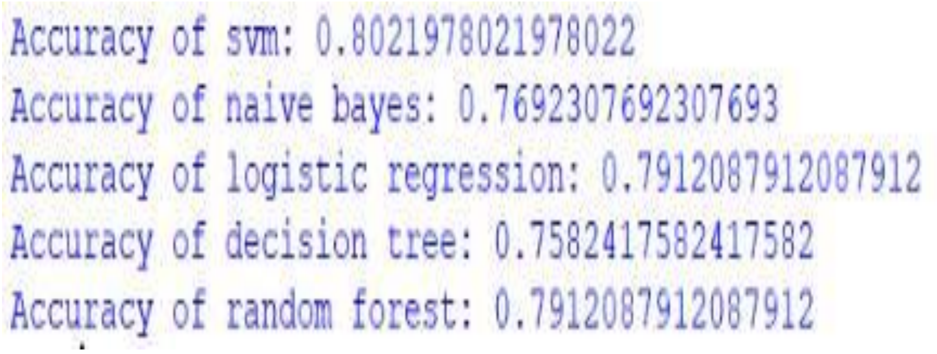
Recall-It is the ratio of correct positive results to the total number of positive results predicted by the system.

It is expressed as:

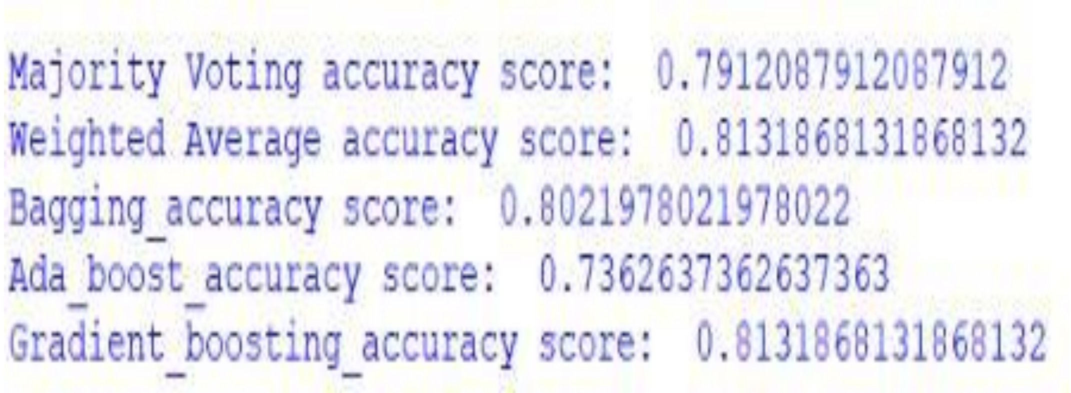
F1 Score-It is the harmonic mean of Precision and Recall. It measures the test accuracy. The range of this metric is 0 to 1.

5.5 PERFORMANCE MEASURES

- The highest accuracy is given by XG-boost.



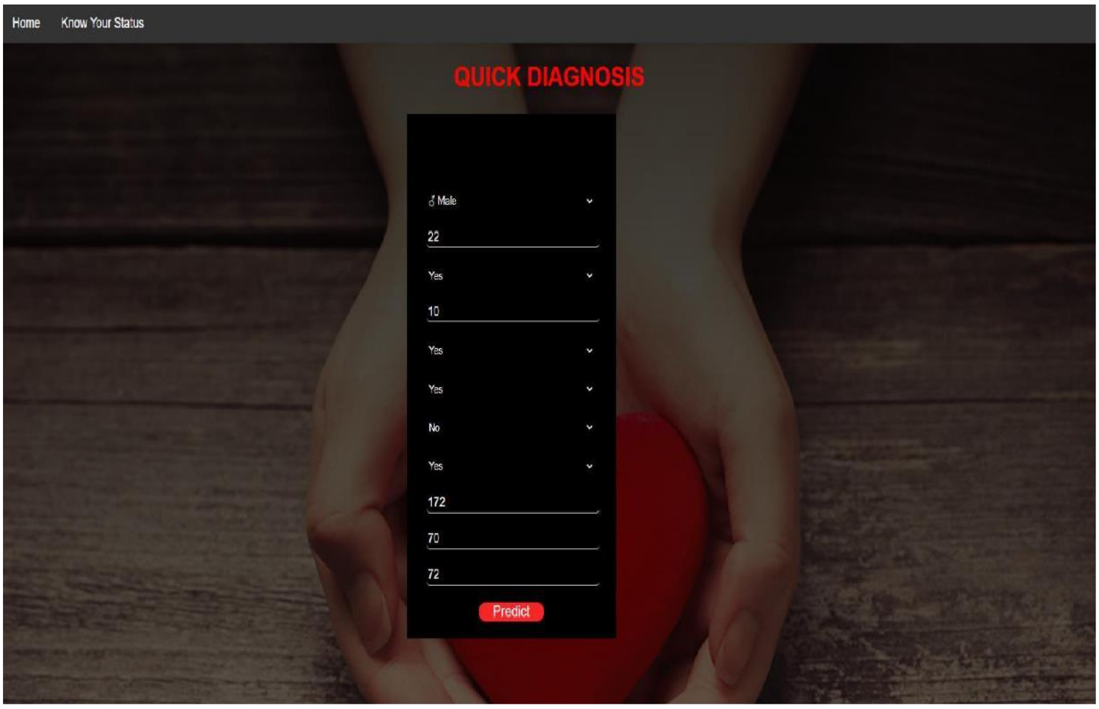
```
Accuracy of svm: 0.8021978021978022
Accuracy of naive bayes: 0.7692307692307693
Accuracy of logistic regression: 0.7912087912087912
Accuracy of decision tree: 0.7582417582417582
Accuracy of random forest: 0.7912087912087912
```



```
Majority Voting accuracy score:  0.7912087912087912
Weighted Average accuracy score:  0.8131868131868132
Bagging_accuracy score:  0.8021978021978022
Ada_boost_accuracy score:  0.7362637362637363
Gradient_boosting_accuracy score:  0.8131868131868132
```

5.6 INPUT AND OUTPUT

5.6.1 Input



♂ Male

22

Yes

10

Yes

Yes

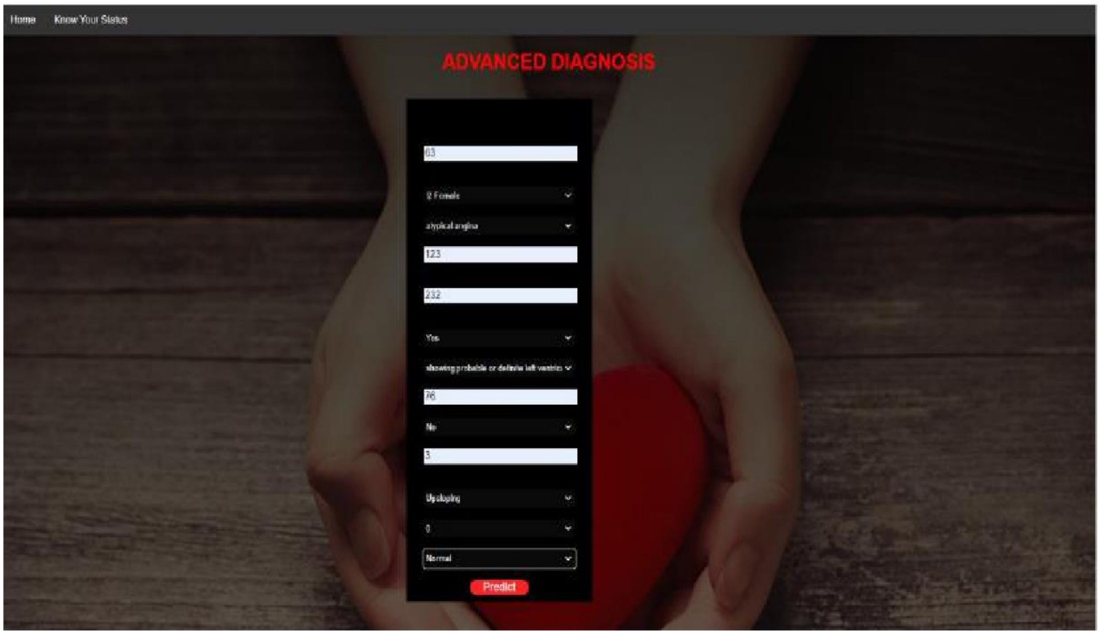
No

Yes

172

70

72



63

♀ Female

atypical angina

123

232

Yes

showing probable or definite left ventric

76

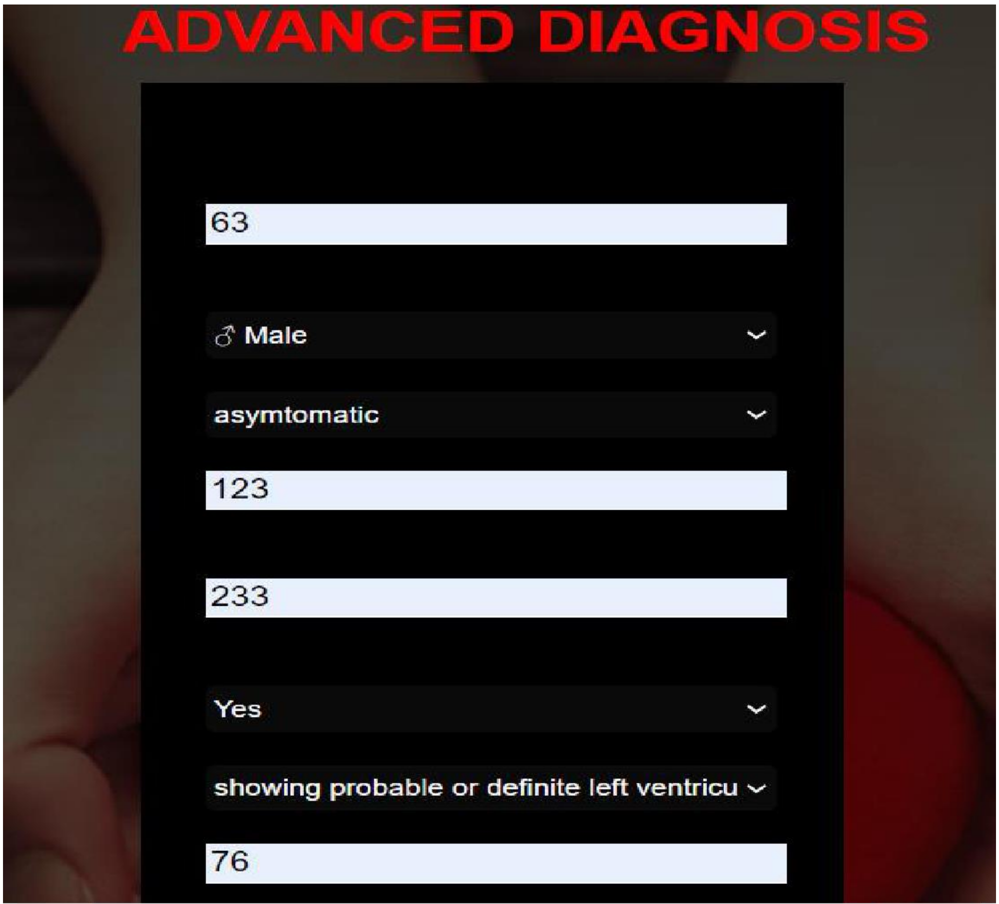
No

3

Upsloping

0

Normal



ADVANCED DIAGNOSIS

63

♂ Male

asymtomatic

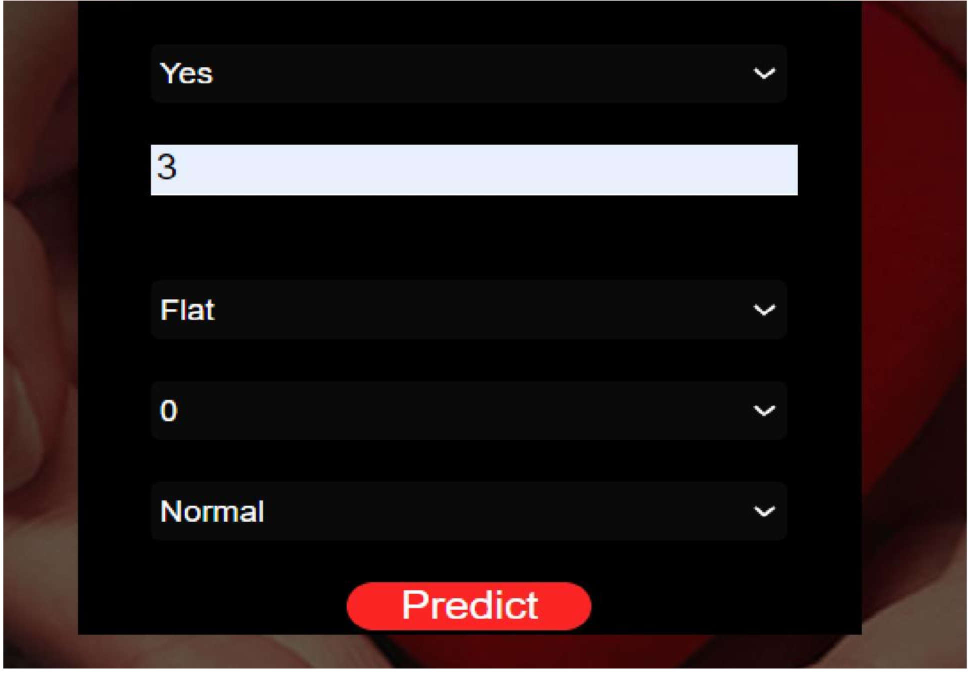
123

233

Yes

showing probable or definite left ventricu

76



Yes

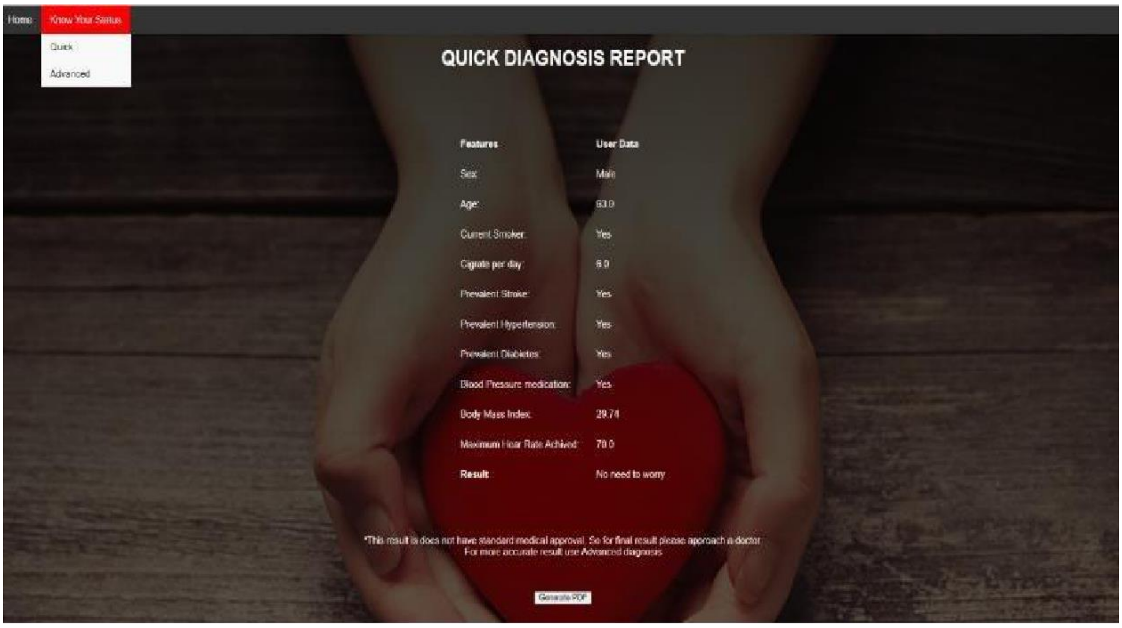
3

Flat

0

Normal

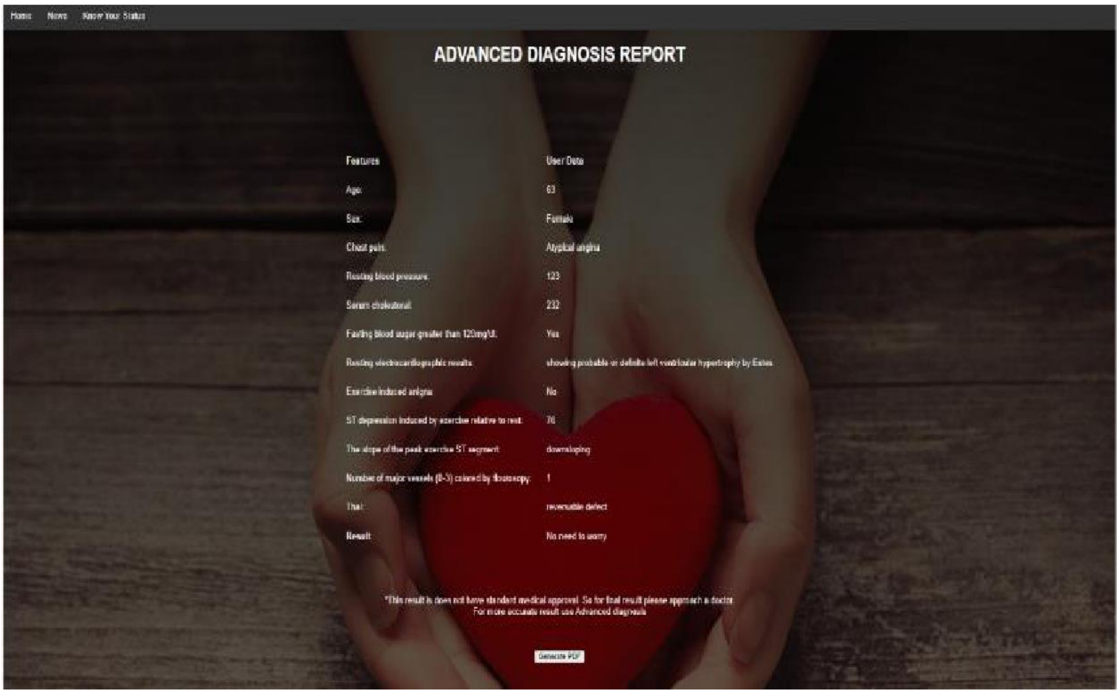
5.6.2 Output



QUICK DIAGNOSIS REPORT

Features	User Data
Sex:	Male
Age:	63.0
Current Smoker:	Yes
Cigrate per day:	8.0
Prevalent Stroke:	Yes
Prevalent Hypertension:	Yes
Prevalent Diabietes:	Yes
Blood Pressure medication:	Yes
Body Mass Index:	29.74
Maximum Hear Rate Achived:	70.0
Result:	No need to worry

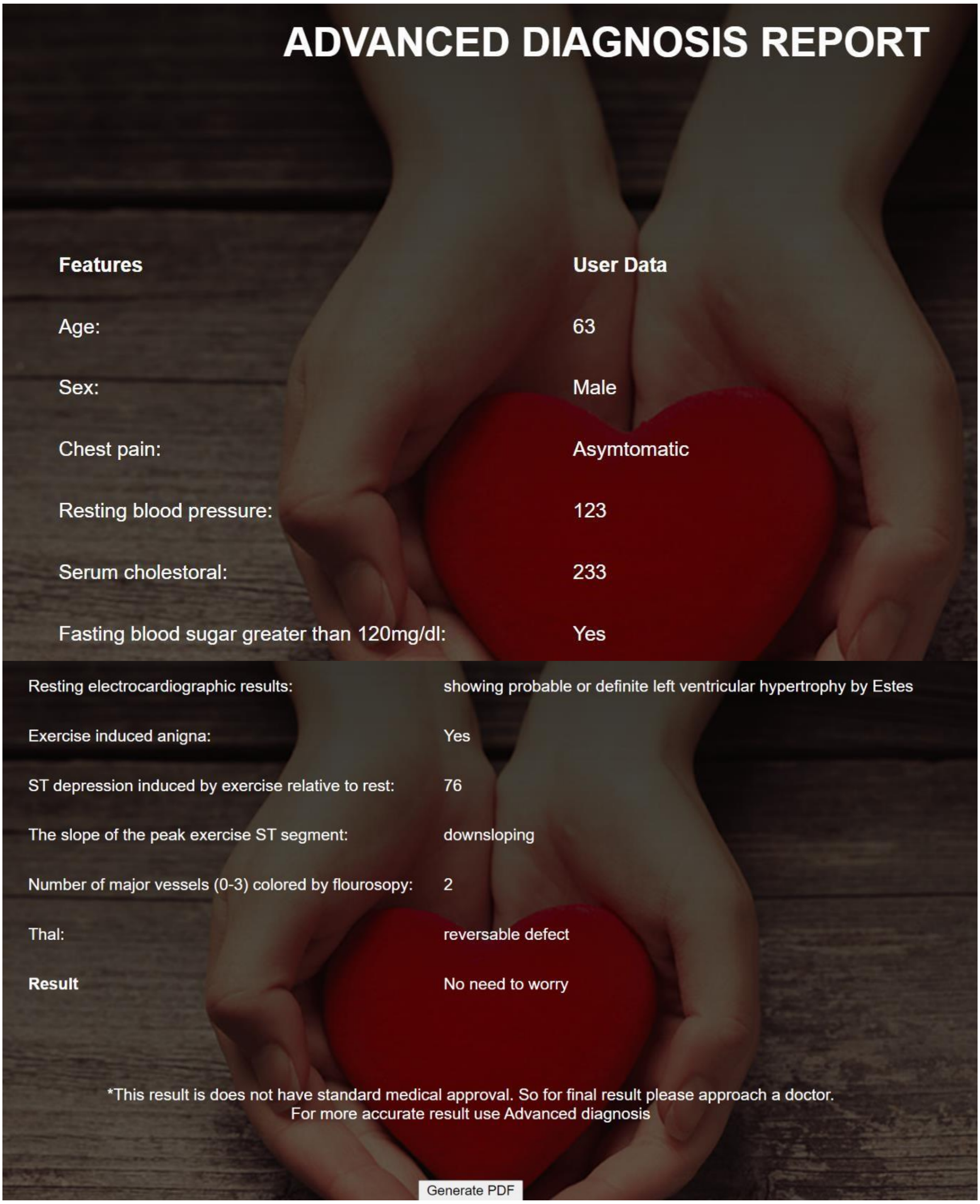
*This result is does not have standard medical approval. So for final result please approach a doctor.
For more accurate result use Advanced diagnosis



ADVANCED DIAGNOSIS REPORT

Features	User Data
Age:	63
Sex:	Female
Chest pain:	Atypical angina
Resting blood pressure:	123
Serum cholestoral:	232
Fasting blood sugar greater than 120mg/dl:	Yes
Resting electrocardiographic results:	showing probable or definite left ventricular hypertrophy by Estes
Exercise induced anigna:	No
ST depression induced by exercise relative to rest:	76
The slope of the peak exercise ST segment:	downsloping
Number of major vessels (0-3) colored by flourosopy:	1
Thal:	reversable defect
Result:	No need to worry

*This result is does not have standard medical approval. So for final result please approach a doctor.
For more accurate result use Advanced diagnosis



ADVANCED DIAGNOSIS REPORT

Features	User Data
Age:	63
Sex:	Male
Chest pain:	Asymtomatic
Resting blood pressure:	123
Serum cholestoral:	233
Fasting blood sugar greater than 120mg/dl:	Yes
Resting electrocardiographic results:	showing probable or definite left ventricular hypertrophy by Estes
Exercise induced anigna:	Yes
ST depression induced by exercise relative to rest:	76
The slope of the peak exercise ST segment:	downsloping
Number of major vessels (0-3) colored by flourosopy:	2
Thal:	reversable defect
Result	No need to worry

*This result is does not have standard medical approval. So for final result please approach a doctor.
For more accurate result use Advanced diagnosis

5.7 RESULT

After performing the machine learning approach for training and testing we find that accuracy of the XG-boost is better compared to other algorithms. Accuracy is calculated with the support of the confusion matrix of each algorithm, here the number count of TP, TN, FP, FN is given and using the equation of accuracy, value has been calculated and it is concluded that extreme gradient boosting is best with 81% accuracy and the comparison is shown below.

TABLE: Accuracy comparison of algorithms Algorithm Accuracy

Algorithm	Accuracy
XG-boost	81.3%
SVM	80.2%
Logistic Regression	79.1%
Random Forest	79.1%
Naive Bayes	76.9%
Decision Tree	75.8%
Adaboost	73.6%

TABLE 2: Accuracy Table

CHAPTER 6

CONCLUSION AND FUTURE WORK

Heart diseases are a major killer in India and throughout the world, 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 diseases is on a raise each year. This prompts for its early diagnosis and treatment. The utilization of suitable technology support in this regard can prove to be highly beneficial to the medical fraternity and patients. In this paper, the seven different machine learning algorithms used to measure the performance are SVM, Decision Tree, Random Forest, Naïve Bayes, Logistic Regression, Adaptive Boosting, and Extreme Gradient Boosting applied on the dataset.

The expected attributes leading to heart disease in patients are available in the dataset which contains 76 features and 14 important features that are useful to evaluate the system are selected among them. If all the features taken into the consideration then the efficiency of the system the author gets is less. To increase efficiency, attribute selection is done. In this n features have to be selected for evaluating the model which gives more accuracy. The correlation of some features in the dataset is almost equal and so they are removed. If all the attributes present in the dataset are taken into account then the efficiency decreases considerably.

All the seven machine learning methods accuracies are compared based on which one prediction model is generated. Hence, the aim is to use various evaluation metrics like confusion matrix, accuracy, precision, recall, and f1-score which predicts the disease efficiently. Comparing all seven the extreme gradient boosting classifier gives the highest accuracy of 81%.

APPENDIX

Python

Python is an interpreted, high-level, general purpose programming language created by Guido Van Rossum and first released in 1991, Python's design philosophy emphasizes code Readability with its notable use of significant White space. Its language constructs and object oriented approach aim to help programmers write clear, logical code for small and large-scale projects. Python is dynamically typed and garbage collected. It supports multiple programming paradigms, including procedural, object-oriented, and functional programming.

Sklearn

Scikit-learn (Sklearn) is the most useful and robust library for machine learning in Python. It provides a selection of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction via a consistent interface in Python. This library, which is largely written in Python, is built upon NumPy, SciPy and Matplotlib.

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. The ancestor of NumPy, Numeric, was originally created by Jim with contributions from several other developers. In 2005, Travis created NumPy by incorporating features of the competing Numarray into Numeric, with extensive modifications. NumPy is open source software and has many contributors.

Librosa

Librosa is a Python package for music and audio analysis. Librosa is basically used when we work with audio data like in music generation(using LSTMs), Automatic Speech Recognition.

It provides the building blocks necessary to create the music information retrieval systems. Librosa helps to visualize the audio signals and also do the feature extractions in it using different signal processing techniques.

Matplotlib

Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy. It provides an object-oriented API for embedding plots into applications using general-purpose GUI toolkits like Tkinter, wxPython, Qt, or GTK. There is also a procedural "pylab" interface based on a statemachine (like OpenGL), designed to closely resemble that of MATLAB, though its use is discouraged.

Seaborn

Seaborn is a Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics. Seaborn is a library in Python predominantly used for making statistical graphics. Seaborn is a data visualization library built on top of matplotlib and closely integrated with pandas data structures in Python. Visualization is the central part of Seaborn which helps in exploration and understanding of data.

SciPy

SciPy contains modules for optimization, linearalgebra, integration, interpolation, special functions, FFT, signal and imageprocessing, ODE solvers and other tasks common in science and engineering. SciPy is also a family of conferences for users and developers of these tools: SciPy (in the United States), EuroSciPy (in Europe) and SciPy.in (in India). Enthought originated the SciPy conference in the United States and continues to sponsor many of the international conferences as well as host the SciPy website. SciPy is a scientific computation library that uses NumPy underneath. It provides more utility functions for optimization, stats and signal processing.

REFERENCES

[1] Soni J, Ansari U, Sharma D & Soni S (2011). Predictive data mining for medical diagnosis: an overview of heart disease prediction. International Journal of Computer Applications, 17(8), 43-8

[2] Dangare C S & Apte S S (2012). Improved study of heart disease prediction system using data mining classification techniques. International Journal of Computer Applications, 47(10), 44-8.

[3] Ordonez C (2006). Association rule discovery with the train and test approach for heart disease prediction. IEEE Transactions on Information Technology in Biomedicine, 10(2), 334-43.

[4] Shinde R, Arjun S, Patil P & Waghmare J (2015). An intelligent heart disease prediction system using k-means clustering and Naïve Bayes algorithm. International Journal of Computer Science and Information Technologies, 6(1), 637-9.

[5] Bashir S, Qamar U & Javed M Y (2014, November). An ensemble-based decision support framework for intelligent heart disease diagnosis. In International Conference on Information Society (i-Society 2014) (pp. 259-64). IEEE. ICCRDA 2020 IOP Conf. Series: Materials Science and Engineering 1022 (2021) 012072 IOP Publishing doi:10.1088/1757-899X/1022/1/012072 9

[6] Jee S H, Jang Y, Oh D J, Oh B H, Lee S H, Park S W & Yun Y D (2014). A coronary heart disease prediction model: the Korean Heart Study. BMJ open, 4(5), e005025.

[7] Ganna A, Magnusson P K, Pedersen N L, de Faire U, Reilly M, Ärnlöv J & Ingelsson E (2013). Multilocus genetic risk scores for coronary heart disease prediction. Arteriosclerosis, thrombosis, and vascular biology, 33(9), 2267-72.

[8] Jabbar M A, Deekshatulu B L & Chandra P (2013, March). Heart disease prediction using lazy associative classification. In 2013 International Mutli-Conference on Automation, Computing,Communication, Control and Compressed Sensing (iMac4s) (pp. 40- 6). IEEE.

[9] Brown N, Young T, Gray D, Skene A M & Hampton J R (1997). Inpatient deaths from acute myocardial infarction, 1982-92: analysis of data in the Nottingham heart attack register. BMJ, 315(7101), 159-64.

[10] Folsom A R, Prineas R J, Kaye S A & Soler J T (1989). Body fat distribution and self-reported prevalence of hypertension, heart attack, and other heart disease in older women. International journal of epidemiologyy, 18(2), 361-7.

[11] Chen A H, Huang S Y, Hong P S, Cheng C H & Lin E J (2011, September). HDPS: Heart disease prediction system. In 2011 Computing in Cardiology (pp. 557-60). IEEE.

[12] Parthiban, Latha and R Subramanian. "Intelligent heart disease prediction system using CANFIS and genetic algorithm." International Journal of Biological, Biomedical and Medical Sciences 3.3 (2008).

[13] Wolgast G, Ehrenborg C, Israelsson A, Helander J, Johansson E & Manefjord H (2016). Wireless body area network for heart attack detection [Education Corner]. IEEE antennas and propagation magazine, 58(5), 84-92.

[14] Patel S & Chauhan Y (2014). Heart attack detection and medical attention using motion sensing device -kinect. International Journal of Scientific and Research Publications, 4(1), 1-4.

[15] Piller L B, Davis B R, Cutler J A, Cushman W C, Wright J T, Williamson J D & Haywood L J (2002). Validation of heart failure events in the Antihypertensive and Lipid Lowering Treatment to Prevent Heart Attack Trial (ALLHAT) participants assigned to doxazosin and chlorthalidone. Current controlled trials in cardiovascular medicine

[16] Raihan M, Mondal S, More A, Sagor M O F, Sikder G, Majumder M A & Ghosh K (2016, December). Smartphone based ischemic heart disease (heart attack) risk prediction using clinical data and data mining approaches, a prototype design. In 2016 19th International Conference on Computer and Information Technology (ICCIT) (pp. 299-303). IEEE.

[17] A. Aldallal and A. A. A. Al-Moosa, "Using Data Mining Techniques to Predict Diabetes and Heart Diseases", 2018 4th International Conference on Frontiers of Signal Processing (ICFSP), pp. 150-154, 2018, September.

[18] Takci H (2018). Improvement of heart attack prediction by the feature selection methods. Turkish Journal of Electrical Engineering & Computer Sciences, 26(1), 1-10.

[19] Ankita Dewan and Meghna Sharma, "Prediction of heart disease using a hybrid technique in data mining classification", 2015 2nd International Conference on Computing for Sustainable Global Development (INDIACom)

[20] Aditya Methaila, Prince Kansal, Himanshu Arya and Pankaj Kumar, "Early heart disease prediction using data mining techniques", Computer Science & Information Technology Journal, pp. 53-59, 2014.

Heart Disease Prediction Using Machine Learning Algorithms

Archana Singh
Computer Science and Engineering
Madan Mohan Malaviya University of Technology
Gorakhapur, 273010
archansingh7971@gmail.com

Rakesh Kumar
Computer Science and Engineering
Madan Mohan Malaviya University of Technology
Gorakhapur, 273010
rkiitr@gmail.com

Abstract—**Heart plays significant role in living organisms. Diagnosis and prediction of heart related diseases requires more precision, perfection and correctness because a little mistake can cause fatigue problem or death of the person, there are numerous death cases related to heart and their counting is increasing exponentially day by day. To deal with the problem there is essential need of prediction system for awareness about diseases. Machine learning is the branch of Artificial Intelligence(AI), it provides prestigious support in predicting any kind of event which take training from natural events. In this paper, we calculate accuracy of machine learning algorithms for predicting heart disease, for this algorithms are k-nearest neighbor, decision tree, linear regression and support vector machine(SVM) by using UCI repository dataset for training and testing. For implementation of Python programming Anaconda(jupytor) notebook is best tool, which have many type of library, header file, that make the work more accurate and precise.**

Keywords—supervised; unsupervised; reinforced; linear regression; decision tree; python programming; jupytor Notebook; confusion matrix;

I. Introduction

Heart is one of the most extensive and vital organ of human body so the care of heart is essential. Most of diseases are related to heart so the prediction about heart diseases is necessary and for this purpose comparative study needed in this field, today most of patient are died because their diseases are recognized at last stage due to lack of accuracy of instrument so there is need to know about the more efficient algorithms for diseases prediction.

Machine Learning is one of the efficient technology for the testing, which is based on training and testing. It is the branch of Artificial Intelligence(AI) which is one of broad area of learning where machines emulating human abilities, machine learning is a specific branch of AI. On the other hand machines learning systems are trained to learn how to process and make use of data hence the combination of both technology is also called as Machine Intelligence.

As the definition of machine learning, it learns from the natural phenomenon, natural things so in this project we uses the biological parameter as testing data such as cholesterol, Blood pressure, sex, age, etc. and on the basis of these, comparison is done in the terms of accuracy of algorithms such as in this project we have used four algorithms which are decision tree, linear regression, k-neighbour, SVM.

In this paper, we calculate the accuracy of four different machine learning approaches and on the basis of calculation we conclude that which one is best among them.

Section 1 of this paper consist the introduction about the machine learning and heart diseases. Section II described, the machine learning classification. Section III illustrated the related work of researchers. Section IV is about the methodology used for this prediction system. Section V is about the algorithms used in this project. Section VI briefly describes the dataset and their analysis with the result of this project. And the last Section VII concludes the summary of this paper with slight view about future scope of this paper.

II. MACHINE LEARNING

Machine Learning is one of efficient technology which is based on two terms namely testing and training i.e. system take training directly from data and experience and based on this training test should be applied on different type of need as per the algorithm required.
There are three type of machine learning algorithms:

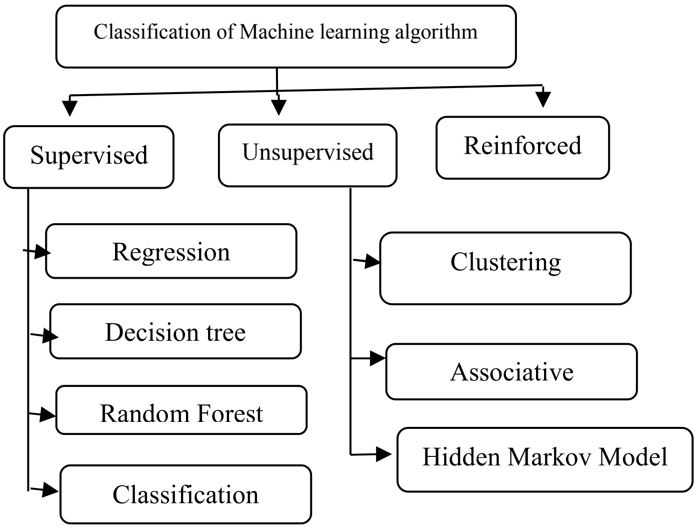


Fig.1 Classification of machine learning

978-1-7281-5846-4/20/$31.00 ©2020 IEEE

A. Supervised Learning

Supervised learning can be define as learning with the proper guide or you can say that learning in the present of teacher .we have a training dataset which act as the teacher for prediction on the given dataset that is for testing a data there are always a training dataset. Supervised learning is based on "train me" concept. Supervised learning have following processes:

- Classification
- Random Forest
- Decision tree
- Regression

To recognize patterns and measures probability of uninterruptable outcomes, is phenomenon of regression. System have ability to identify numbers, their values and grouping sense of numbers which means width and height, etc. There are following supervised machine learning algorithms:

- Linear Regression
- Logistical Regression
- Support Vector Machines (SVM)
- Neural Networks
- Random Forest
- Gradient Boosted Trees
- Decision Trees
- Naive Bayes

B. Unsupervised Learning

Unsupervised learning can be define as the learning without a guidance which in Unsupervised learning there are no teacher are guiding. In Unsupervised learning when a dataset is given it automatically work on the dataset and find the pattern and relationship between them and according to the created relationships, when new data is given it classify them and store in one of them relation . Unsupervised learning is based on "self sufficient " concept.

For example suppose there are combination fruits mango, banana and apple and when Unsupervised learning is applied it classify them in three different clusters on the basis if there relation with each other and when a new data is given it automatically send it to one of the cluster .

Supervisor learning say there are mango, banana and apple but Unsupervised learning said it as there are three different clusters. Unsupervised algorithms have following process:

- Dimensionality
- Clustering

There are following unsupervised machine learning algorithms:

- t-SNE
- k-means clustering
- PCA

C. Reinforcement

Reinforced learning is the agent ability to interact with the environment and find out the outcome. It is based on "hit and trial" concept. In reinforced learning each agent is awarded with positive and negative points and on the basis of positive points reinforced learning give the dataset output that is on the basis of positive awards it trained and on the basis of this training perform the testing on datasets

III. RELATED WORK

Heart is one of the core organ of human body, it play crucial role on blood pumping in human body which is as essential as the oxygen for human body so there is always need of protection of it, this is one of the big reasons for the researchers to work on this. So there are number of researchers working on it .There is always need of analysis of heart related things either diagnosis or prediction or you can say that protection of heart disease .There are various fields like artificial intelligence, machine learning, data mining that contributed on this work .

Performance of any algorithms depends on variance and biasness of dataset[4]. As per research on the machine learning for prediction of heart diseases himanshu et al.[4] naive bayes perform well with low variance and high biasness as compare to high variance and low biasness which is knn. With low biasness and high variance knn suffers from the problem of over fitting this is the reason why performance of knn get decreased. There are various advantage of using low variance and high biasness because as the dataset small it take less time for training as well as testing od algorithm but there also some disadvantages of using small size of dataset. When the dataset size get increasing the asymptotic errors are get introduced and low biasness, low variance based algorithms play well in this type of cases. Decision tree is one of the non-parametric machine learning algorithm but as we know it suffers from the problem over fitting but it cloud be solve by some over fitting removable techniques. Support vector machine is algebraic and statics background algorithm, it construct a linear separable n-dimensional hyper plan for the classification of datasets.

The nature of heart is complex, there is need of carefully handling of it otherwise it cause death of the person. The severity of heart diseases is classified based on various methods like knn, decision tree, generic algorithm and naïve bayes [3]. Mohan et al.[3] define how you can combine two different approaches to make a single approach called hybrid approach which have the accuracy 88.4% which is more than of all other.

Some of the researchers have worked on data mining for the prediction of heart diseases. Kaur et al.[6] have worked on this and define how the interesting pattern and knowledge are derived from the large dataset. They perform accuracy comparison on various machine learning and data mining

978-1-7281-5846-4/20/$31.00 ©2020 IEEE

approaches for finding which one is best among then and get the result on the favor of svm.

Kumar et al.[5] have worked on various machine learning and data mining algorithms and analysis of these algorithms are trained by UCI machine learning dataset which have 303 samples with 14 input feature and found svm is best among them, here other different algorithms are naivy bayes, knn and decision tree.

Gavhane et al.[1] have worked on the multi layer perceptron model for the prediction of heart diseases in human being and the accuracy of the algorithm using CAD technology. If the number of person using the prediction system for their diseases prediction then the awareness about the diseases is also going to increases and it make reduction in the death rate of heart patient.

Some researchers have work on one or two algorithm for predication diseases. Krishnan et al.[2] proved that decision tree is more accurate as compare to the naïve bayes classification algorithm in their project.

Machine learning algorithms are used for various type of diseases predication and many of the researchers have work on this like Kohali et al.[7] work on heart diseases prediction using logistic regression, diabetes prediction using support vector machine, breast cancer prediction using Adaboot classifier and concluded that the logistic regression give the accuracy of 87.1%, support vector machine give the accuracy of 85.71%, Adaboot classifier give the accuracy up to 98.57% which good for predication point of view.

A survey paper on heart diseases predication have proven that the old machine learning algorithms does not perform good accuracy for the predication while hybridization perform good and give better accuracy for the predication[8].

IV. METHODOLOGY OF SYSTEM

Processing of system start with the data collection for this we uses the UCI repository dataset which is well verified by number of researchers and authority of the UCI [15].

A. Data Collection

First step for predication system is data collection and deciding about the training and testing dataset. In this project we have used 73% training dataset and 37% dataset used as testing dataset the system.

B. Attribute Selection

Attribute of dataset are property of dataset which are used for system and for heart many attributes are like heart bit rate of person, gender of the person, age of the person and many more shown in TABLE.1 for predication system.

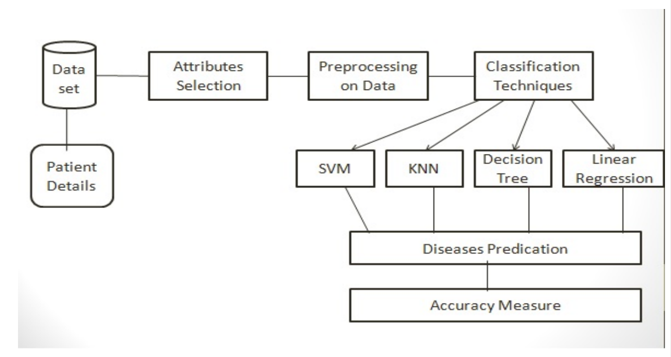


Fig.2 Architecture of Prediction System

TABLE.1 Attributes of the Dataset

S. No.	Attribute	Description	Type
1	Age	Patient's age (29 to 77)	Numaric
2	Sex	Gender of patient(male-0 female-1)	Nominal
3	Cp	Chest pain type	Nominal
4	Trestbps	Resting blood pressure(in mm Hg on admission to hospital ,values from 94 to 200)	Numerical
5	Chol	Serum cholesterol in mg/dl, values from 126 to 564)	Numerical
6	Fbs	Fasting blood sugar>120 mg/dl, true-1 false-0)	Nominal
7	Resting	Resting electrocardiographics result (0 to 1)	Nominal
8	Thali	Maximum heart rate achieved(71 to 202)	Numerical
9	Exang	Exercise included agina(1-yes 0-no)	Nominal
10	Oldpeak	ST depression introduced by exercise relative to rest (0 to .2)	Numerical
11	Slope	The slop of the peak exercise ST segment (0 to 1)	Nominal
12	Ca	Number of major vessels (0-3)	Numerical
13	Thal	3-normal	Nominal
14	Targets	1 or 0	Nominal

978-1-7281-5846-4/20/$31.00 ©2020 IEEE

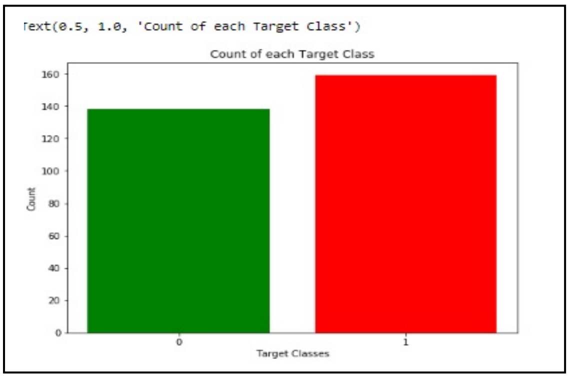
C. *Preprocessing of data*

Preprocessing needed for achieving prestigious result from the machine learning algorithms. For example Random forest algorithm does not support null values dataset and for this we have to manage null values from original raw data.

For our project we have to convert some categorized value by dummy value means in the form of "0"and "1" by using following code:

D. *Data Balancing*

Data balancing is essential for accurate result because by data balancing graph we can see that both the target classes are equal. Fig.3 represents the target classes where "0" represents with heart diseases patient and "1" represents no heart diseases pateints.



```
Text(0.5, 1.0, 'Count of each Target Class')
```

Fig.3 Target class view

E. *Histogram of attributes*

Histogram of attributes shows the range of dataset attributes and code which is used to create it.

dataset.hist()

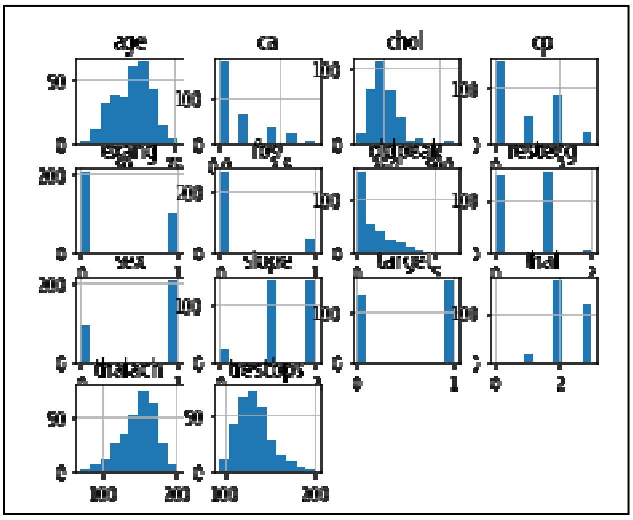


Fig.4 Histogram of attributes

V. MACHINE LEARNING ALGORITHMS

A. *Linear regression*

It is the supervised learning technique. It is based on the relationship between independent variable and dependent variable as seen in Fig.5 variable "x" and "y" are independent and dependent variable and relation between them is shown by equation of line which is linear in nature that why this approach is called linear regression.

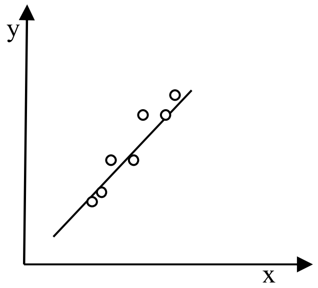


Fig.5 relation between x and y

It gives a relation equation to predict a dependent variable value "y" based on a independent variable value "x" as we can see in the Fig.5 so it is concluded that linear regression technique give the linear relationship between x(input) and y(output).

B. *Decision tree*

On the other hand decision tree is the graphical representation of the data and it is also the kind of supervised machine learning algorithms.

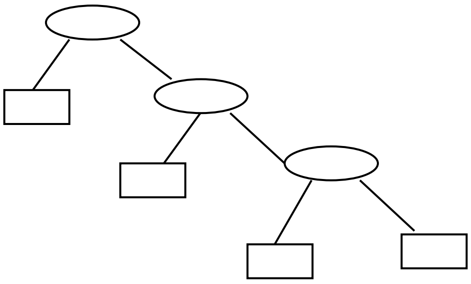


Fig.6 Decision tree

For the tree construction we use entropy of the data attributes and on the basis of attribute root and other nodes are drawn.

$$\text{Entropy}= -\sum Pij \log Pij \qquad (1)$$

In the above equation of entropy (1) Pij is probability of the node and according to it the entropy of each node is calculated. The node which have highest entropy calculation is selected as the root node and this process is repeated until all the nodes of the tree are calculated or until the tree constructed.

When the number of nodes are imbalanced then tree is create the over fitting problem which is not good for the

978-1-7281-5846-4/20/$31.00 ©2020 IEEE

calculation and this is one of reason why decision tree have less accuracy as compare to linear regression.

C. Support Vector Machine

It is one category of machine learning technique which work on the concept of hyperplan means it classify the data by creating hyper plan between them.

Training sample dataset is (Yi, Xi) where i=1,2,3,…….n and Xi is the ith vector, Yi is the target vector. Number of hyper plan decide the type of support vector such as example if a line is used as hyper plan then method is called linear support vector.

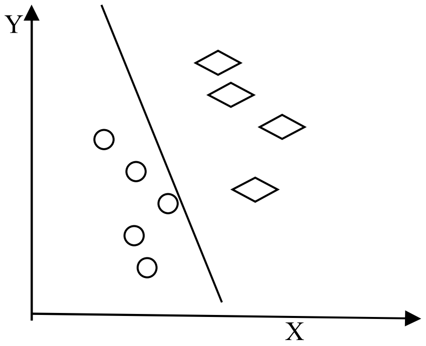


Fig.7 Linear Regression

D. K-nearest Neighbour

It work on the basis of distance between the location of data and on the basis of this distinct data are classified with each other. All the other group of data are called neighbor of each other and number of neighbor are decided by the user which play very crucial role in analysis of the dataset.

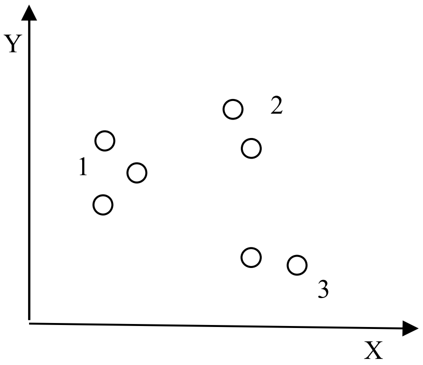


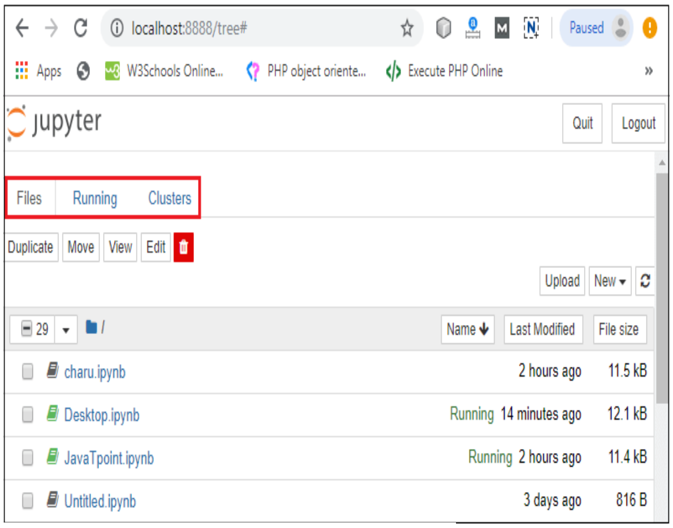
Fig.8 KNN where k=3

In the above Fig. k=3 shows that there are three neighbor that means three different type of data are there. Each cluster represented in two dimensional space whose coordinates are represented as (Xi,Yi) where Xi is the x-axis, Y represent y-axis and i= 1,2,3,….n.

VI. Result Analysis

A. About Jupytor Notebook

Jupiter notebook is used as the simulation tool and it is confortable for python programming projects. Jupytor notebook contains rich text elements and code also, which are figures, equations, links and many more. Because of the mix of rich text elements and code, these documents are perfect location to bring together an analysis description, and its results, as well as, they can execute data analysis in real time. Jupyter Notebook is an open-source, web-based interactive graphics, maps, plots, visualizations, and narrative text.



charu.ipynb	2 hours ago	11.5 kB
Desktop.ipynb	Running 14 minutes ago	12.1 kB
JavaTpoint.ipynb	Running 2 hours ago	11.4 kB
Untitled.ipynb	3 days ago	816 B

Fig.9 Jupyter Notebook

B. Accuracy calculation

Accuracy of the algorithms are depends on four values namely true positive(TP), false positive(FP), true negative(TN) and false negative(FN).

$$\text{Accuracy} = (FN+TP) / (TP+FP+TN+FN) \qquad (2)$$

The numerical value of TP, FP, TN, FN defines as:

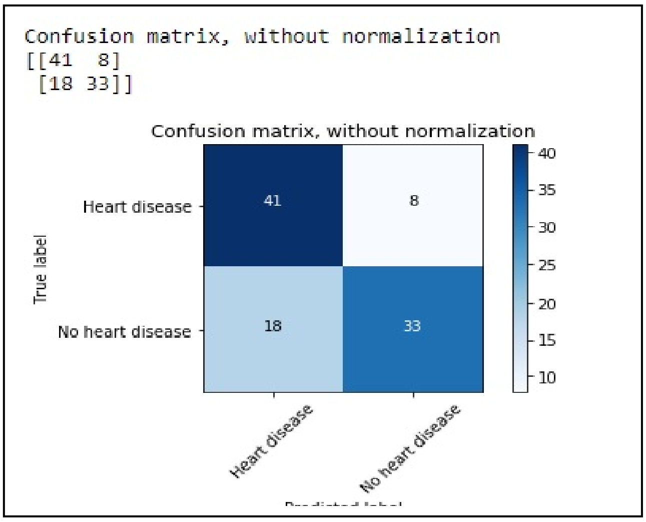
TP= Number of person with heart diseases

TN= Number of person with heart diseases and no heart diseases

FP= Number of person with no heart diseases

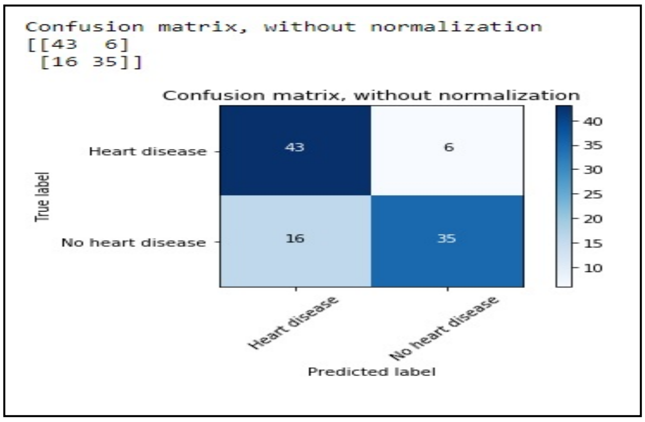
FN= Number of person with no heart diseases and with heart diseases

978-1-7281-5846-4/20/$31.00 ©2020 IEEE



```
Confusion matrix, without normalization
[[41  8]
 [18 33]]
```

Fig.10 Confusion matrix for Decision tree



```
Confusion matrix, without normalization
[[43  6]
 [16 35]]
```

Fig.11 Confusion Matrix for linear regression

C. Result

After performing the machine learning approach for testing and training we find that accuracy of the knn is much efficient as compare to other algorithms. Accuracy should be calculated with the support of confusion matrix of each algorithms as shown in Fig.6 and Fig.7 here number of count of TP, TN, FP, FN are given and using the equation (2) of accuracy, value has been calculated and it is conclude that knn is best among them with 87% accuracy and the comparison is shown in TABLE.2

TABLE.2 Accuracy comparison

Algorithm	Accuracy
Support Vector machine	83%
Decision tree	79%
Linear regression	78%
k-nearest neighbor	87%

VII. CONCLUSION AND FUTURE SCOPE

Heart is one of the essential and vital organ of human body and prediction about heart diseases is also important concern for the human beings so that the accuracy for algorithm is one of parameter for analysis of performance of algorithms. Accuracy of the algorithms in machine learning depends upon the dataset that used for training and testing purpose. When we perform the analysis of algorithms on the basis of dataset whose attributes are shown in TABLE.1 and on the basis of confusion matrix, we find KNN is best one.

For the Future Scope more machine learning approach will be used for best analysis of the heart diseases and for earlier prediction of diseases so that the rate of the death cases can be minimized by the awareness about the diseases.

References

[1] Santhana Krishnan J and Geetha S, “Prediction of Heart Disease using Machine Learning Algorithms” ICIICT, 2019.

[2] Aditi Gavhane, Gouthami Kokkula, Isha Panday, Prof. Kailash Devadkar, “Prediction of Heart Disease using Machine Learning”, Proceedings of the 2nd International conference on Electronics, Communication and Aerospace Technology(ICECA), 2018.

[3] Senthil kumar mohan, chandrasegar thirumalai and Gautam Srivastva, “Effective Heart Disease Prediction Using Hybrid Machine Learning Techniques” IEEE Access 2019.

[4] Himanshu Sharma and M A Rizvi, “Prediction of Heart Disease using Machine Learning Algorithms: A Survey” International Journal on Recent and Innovation Trends in Computing and Communication Volume: 5 Issue: 8 , IJRITCC August 2017.

[5] M. Nikhil Kumar, K. V. S. Koushik, K. Deepak, “Prediction of Heart Diseases Using Data Mining and Machine Learning Algorithms and Tools” International Journal of Scientific Research in Computer Science, Engineering and Information Technology ,IJSRCSEIT 2019.

[6] Amandeep Kaur and Jyoti Arora,“Heart Diseases Prediction using Data Mining Techniques: A survey” International Journal of Advanced Research in Computer Science , IJARCS 2015-2019.

[7] Pahulpreet Singh Kohli and Shriya Arora, “Application of Machine Learning in Diseases Prediction”, 4th International Conference on Computing Communication And Automation(ICCCA), 2018.

[8] M. Akhil, B. L. Deekshatulu, and P. Chandra, “Classification of Heart Disease Using K- Nearest Neighbor and Genetic Algorithm,” Procedia Technol., vol. 10, pp. 85–94, 2013.

[9] S. Kumra, R. Saxena, and S. Mehta, “An Extensive Review on Swarm Robotics,” pp. 140–145, 2009.

[10] Hazra, A., Mandal, S., Gupta, A. and Mukherjee, “ A Heart Disease Diagnosis and Prediction Using Machine Learning and Data Mining Techniques: A Review” Advances in Computational Sciences and Technology , 2017.

[11] Patel, J., Upadhyay, P. and Patel, “Heart Disease Prediction Using Machine learning and Data Mining Technique” Journals of Computer Science & Electronics , 2016.

[12] Chavan Patil, A.B. and Sonawane, P.“To Predict Heart Disease Risk and Medications Using Data Mining Techniques with an IoT Based Monitoring System for Post-Operative Heart Disease Patients” International Journal on Emerging Trends in Technology, 2017.

[13] V. Kirubha and S. M. Priya, “Survey on Data Mining Algorithms in Disease Prediction,” vol. 38, no. 3, pp. 124–128, 2016.

[14] M. A. Jabbar, P. Chandra, and B. L. Deekshatulu, “Prediction of risk score for heart disease using associative classification and hybrid feature subset selection,” Int. Conf. Intell. Syst. Des. Appl. ISDA, pp. 628–634, 2012.

[15] https://archive.ics.uci.edu/ml/datasets/Heart+Disease

978-1-7281-5846-4/20/$31.00 ©2020 IEEE

Heart Disease Prediction Using Machine Learning Algorithms

Pranitha Gadde
Assistant Professor, Department of Computer Science
ANITS,Visakhapatnam
India
pranitha.cse@anits.edu.in

Gunturu Deepthi
Student, Department of Computer Science
ANITS,Visakhapatnam
India
gunturudeepthi03@gmail.com

Cherukuri Shivani
Student, Department of Computer Science
ANITS,Visakhapatnam
India
cherukurishivani8899@gmail.com

Koruprolu Nagavinith
Student, Department of Computer Science
ANITS,Visakhapatnam
India
vineeth99.k@gmail.com

Kesuboyina Hanudeep Kumar
Student, Department of Computer Science
ANITS,Visakhapatnam
India
hanu.hunny@gmail.com

***Abstract*— The Heart is a vital organ in living beings. Millions of deaths occur worldwide due to heart diseases every year. Prediction and diagnosis of the diseases related to the heart require more accuracy, precision, and faultlessness, as the slightest mistake can cause various problems like fatigue and even result in the death of the person. It is an arduous task to predict the disease as it needs expertise and proficiency in the field. In various data repositories, large datasets are available which are used to solve real-world applications. In this paper, we calculate the accuracy of machine learning algorithms for predicting the possibility of heart disease. For this algorithms are naive bayes, decision tree, random forest, logistic regression, support vector machine (SVM), ada-boost, and xg-boost by using the UCI repository dataset for training and testing. All this is used in the frontend to predict whether the patient has Heart disease or not. We intend to draw out hidden patterns by applying various techniques, which are significant in causing heart disease and to predict its presence. Our objective is to find out a suitable machine learning technique that is computationally efficient as well as accurate for the prediction of heart disease.**

Keywords—svm; naive bayes; decision tree; random forest; logistic regression; adaboost; xgboost; python programming; confusion matrix; correlation matrix

I. Introduction

The Heart is an indispensable organ in Human beings. Heart disease is the main reason for the deaths of many people in the world. As per WHO, every year 12 million deaths are caused due to cardiovascular disease. Heart Disease is like a silent killer which results in the death of a person without obvious symptoms. Early identification of disease leads to prevention of disease and which in turn reduces the complications. As we say, prevention is better than cure, preventing heart disease can be able to prevent many premature deaths and reduce the mortality rate.

Doctors may not be able to monitor the patient for 24 hours. Although there are lots of instruments in the market, they are not capable of detecting heart disease accurately and some of the instruments are very expensive and would also require expertise in the field. Machine learning is a trending technology, which is a subclass of artificial intelligence. Machine learning allows machines to enhance at tasks with experience. Machine learning enables a system to identify patterns by itself and make predictions.

In this project, we use machine learning to predict whether a person is having heart disease or not. We consider various attributes of patients like gender, chest pain type, fasting blood pressure, serum cholesterol, exang, etc. We utilize various algorithms like SVM, Naive Bayes, Decision Tree, Random Forest, Logistic Regression, Ada-boost, Xg-boost. Based on attributes, we perform a comparative analysis of algorithms regarding the accuracy, and whichever algorithm is giving better accuracy, is considered for heart disease prediction.

II. Related work

Heart plays a vital role in human beings. At present, most of the deaths are due to cardiovascular disease. Early prediction of heart disease can save lots of lives. Many researchers are working on heart disease prediction using various technologies like artificial intelligence, machine learning. The following are some works related to heart disease prediction.

[1] Purushottam ,et ,al proposed a paper “Efficient Heart Disease Prediction System” using hill climbing and decision tree algorithms .They used 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 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%.

[2]Santhana Krishnan. J ,et ,al proposed a paper "Prediction of Heart Disease Using Machine Learning Algorithms" using decision tree and naive bayes algorithm for prediction of heart disease.In decision tree algorithm the tree is built using certain conditions which gives True or False decisions. The algorithms like SVM, KNN are results based on vertical or horizontal split conditions depends on dependent variables. But decision tree for a tree like structure having root node, leaves and branches base on the decision made in each of tree Decision tree also help in the understating the importance of the attributes in the dataset. They have also used Cleveland data set. Dataset splits in 70% training and 30% testing by using some methods. This algorithm gives 91% accuracy. The second algorithm is Naive Bayes,which is used for classification.It can handle complicated, nonlinear, dependent data so it is found suitable for heart disease dataset as this dataset is also complicated, dependent and nonlinear in nature. This algorithm gives an 87% accuracy.

[3]Sonam Nikhar et al proposed paper " Prediction of Heart Disease Using Machine Learning Algorithms" their research gives point to point explanation of Naïve Bayes and decision tree classifier that are used especially in the prediction of Heart Disease. Some analysis has been led to think about the execution of prescient data mining strategy on the same dataset, and the result decided that Decision Tree has highest accuracy than Bayesian classifier.

[4]Aditi Gavhane et al proposed a paper "Prediction of Heart Disease Using Machine Learning",in which training and testing of dataset is 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 layers between these two input and output layers.Through hidden layers each input node is connected to output layer. This connection is assigned with some random weights.The other input is called bias which is assigned with weight b. Based on requirement the connection between the nodes can be feedforwarded or feedback.

[5]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 neighbour ,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).

[6]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 the heart disease among people different neural systems and data mining techniques are used.

[7]Abhay Kishore et al,,proposed "Heart Attack Prediction Using Deep Learning" in which heart attack prediction system by using Deep learning techniques and to predict the probable aspects of heart related infections of the patient Recurrent Neural System is used.This model uses deep learning and data mining to give the best precise model and least blunders.This paper acts as strong reference model for another type of heart attack prediction models

[8]Senthil Kumar Mohan et al, proposed "Effective Heart Disease Prediction Using Hybrid Machine Learning Techniques" in which their main objective is to improve exactness in cardiovascular problems.The algorithms used are KNN, LR, SVM, NN to produce an improved exhibition level with a precision level of 88.7% through the prediction model for heart disease with hybrid random forest with linear model(HRFLM).

[9]Anjan N. Repaka et al,proposed a model stated the performance of prediction for two classification models, which is analyzed and compared to previous work. The Experimental results shows that accuracy is improved in finding the percentage of risk prediction of our proposed method in comparison with other models.

[10]Aakash Chauhan et al, proposed "Heart Disease Prediction using Evolutionary Rule Learning". Data is directly retrieved from electronic records that reduces the manual tasks. The amount of services are decreased and shown major number of rules helps within the best prediction of heart disease.Frequent pattern growth association mining is performed on patient's dataset to generate strong association rules.

III. METHODOLOGY OF SYSTEM

The system architecture gives an overview of the working of the system. The working of the system starts with the collection of data and selecting the important attributes. Then the required data is preprocessed into the required format. The data is then divided into two parts training and testing data. The algorithms are applied and the model is trained using the training data. The accuracy of the system is obtained by testing the system using the testing data.

This system is implemented using the following modules.

1.) Collection of Dataset
2.) Selection of attributes
3.) Data Pre-Processing

4.) Balancing of Data
5.) Disease Prediction

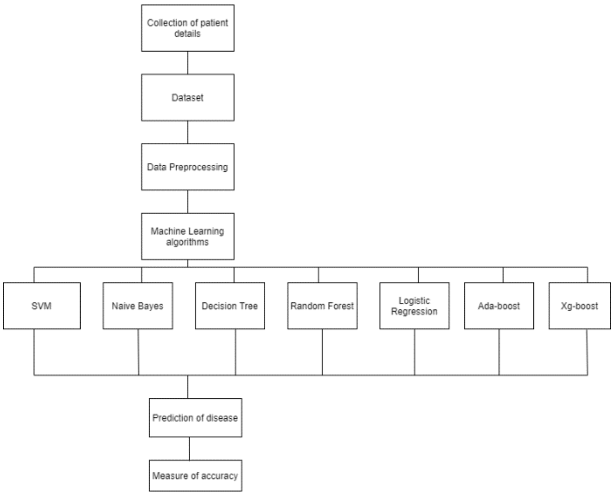


Fig: Architecture of Prediction System

1) Collection of dataset:

Initially, we collect a dataset for our heart disease prediction system. After the collection of the dataset, we split the dataset into training data and testing data. The training dataset is used for prediction model learning and testing data is used for evaluating the prediction model. For this project, 70% of training data is used and 30% of data is used for testing.
The dataset used for this project is Heart Disease UCI. The dataset consists of 76 attributes; out of which, 14 attributes are used for the system.

2) Selection of attributes

Attribute or Feature selection includes the selection of appropriate attributes for the prediction system. This is used to increase the efficiency of the system. Various attributes of the patient like gender, chest pain type, fasting blood pressure, serum cholesterol, exang, etc are selected for the prediction. The Correlation matrix is used for attribute selection for this model.

3) Pre-processing of Data

Data pre-processing is an important step for the creation of a machine learning model. Initially, data may not be clean or in the required format for the model which can cause misleading outcomes. In pre-processing of data, we transform data into our required format.
It is used to deal with noises, duplicates, and missing values of the dataset. Data pre-processing has the activities like importing datasets, splitting datasets, attribute scaling, etc. Pre-processing of data is required for improving the accuracy of the model.

4) Balancing of Data

Imbalanced datasets can be balanced in two ways. They are Under Sampling and Over Sampling

1. Under Sampling:
In Under Sampling, dataset balance is done by the reduction of the size of the ample class. This process is considered when the amount of data is adequate.

2. Over Sampling:
In Over Sampling, dataset balance is done by increasing the size of the scarce samples. This process is considered when the amount of data is inadequate.

5) Prediction of Disease

Various machine learning algorithms like SVM, Naive Bayes, Decision Tree, Random Tree, Logistic Regression, Ada-boost, Xg-boost are used for classification. Comparative analysis is performed among algorithms and the algorithm that gives the highest accuracy is used for heart disease prediction.

IV. MACHINE LEARNING ALGORITHMS

Machine learning is a powerful technology that is a systematic study of various algorithms that provide the system with the potential to replicate human learning activities without being actually programmed. Machine learning is further divided into three types: Unsupervised Learning, Supervised Learning, and Reinforcement Learning.

1. Naive Bayes

Naive Bayes algorithm is used to resolve classification problems. It is a supervised machine learning algorithm, which is based on the Bayes theorem. Naïve Bayes Classifier is one of the simple and most effective Classification algorithms which helps in building fast machine learning models that can make quick predictions. It is mainly used in data classification that includes a high-level training dataset. The Naive Bayes algorithm predicts the data based on probability, so it is also known as a probabilistic classifier. Naive Bayes classifier assumes that every particular feature in the dataset is independent of all other features.

$$P (h|D) = \frac{(P(D|h))*(P(h))}{P(D)}$$

(1) P (D): the probability of the data (regardless of the hypothesis). This is known as the marginal probability or probability of evidence.
(2) P (h): the probability of hypothesis h being true (regardless of the data). This is referred to as the prior probability of h.
(3) P (h|D): the probability of hypothesis h given the data D. This is known as posterior probability.
(4) P (D|h): the probability of data d given that the hypothesis h was true. This is known as likelihood probability.

2. Logistic Regression:

Logistic regression is also a supervised learning classification algorithm that is used to solve both classification and regression problems. In classification problems, the target variable may be in a binary or discrete format either 0 or 1. Logistic regression algorithm works on the sigmoid function, so the categorical variable results as 0 or 1, Yes or No, True or False, etc. It is a predictive analysis algorithm that works on mathematical functions.

Logistic regression uses a sigmoid function or logistic function which is a complex cost function. The sigmoid functions return the value between 0 and 1. If the value less than 0.5 then it is considered as 0 and greater than 0.5 it is considered as 1. Thus to build a model using logistic regression sigmoid function is required.

There are three main types of logistic regression:

1) Binomial: The target variable can have only 2 possibilities either “0” or “1” which may represent “win” or “loss”, “pass” or “fail”, “true” or “false”, etc.

2) Multinomial: Here, the target variable can have 3 or more possibilities that are not ordered which means it has no measure in quantity like “disease A” or “disease B” or “disease C”.

3) Ordinal: In this case, the target variables deal with ordered categories. For example, a test score can be categorized as: “poor”, “average”, “good”, and “excellent”. Here, each category can be given a score like 0, 1, 2, and 3.

$$f(x) = \frac{1}{(1+e^{X})}$$

f(x) = Output between the 0 and 1 value

e = base of the natural logarithm

x = input to the function.

The value of the logistic regression must range from 0 to 1, does not go beyond this limit, so the only possible curve formed is S-shaped. The S-form curve formed is known as the sigmoid function or the logistic function. In logistic regression, the threshold value plays an important role, which defines the probability of either 0 or 1. The values above the threshold value reach 1, and a value below the threshold value reaches 0.

3. Random Forest

Random Forest classifier is a supervised learning technique in machine learning. It can be used to solve both Classification and Regression problems in machine learning. It is based on the process of combining multiple classifiers to solve a complex problem and to improve the performance of the model, which is known as ensemble learning. Random Forest consists of several decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset. Rather than relying on a single decision tree, the random forest acquires the prediction from each tree, and based on the majority of votes for predictions, it predicts the final output. The higher number of trees in the forest leads to better accuracy and also prevents the problem of over fitting. The final output is taken by using the majority voting classifier for a classification problem while in the case of a regression problem the final output is the mean of all the outputs.

4. Support Vector Machine

Support vector machine (SVM) is a supervised learning algorithm that is used to analyze data. It is used to resolve classification and regression problems. An SVM model is a delineation of the examples as points in space, mapped so that the examples of the discrete categories are divided by a clear gap. The points are separated by a plane which is known as a hyper plane. A set of training data is given to it to mark them as belonging to either one of two categories; an SVM training algorithm then builds a model that assigns new examples of the same space are mapped and then predicts to which category they belong, making it a non-probabilistic binary linear classifier. SVM can be of two types:

- Linear SVM: Linear SVM is used for data that is linearly separable. It means if a dataset can be segregated into two different classes by using a single straight line, then such data is labelled as linearly separable data, and the classifier is used called a Linear SVM classifier.
- Non-linear SVM: Non-Linear SVM is used for data that cannot be separated linearly, which means if a dataset cannot be sorted by using a straight line, then such data is referred to as non-linear data and the classifier used is called a Non-linear SVM classifier.

5. Decision Tree

Decision Tree algorithm is also a supervised learning technique, mostly preferred for solving Classification problems but can be used for both classification and regression problems. The decision tree is a tree-structured classifier, where branches represent the decision rules which are used to make any decision and have multiple branches, internal nodes represent the features of a dataset, and each leaf node represents the outcome of the decisions and does not contain any further branches. It is a graphical representation for getting all the possible solutions to a problem/decision based on given constraints. The decisions or the analysis are performed based on features of the given dataset. A decision tree simply asks a question and based on the answer, it further splits the tree into sub trees.

6. Ada-boost

AdaBoost is short for Adaptive Boosting and is a widely accepted boosting technique that combines multiple weak classifiers to build a strong classifier. It is done by building a model using a series of weak models. AdaBoost was developed for binary classification. It selects a training subset randomly and builds a model. It then iteratively trains the AdaBoost machine learning model by selecting the training set based on the accurate prediction of the last training. It assigns the higher weight to the erroneously classified observations so that in the next iteration these observations would have a higher prospect for classification. This is done to correct the errors present in the first model. It also assigns weight to the trained classifier in every iteration according to the accuracy of the classifier. The more accurate classifier will get high weight. This process iterates until the entire training data fits without any error or until it reaches the specified maximum number of models are added.

7. Xg-boost

Xgboost is short for Extreme Gradient Boosting. XgBoost is an ensemble method based on decision trees that build out a strong learner from several weak learners. XgBoost algorithm is used to boost the performance of the model and is used to provide better accuracy. In this, decision trees are built sequentially. Weights are randomly assigned to all the features which are independent of each other and fed to the decision tree which predicts the results. The weights of wrongly predicted features by the decision trees are increased and these features are sent to the next decision tree. The correctly predicted weights of features by the decision trees are reduced. This process continues sequentially until correct results are predicted. These individual classifiers then combine to produce a strong and more precise model. Xgboost is used for solving both regression and classification problems.

V. Performance Analysis

In this project, various machine learning algorithms like SVM, Naive Bayes, Decision Tree, Random Forest, Logistic Regression, Adaboost, Xgboost are used to predict heart disease. Heart Disease UCI dataset, has a total of 76 attributes, out of those only 14 attributes are considered for the prediction of heart disease. Various attributes of the patient like gender, chest pain type, fasting blood pressure, serum cholesterol, exang, etc are considered for this project. The accuracy for individual algorithms has to measure and whichever algorithm is giving the best accuracy, that is considered for the heart disease prediction. For evaluating the experiment, various evaluation metrics like accuracy, confusion matrix, precision, recall, and f1-score are considered.

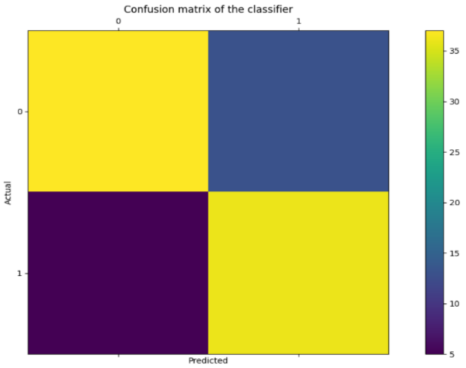
Accuracy- Accuracy is the ratio of the number of correct predictions to the total number of inputs in the dataset.

It is expressed as:

$$\text{Accuracy} = \frac{(TP+TN)}{(TP+FP+FN+TN)}$$

Confusion Matrix- It gives us a matrix as output and gives the total performance of the system.

Fig: Confusion Matrix

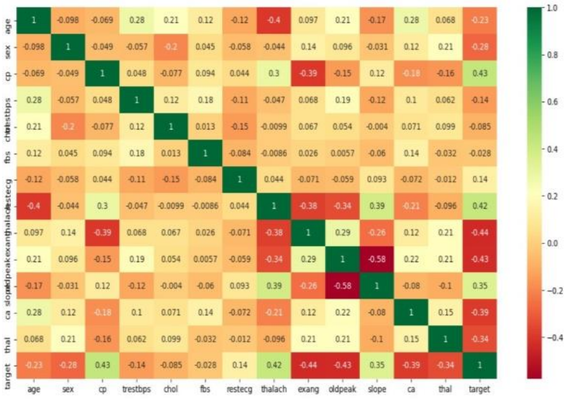


Where TP: True positive
FP: False Positive
FN: False Negative
TN: True Negative

Correlation Matrix:

The correlation matrix in machine learning is used for feature selection. It represents dependency between various attributes.

Fig: Correlation matrix



Precision- It is the ratio of correct positive results to the total number of positive results predicted by the system.

It is expressed as:

$$\text{Precision(P)} = \frac{\text{TP}}{(\text{TP + FP})}$$

Recall- It is the ratio of correct positive results to the total number of positive results predicted by the system.

It is expressed as:

$$\text{Recall(R)} = \frac{\text{TP}}{(\text{TP +FN})}$$

F1 score- It is the harmonic mean of Precision and Recall. It measures the test accuracy. The range of this metric is 0 to 1.

It is expressed as:

$$\text{F1 score} = 2 * \frac{1}{\left(\frac{1}{\text{Precision}}\right)+\left(\frac{1}{\text{Recall}}\right)} = \frac{2PR}{(P+R)}$$

VI. Result

After performing the machine learning approach for training and testing we find that accuracy of the xgboost is better compared to other algorithms. Accuracy is calculated with the support of the confusion matrix of each algorithm, here the number count of TP, TN, FP, FN is given and using the equation of accuracy, value has been calculated and it is concluded that extreme gradient boosting is best with 81% accuracy and the comparison is shown below.

TABLE: Accuracy comparison of algorithms

Algorithm	Accuracy
XGBoost	81.3%
SVM	80.2%
Logistic Regression	79.1%
Random Forest	79.1%
Naive Bayes	76.9%
Decision Tree	75.8%
Adaboost	73.6%

The highest accuracy is given by the XGBoost algorithm

VII. Conclusion And Future Scope

Heart diseases are a major killer in India and throughout the world, 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 diseases is on a raise each year. This prompts for its early diagnosis and treatment. The utilization of suitable technology support in this regard can prove to be highly beneficial to the medical fraternity and patients. In this paper, the seven different machine learning algorithms used to measure the performance are SVM, Decision Tree, Random Forest, Naïve Bayes, Logistic Regression, Adaptive Boosting, and Extreme Gradient Boosting applied on the dataset. The expected attributes leading to heart disease in patients are available in the dataset which contains 76 features and 14 important features that are useful to evaluate the system are selected among them. If all the features taken into the consideration then the efficiency of the system the author gets is less. To increase efficiency, attribute selection is done. In this n features have to be selected for evaluating the model which gives more accuracy. The correlation of some features in the dataset is almost equal and so they are removed. If all the attributes present in the dataset are taken into account then the efficiency decreases considerably. All the seven machine learning methods accuracies are compared based on which one prediction model is generated. Hence, the aim is to use various evaluation metrics like confusion matrix, accuracy, precision, recall, and f1-score which predicts the disease efficiently. Comparing all seven the extreme gradient boosting classifier gives the highest accuracy of 81%.

VIII. References

[1] Soni J, Ansari U, Sharma D & Soni S (2011). Predictive data mining for medical diagnosis: an overview of heart disease prediction. International Journal of Computer Applications, 17(8), 43-8

[2] Dangare C S & Apte S S (2012). Improved study of heart disease prediction system using data mining classification techniques. International Journal of Computer Applications, 47(10), 44-8.

[3] Ordonez C (2006). Association rule discovery with the train and test approach for heart disease prediction. IEEE Transactions on Information Technology in Biomedicine, 10(2), 334-43.

[4] Shinde R, Arjun S, Patil P & Waghmare J (2015). An intelligent heart disease prediction system using k-means clustering and Naïve Bayes algorithm. International Journal of Computer Science and Information Technologies, 6(1), 637-9.

[5] Bashir S, Qamar U & Javed M Y (2014, November). An ensemble-based decision support framework for intelligent heart disease diagnosis. In International Conference on Information Society (i-Society 2014) (pp. 259-64). IEEE. ICCRDA 2020 IOP Conf. Series: Materials Science and Engineering 1022 (2021) 012072 IOP Publishing doi:10.1088/1757-899X/1022/1/012072 9

[6] Jee S H, Jang Y, Oh D J, Oh B H, Lee S H, Park S W & Yun Y D (2014). A coronary heart disease prediction model: the Korean Heart Study. BMJ open, 4(5), e005025.

[7] Ganna A, Magnusson P K, Pedersen N L, de Faire U, Reilly M, Ärnlöv J & Ingelsson E (2013). Multilocus genetic risk scores for coronary heart disease prediction. Arteriosclerosis, thrombosis, and vascular biology, 33(9), 2267-72.

[8] Jabbar M A, Deekshatulu B L & Chandra P (2013, March). Heart disease prediction using lazy associative classification. In 2013 International Mutli-Conference on Automation, Computing,Communication, Control and Compressed Sensing (iMac4s) (pp. 40- 6). IEEE.

[9] Brown N, Young T, Gray D, Skene A M & Hampton J R (1997). Inpatient deaths from acute myocardial infarction, 1982-92: analysis of data in the Nottingham heart attack register. BMJ, 315(7101), 159-64.

[10] Folsom A R, Prineas R J, Kaye S A & Soler J T (1989). Body fat distribution and self-reported prevalence of hypertension, heart attack, and other heart disease in older women. International journal of epidemiologyy, 18(2), 361-7.

[11] Chen A H, Huang S Y, Hong P S, Cheng C H & Lin E J (2011, September). HDPS: Heart disease prediction system. In 2011 Computing in Cardiology (pp. 557-60). IEEE.

[12] Parthiban, Latha and R Subramanian. "Intelligent heart disease prediction system using CANFIS and genetic algorithm." International Journal of Biological, Biomedical and Medical Sciences 3.3 (2008).

[13] Wolgast G, Ehrenborg C, Israelsson A, Helander J, Johansson E & Manefjord H (2016). Wireless body area network for heart attack detection [Education Corner]. IEEE antennas and propagation magazine, 58(5), 84-92.

[14] Patel S & Chauhan Y (2014). Heart attack detection and medical attention using motion sensing device -kinect. International Journal of Scientific and Research Publications, 4(1), 1-4.

[15] Piller L B, Davis B R, Cutler J A, Cushman W C, Wright J T, Williamson J D & Haywood L J (2002). Validation of heart failure events in the Antihypertensive and Lipid Lowering Treatment to Prevent Heart Attack Trial (ALLHAT) participants assigned to doxazosin and chlorthalidone. Current controlled trials in cardiovascular medicine

[16] Raihan M, Mondal S, More A, Sagor M O F, Sikder G, Majumder M A & Ghosh K (2016, December). Smartphone based ischemic heart disease (heart attack) risk prediction using clinical data and data mining approaches, a prototype design. In 2016 19th International Conference on Computer and Information Technology (ICCIT) (pp. 299-303). IEEE.

[17] A. Aldallal and A. A. A. Al-Moosa, "Using Data Mining Techniques to Predict Diabetes and Heart Diseases", *2018 4th International Conference on Frontiers of Signal Processing (ICFSP)*, pp. 150-154, 2018, September.

[18] Takci H (2018). Improvement of heart attack prediction by the feature selection methods. Turkish Journal of Electrical Engineering & Computer Sciences, 26(1), 1-10.

[19] Ankita Dewan and Meghna Sharma, "Prediction of heart disease using a hybrid technique in data mining classification", *2015 2nd International Conference on Computing for Sustainable Global Development (INDIACom)*

[20] Aditya Methaila, Prince Kansal, Himanshu Arya and Pankaj Kumar, "Early heart disease prediction using data mining techniques", *Computer Science & Information Technology Journal*, pp. 53-59, 2014.