HEART ATTACK PREDICTION USING XGBOOST

A PROJECT REPORT

Submitted by

MYAKALA CHAITANYA RAJEEV [RA1611003010350]
CHAPPIDI SAI CHAITANYA [RA1611003010374]

Under the guidance of
Mr. M. KARTHIKEYAN
(Assistant Professor, Department of Computer Science & Engineering)

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

of

FACULTY OF ENGINEERING AND TECHNOLOGY

SRM
INSTITUTE OF SCIENCE & TECHNOLOGY
Deemed to be University u/s 3 of UGC Act, 1956

S.R.M. Nagar, Kattankulathur, Kancheepuram District

MAY 2020

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

(Under Section 3 of UGC Act, 1956)

BONAFIDE CERTIFICATE

Certified that this project report titled " **HEART ATTACK PREDICTION USING XGBOOST** " is the bonafide work of "**MYAKALA CHAITANYA RAJEEV [RA1611003010350],CHAPPIDI SAI CHAITANYA [RA1611003010374]**", who carried out the project work under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE

M. Karthikeyan

Mr. M. KARTHIKEYAN
GUIDE
Assistant Professor
Dept. of Computer Science & Engineering

SIGNATURE

Dr. B. AMUTHA
HEAD OF THE DEPARTMENT
Dept. of Computer Science and Engineering

Signature of the Internal Examiner

Signature of the External Examiner

Own Work Declaration

Department of Computer Science and Engineering

SRM Institute of Science & Technology

Own Work* Declaration Form

This sheet must be filled in (each box ticked to show that the condition has been met). It must be signed and dated along with your student registration number and included with all assignments you submit – work will not be marked unless this is done.

To be completed by the student for all assessments

Degree/ Course : B.Tech / CSE

Student Name : Myakala Chaitanya Rajeev

Registration Number : RA1611003010350

Title of Work : Heart Attack Prediction using XGBoost

I / We hereby certify that this assessment compiles with the University's Rules and Regulations relating to Academic misconduct and plagiarism**, as listed in the University Website, Regulations, and the Education Committee guidelines.

I / We confirm that all the work contained in this assessment is my / our own except where indicated, and that I / We have met the following conditions:

- Clearly references / listed all sources as appropriate
- Referenced and put in inverted commas all quoted text (from books, web, etc)
- Given the sources of all pictures, data etc. that are not my own
- Not made any use of the report(s) or essay(s) of any other student(s) either past or present
- Acknowledged in appropriate places any help that I have received from others (e.g. fellow students, technicians, statisticians, external sources)
- Compiled with any other plagiarism criteria specified in the Course handbook / University website

I understand that any false claim for this work will be penalised in accordance with the University policies and regulations.

DECLARATION:

I am aware of and understand the University's policy on Academic misconduct and plagiarism and I certify that this assessment is my / our own work, except where indicated by referring, and that I have followed the good academic practices noted above.

If you are working in a group, please write your registration numbers and sign with the date for every student in your group.

Own Work Declaration

Department of Computer Science and Engineering

SRM Institute of Science & Technology

Own Work* Declaration Form

This sheet must be filled in (each box ticked to show that the condition has been met). It must be signed and dated along with your student registration number and included with all assignments you submit – work will not be marked unless this is done.

To be completed by the student for all assessments

Degree/ Course : B.Tech / CSE

Student Name : Chappidi Sai Chaitanya Reddy

Registration Number : RA1611003010374

Title of Work : Heart Attack Prediction using XGBoost

I / We hereby certify that this assessment compiles with the University's Rules and Regulations relating to Academic misconduct and plagiarism**, as listed in the University Website, Regulations, and the Education Committee guidelines.

I / We confirm that all the work contained in this assessment is my / our own except where indicated, and that I / We have met the following conditions:

- Clearly references / listed all sources as appropriate
- Referenced and put in inverted commas all quoted text (from books, web, etc)
- Given the sources of all pictures, data etc. that are not my own
- Not made any use of the report(s) or essay(s) of any other student(s) either past or present
- Acknowledged in appropriate places any help that I have received from others (e.g. fellow students, technicians, statisticians, external sources)
- Compiled with any other plagiarism criteria specified in the Course handbook / University website

I understand that any false claim for this work will be penalised in accordance with the University policies and regulations.

DECLARATION:

I am aware of and understand the University's policy on Academic misconduct and plagiarism and I certify that this assessment is my / our own work, except where indicated by referring, and that I have followed the good academic practices noted above.

If you are working in a group, please write your registration numbers and sign with the date for every student in your group.

Ch.Sai Chaitanya Reddy

ACKNOWLEDGEMENT

We express our humble gratitude to **Dr. Sandeep Sancheti**, Vice Chancellor, SRM Institute of Science and Technology,for the facilities extended for the project work and his continued support.

We extend our sincere thanks to **Dr. C. Muthamizhchelvan**, Director, Faculty of Engineering and Technology, SRM Institute of Science and Technology, for his invaluable support.

We wish to thank **Dr.B.Amutha**, Professor & Head, Department of Computer Science and Engineering, SRM Institute of Science and Technology, for her valuable suggestions and encouragement throughout the period of theproject work.

We are extremely grateful to our Academic Advisor **Dr.A.JeyaSekar,** Associate Professor,and **Dr.R.Annie Uthra** ,Associate Professor,Department of Computer Science and Engineering, SRM Institute of Science and Technology, for their great support at all the stages of project work.

We would like to convey our thanks to our Panel Head, [**Mrs.K.R.JANSI],** [Assitant Professor (Sr.G)], Department of Computer Science and Engineering, SRM Institute of Science and Technology, for her inputs during the project reviews.

We register our immeasurable thanks to our Faculty Advisor**, [Mrs.P. Akilandeswari],**[Assistant Professor (Sr.G)], Department of Computer Science and Engineering, SRM Institute of Science and Technology, for leading and helping us to complete our course.

Our inexpressible respect and thanks to my guide, **[Mr. M. Karthikeyan],**[Assistant Professor (O.G)], Department of Computer Science and Engineering, SRM Institute of Science and Technology, for providing me an opportunity to pursue my project under his mentorship. He provided me the freedom and support to explore the research topics of my interest. His passion for solving the real problems and making a difference in the world has always been inspiring.

We sincerely thank staff and students of the Computer Science and Engineering Department, SRM Institute of Science and Technology, for their help during my research. Finally, we would like to thank my parents, our family members and our friends for their unconditional love, constant support and encouragement.

[Myakala Chaitanya Rajeev]

[Chappidi Sai Chaitanya]

ABSTRACT

Heart disease has become more common these days. Heart attacks are the majority death cases in the current time. Lot of risk involved in people's life. Machine learning has been effective in taking decisions and predicting from huge amount of data set given by the medical healthcare industry. We have seen Machine Learning techniques are been used irrespective of the fields. There have been various factors that have been affecting the risk to the lives. Variation in Blood Pressure, sugar, pulse rate, shortness of breath. Etc. can lead to cardiovascular diseases that in turn block the blood vessels that carry rich oxygenated blood. This may cause coronary artery disease, heart failure, congenital heart disease. There are several attributes that are taken into account in order to predict the heart attack. There are many forms of heart diseases that can be predicted through various factors and can be diagnosed with various medical tests. The main aim of this project is to predict heart attack with at most accuracy. We have used Extreme gradient boosting algorithm for predication of heart attack. The pre-research and reading obtained front this algorithm is used in detection of heart attack at early level and can be cured by proper diagnosis. We have produced an efficient performance level with an accuracy level over 90% though the prediction model for predicting of heart attack. Using machine learning to predict this kind of diseases is a major necessity for bringing the healthcare industry to great heights.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

AI	Artificial Intelligence
UI	User Interface
ML	Machine Learning
UCI	University of California
SVM	Support Vector Machine
XG	eXtreme Gradient
ECG	Electrocardiography
BP	Blood Pressure
RF	Random Forest
LM	Linear Method
NN	Neaural Network
TP	True Positive
TN	True Negative
FP	False Positive
FN	False Negative
KNN	K-Nearest Neighbors

CHAPTER 1

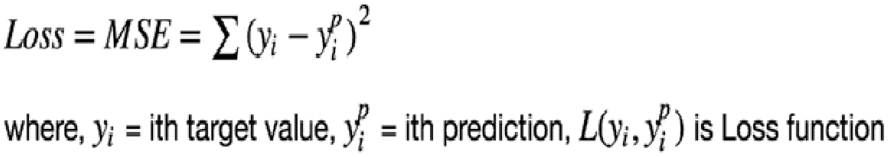
INTRODUCTION

1.1 General Overview

Medicinal services field has a tremendous measure of information, for processing those information certain methods are utilized. data mining is one of the strategies frequently utilized. Coronary illness is the Leading reason for death around the world. This System predicts the emerging prospects of Heart Disease. The results of this framework give the odds of happening coronary illness as far as rate. The datasets utilized are arranged regarding clinical parameters. This framework assesses those parameters utilizing the data mining characterization method. The datasets are handled in python programming utilizing many Machine Learning Algorithms. The way to improve coronary illness human services execution and diminish the demise rate is transforming the passive healthcare mode into an unavoidable way. There are various works related to the prediction of diseases using various data mining algorithms and machine learning algorithms in clinical focuses. Different Machine Learning (ML) algorithms like Linear Regression, Logistic Regression, Naïve Baye's, Decision Trees, Support Vector Machine, Random ForestRandom Forest (RF), eXtreme Gradient (XG)Boost have been used. Machine learning involves the use of algorithms built based on the existing information to make predictions and take the decision without being unanticipated or depending on an external command.

1.2 Heart Attack Prediction Using XGBOOST

Machine Learning is one of the best applications of Artificial Intelligence Artificial Intelligence (AI) and the one that makes the computers capable to learn without any explicit programming. Ma- chine learning involves the use of algorithms built based on the existing information to make predictions and take the decision without being unanticipated or depending on an external command. Machine learning is one of the most popular technologies in the world and has been explored a lot by experts on its usage and applications. Every top machine learning development company seeks to employ the best skills to render bet- ter service. The main aim is to make the systems capable of doing various intellectual activities that humans do. This is to reduce the need for human assistance and to auto- mate repeated tasks. As you learn about machine learning you would be able to develop software that enables the machine to understand, analyze and respond to the situation promptly. Machine becoming acquainted with a product of engineered insight (AI) that gives frameworks the possibility to consequently dissect and improve from trip excepting being expressly modified. Machine examining centers around the improvement of PC applications that can get admission to statistics and use it analyze for themselves. The technique of becoming acquainted starts with observation or data, for example, models, direct understanding or guidance.so as to appear the same as patterns and make higher choices for later based fully on the models that we give. The significant goal is to empower computer systems to examine naturally except human interference or help and take decisions accordingly. AI is the analysis of getting computers to function without any unambiguous alteration. AI has brought us self-driving cars in the previous decade, realistic speech recognition, competitive web search, and an increasingly developed understanding of the human genome. Today, AI is so inescapable that you will probably use it many times a day without realizing it. In fact, several experts agree it is the most suitable way to become more accepted in favor of AI at the human level. Machine Learning is one of the most enervating developments one's ever run across. It gives the Machine, as it is obvious from the name, which makes it more like people: the ability to learn. Today, AI is used widely, even in a lot more places than one would expect. When we seek to predict the target variable using any machine learning method, noise, uncertainty and bias are the key causes of variation in real and expected values. Ensemble helps in removing these variables. An ensemble is just a set of predictors that come together to give a good estimate. Ensembling procedures are additionally ordered into Bagging and Boosting. Bagging is a basic ensembling strategy in which we construct numerous free models and consolidate them utilizing some model averaging methods. We commonly take irregular sub-test of information for every model, with the goal that all the models are minimal and not the same as one another. Every observation is selected with replacement for each model to be used as data. So, each model would have different bootstrap dependent observations. Since several uncorrelated learners are needed to create a final model, this technique eliminates error by reducing variance. Random Forest models are an example of bagging ensembles. This methodology incorporates the logic in which the subsequent predictors learn from the previous predictors 'mistakes. The results therefore have an uneven likelihood of occurring in subsequent models. The factors may be taken from a number of models, such as classifiers, decision trees, regressors etc. Since new factors are learning from errors made by previous predictors, getting close to real predictions takes less time / iterations to achieve. We have to carefully pick the stop criteria or this might result in overfitting on training data. Any supervised learning algorithm has the purpose of defining and minimizing a loss function. The formula for mean squared error (MSE) which is defined as loss is given below:

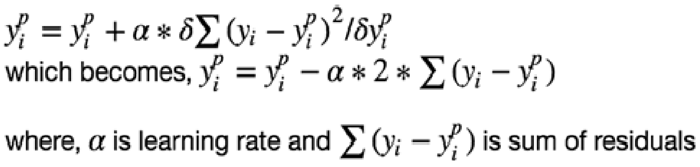


$$Loss = MSE = \sum (y_i - y_i^p)^2$$

where, y_i = ith target value, y_i^p = ith prediction, $L(y_i, y_i^p)$ is Loss function

Figure 1.1: Gradient Boosting Equation

We want our estimates, so the minimum is our loss function (MSE). We can find the values where MSE is minimum by using gradient descent and by updating our predictions based on a learning rate.



$$y_i^p = y_i^p + \alpha * \delta \sum (y_i - y_i^p)^2 / \delta y_i^p$$

which becomes, $y_i^p = y_i^p - \alpha * 2 * \sum (y_i - y_i^p)$

where, α is learning rate and $\sum (y_i - y_i^p)$ is sum of residuals

Figure 1.2: Gradient Boosting Equation-2

Therefore, we are essentially updating the forecasts in such a way that the sum of our residuals is close to 0 (or minimum) and expected values are near enough to actual values.

The idea behind the gradient boosting algorithm is to exploit the patterns in residuals and to enhance and develop a model with poor predictions and to improve it. When we enter a stage where residuals do not have any modeling pattern, we may avoid residual modeling. We reduce our loss function algorithmically, so that the test loss reaches its minimum.

1.3 XGBoost

XGBoost is an improved disseminated slope boosting library planned to be extraordinarily powerful, versatile and adaptable. It utilizes different AI calculations under the Gradient Boosting structure. XGBoost gives an equivalent tree boosting (in any case called GBDT, GBM) that deals with various data science issues in a brisk and cor- rect way. A similar code runs on big circulated condition and can deal with issues that have gone beyond billions of models. It runs on Windows, macOS, and Ubuntu(Linux). From the model definition, it aims to provide a "Scalable, Compact and Distributed Gradient Boosting Library," which runs on a solitary ma-chine, much like the Apache Hadoop, Apache Spark, and Apache Flink handling frameworks in circulation. In certain victorious classes in other machine learning rivalries it has increased a lot in prevalence and consideration as of late as the estimation of decision.

XGBoost initially started as Tianqi Chen's research project as a major component of the Distributed Machine Learning Community gathering. At first, it began as a terminal program that could be configured using a document on lib SVM configura- tion. After its use in the winning solution of the Higgs Machine Learning Challenge, it turned out to be notable in the ML rivalry hovers. After that, the bundles Python and R were developed, and now XGBoost has bundle use for Java, Scala, Julia, Perl and different languages. This took the library to more developers and introduced the Kaggle people category to its popularity, where it was used for a large number of rivalries.

It was introduced long ago with various packages making it easier to use in their individual societies. It has now been arranged for Python clients with the scikit-learn and for R clients with the caret package. It can also be organized in Data Flow frameworks such as Apache Spark, Apache Hadoop, and Apache Flink using the Rabit and XGBoost4J that are concerned. Additionally XGBoost is available for FPGAs on OpenCL. Tianqi Chen and Carlos Guestrin distributed reliable, adaptable use of XGBoost.

XGBoost's striking highlights make it different from other angle boosters.

CHAPTER 2

LITERATURE SURVEY

Heart Attack Prediction Using Machine Learning Methods is very useful and proved its importance in the past few years.XGBoost is an efficient implementation of gradient boosting techniques. Although there is no new mathematical break through. It is one of the well-built versions of Gradient boosting which used to optimal and to improve accuracy. It contains both a linear model and a tree learning algorithm. Boosting is a procedure that utilizes a lot of AI calculations to join frail learners to frame solid learners so as to increase the accuracy of the model. Boosting is a kind of ensemble learning. It comprises of Sequential learning (Boosting) and Parallel Learning (Bagging) eg: Random Forest. Ensemble learning is a strategy that is utilized to upgrade the exhibition of the AI model with improved proficiency and exactness. To create various weak learners and consolidate their predictions from one in number standard. Presently their weak learners are created by applying base Machine Learning algorithms on various distributions of the dataset. By and large, base AI calculations are decision trees. so what these base algorithms do is that they produce weak guidelines for every iteration. so after numerous cycles, the weak learners are consolidated and they structure solid learners that will anticipate the more exact result. There are three sorts of Boosting procedures. 1. Adaptive Boosting 2.Gradient Boosting 3.Extreme gradient boosting. Among this XG Boosting is a propelled rendition of gradient boosting strategy that is intended to concentrate on computational speed and model productivity. It really falls under the classification of dispersed machine learning community most advanced version of gradient boosting. Some of the advantages of XGBoost algorithm are it's highly flexible which means we can set custom evaluation criteria and optimization objectives. Processing is faster than gradient boosting.. It has built-in methods to handle missing data. Tree pruning: In Gradient boosting algorithm stops when it encounters -ve loss in the split but in case of XGBoost it digs to maximum depth and starts pruning the splits without any positive gain. It is a decision tree based algorithms which is considered best for small or medium structured date.Building models using XGBoost is quite easy.

But improving it's efficiency is really hard. We have various parameters in XGBoost, which requires tuning.

2.1 Review of Survey

[1] Manasa. K. N, Prince Kumar Gupta displayed a framework which is appropriate for real- time heart diseases forecast and can be utilized by the clients who have coronary disease.In this paper they propose an helped forecast framework which leverages insights in mining strategies to appear the relationship among the regular physical exam records and the capacity wellness threat it can anticipate examinees threat of physical notoriety subsequent year based completely on the physical examination data this year.The information is in tall measurement: Physical examination information as a rule incorporates examinees fundamental inarrangement, research facility comes about and a few demonstrative information. It implies unique information is tall dimensional and we ought to apply fitting dimensionality lessening strategy or otherwise we may experience tall computational complexity. Moreover, measurement input may moreover bring low precision issue when dataset isn't sufficient. [2] Nitten S. Rajliwall, Rachel Davey, Girija Chetty, proposed a structure which depends on supervised learning algorithms and preparing dependent on group level including category division dependent on sex ,level of education and age. "They propose a unified predictive modelling framework for"demonstrating system for tending to these difficulties. They proposed brought together structure permits the prescient AI models to be worked for various information assortment settings, including static or low speed settings, (EHR records from infrequent medical clinic affirmation records and ordinary clinical visits), and high speed ADL records (Activities of Daily Living) from wearable's and wellness trackers. Gradient boosting is the first model of XGBoost, consolidating feeble base learning models into a more stronger learning models in an iterative design. at every emphasis of slope boosting, the remaining will be utilized to address the past indicator that the predetermined misfortune capacity can be improved. As an improvement, regularization is added to the misfortune capacity to set up the objective capacity in XGBoost estimating the model execution. [3]SenthilKumar Mohan, ChandrasegarThirumalai, GautamSrivatsava proposed hybrid HRFLM approach 6 which is utilized in joining the features

of Random Forest (Random Forest (RF)) and Linear Method (Linear Method (Linear Method (LM))). They have increased the accuracy to 88.7% through the forecast model for coronary illness with the HRFLM. They have made neural networks utilizing pulse time series. This technique utilizes different clinical records for expectation, for example, "Left bundle branch block (LBBB), Right bundle branch block (RBBB), Atrial fibrillation (AFIB), Normal Sinus Rhythm (NSR), Sinus bradycardia (SBR), Atrial flutter (AFL), Premature Ventricular Contraction (PVC)), and Second degree block (BII) " to discover the specific state of the patient corresponding to coronary illness. The dataset with an outspread premise work organize (RBFN) is utilized for arrangement, where 70% of the information is utilized for preparing and the staying 30% is utilized for characterization.

[4] ShadmanNashif, Md. Rakib Raihan proposed a model which is a cloud dependent on coronary illness expectation model. This model is used to distinguish heart infections with the utilization of AI algorithms.In this examination, a model of a cloud-based coronary illness forecast framework had been proposed to recognize coronary illness utilizing ML algorithms. For the exact prediction of coronary illness, a proficient ML strategy ought to be utilized which had been taken from a particular investigation among a few ML algorithms in WEKA tool. Another significant component of the proposed framework was that when any ongoing parameter of the patient surpasses the edge, the recommended specialist can know through GSM technology. In addition, to observe the coronary illness all the time by his/her specialist, a continuous patient observing framework was created and introduced using Arduino, fit for detecting some real-time parameters, for example, "body temperature, blood pressure Blood Pressure (BP), humidity, heartbeat. The created framework can transmit the recorded information to a server which are updated every 10 seconds." [5] Susmitha Manikandan proposed a module of the framework comprise of binary classification model which is utilized to anticipate the risk factor of a patient dependent on their clinical data.In this investigation, a model of a framework that includes a binary classification model to determine the risk factor of an individual dependent on his/her clinical information is proposed. The framework is well outfitted with a GUI. "The classification follows a supervised learning wherein the dataset used was obtained from University of California (University of California (UCI)), Irvine's machine learning repository.."The ref-

erence is restricted to the Cleveland's dataset which was gathered as unstructured information as clinical reports and changed over to an organized dataset. The dataset speaks to a twofold characterization issue. The dataset contains 14 attributes altogether, out of which 13 are predictor attributes and one component is a "binary response variable"." [6] Aditi Gavhane, Gowtami Kokkula, Isha Pandya "proposed a framework where they utilized the Neaural Network (Neaural Network (NN)) Algorithm and multi layered perceptron for preparing and testing the"dataset". [7] D. K. Ravish, K.J. Shanthi, Nayana R Shenoy, S. Nisargh In this paper they have developed an efficient way to acquire the clinical and ElectrocardiographyElectrocardiography (ECG) data. For training the Artificial Neural Network to accurately diagnose and predict heart abnormalities if found any. [8] C. M Chethan- Malode, K. Bhargavi, B. G Gunasheela In this paper they have used fuzzy rule and set theory which is concatenated with Support Vector Machine (SVM) classifier to identify and differentiate heart attack risk among adolescents. [9]KwenbingChang,YinglaiLiu,XueyiWu,YiyongX- aio, Shenghan Zhou, Wen Cao. In this paper they have used XGBSVM which means XGboost plus SVM hybrid model to predict heart diseases within three years. [10]ProchetaNag,SaikatMondal,FoysalAhmed,ArunMore,M.Raihanused.Inthispaperthey have used classification techniques of data mining and decision tree to predict whether the chest pain is for heart attack or any other. [11]BoshraBahrami, Mirsaeid Hosseini Shirvani In this paper they have provided an intuition about data mining technique used to forecast cardiovascular diseases. Forest in 10-fold Cross-Validation which gave an accuracy of 80%. [5] Susmitha Manikandanused Naïve Bayes algorithm which gave an accuracy of 81.25. [12] C. M ChethanMalode, K. Bhargavi, B. G Gunasheela used settheory and Fuzzy theory enabled SVM Approach which gave an accuracy of 85.6

2.2 Inference from the Review

Table I - Inferences from the review

Title	**Journal**	**Techniques Used**	**Year**
Disease Prediction by Machine Learning with the help of Big Data from Healthcare Communities.	International Journal of Engineering Science And Computing (IEEE)	Logistic Regression, Naive Bayes, KNN Classifier.	2017
Cardiovascular Risk Prediction Using XGBoost.	Institute of Electrical and Electronics Engineers (IEEE)	XGBoost	2018
Effective Heart Disease Prediction Using Hybrid Machine Learning Techniques.	Institute of Electrical and Electronics Engineers (IEEE)	Hybrid Random Forest With Linear Method	2019
Heart Disease Detection by Machine Learning Algorithms and a Real-Time Cardiovascular Health Monitoring System	World Journal of Engineering and Technology	Logistic Regression, NaiveBayes, ANN, SVM, Random Forest	2018
Heart attack Prediction System	International Conference on Energy, Communication, Data Analytics and Soft Computing (IEEE)	Naïve Bayes, Decision Trees, K-Nearest Neighbour and Random Forest	2017

Continued on next page

continued..

Title	**Journal**	**Techniques Used**	**Year**
Prediction Of Heart Disease Using Machine Learning	2nd International conference on Electronics, Communication and Aerospace Technology	Neural Network algorithm Multi-Layer Perceptron (MLP)	2018
Heart Function Monitoring, Prediction and Prevention Of Heart Attacks Using ANN	Institute of Electrical and Electronics Engineers (IEEE)	Artificial Neural Networks(ANN)	2014
A New Hybrid XGBSVM Model:Application For Hypertensive Heart Disease	Institute of Electrical and Electronics Engineers (IEEE)	New XGBSVM hybrid model. This model is based on the original machine learning theory and is a new machine learning development.	2019
Softset and Fuzzy Rules Enabled SVM Approach For Heart Attack Risk Classification Among Adolescents	Fourth International Conference on Computing Communication Control and Automation (IEEE)	Softset theory and Fuzzy rules are combined with SVM	2018
A simple Acute Myocardial Infraction Prediction System Using Clinical Data And Data Mining Techniques	International Conference of Computer and Information Technology	SVM, Random Forest, KNN, ANN	2017

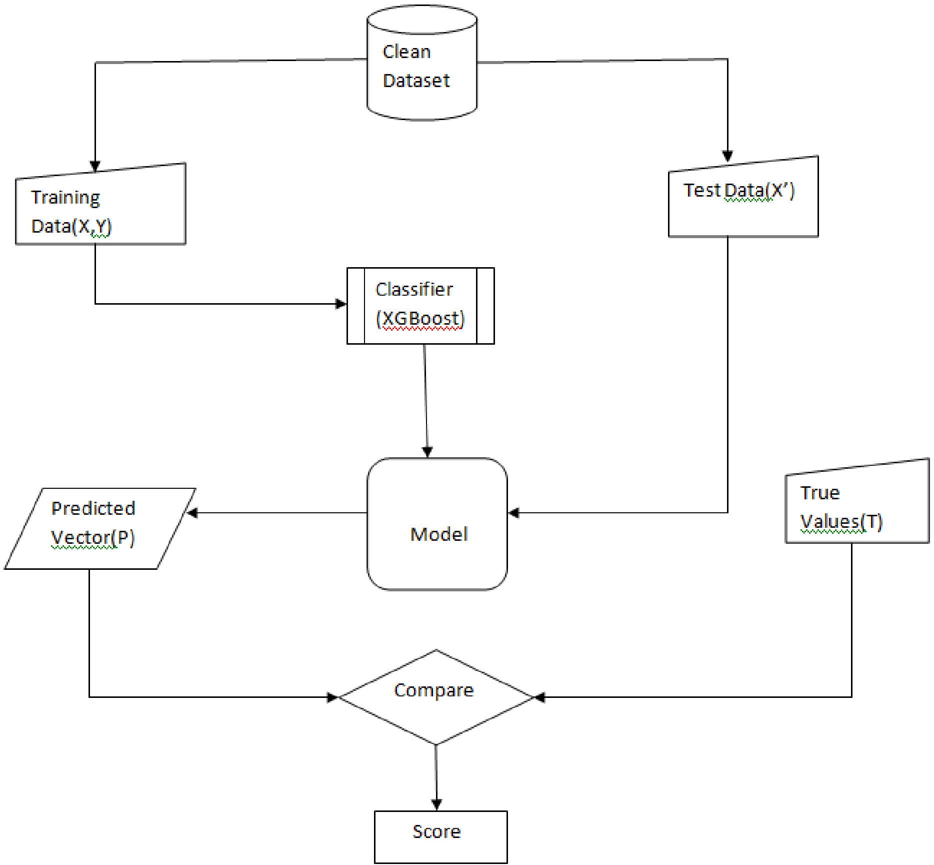
CHAPTER 3

PROPOSED METHODOLOGY

The literature review and survey paper give us some valuable insight regarding heart attack predication. Our proposed model can predict heart attack.The proposed system has 13 attributes that are being considered. The system is well structured with a thorough graphical UI that is anything but difficult to utilize and comprehend. The grouping follows a regulated learning wherein the dataset utilized was acquired from the University of California, Irvine's AI archive. The reference is restricted to Cleveland's dataset which was gathered as unstructured information as clinical reports and changed over to an organized dataset. We have utilized XGBoost calculation and at last, a simple to utilize web interface was grown so the framework can be utilized by people with zero specialized information, along these lines totally getting the center modules and using details of the system.

3.1 Project Architecture

Figure 3.1 describes the entire system architecture of this project.We have built a prototype model which has a binary classification to measure and to intimate the risk of causing heart attack based on the medical data of the individual data. After gathering many files the data is processed. Various patient files are present in the dataset. There are 303 files in total. 6 files in the total have some values missing. The files with missing values have been taken out. Now for the purpose of pre-processing the left 297 record are used. A variable is set for the parameters of the dataset. This variable is helpful to detect whether the person is more/less likely to get heart attack. If the patient is more likely to get heart attack, the variable is set to 1, otherwise it will be set to 0. The results show that 137 records out of 297 with value 1 indicating the occurrence of the heart attack and the rest 160 columns have 0 as value indicating less chance of heart attack.



X = 207 x 13 2D matrix

Y = 207 x 1 array

X' = 90 x 13 2Darray

P = 90 x 1 array

T = 90 x 1 array

Figure 3.1: Flow Chart Diagram of the Project

3.1.1 Dataset Description

The following parameters are available in the last numerical informational collection. The dataset is in .csv format. There are a sum of 14 parameters.The unstructured dataset are changed over to an organized dataset. The dataset has 14 traits from which 13 are indicator factors. One is a binary response variable. Extreme Gradient Boosting was utilized for the classification process.

1. Age: The age parameter taken in the dataset is in years.
2. Sex: Two genders are considered in the dataset. It it is male it is assigned 1 and for female it is 0.

3. Chest pain: There are 4 types of angina and they are given numbers to distinguish them. For typical angina it is 1, 2 for atypical angina, 3 for non-anginal pain and 4 for asymptomatic angina.

4. Test bps: Resting blood pressure (in mm HG) 120mm Hg is systolic, 80mm Hg diastolic and resting heart rate (60-100 BPM).

5. Cholesterol: Cholesterol is measured in mg/dl. Measuring High and Low density lipids in body fluids. High Density Lipids is good where as Low Density Lipids is bad.

6. FBS(Fasting Blood Sugar): Blood Sample taken after a patients fasts for at least 8 hours. For normal person it is less than 100 mg/dl.

7. Restecg: Resting 12-lead electrocardiography (ECG) is a non-invasive examination that can detect anomalies including arrhythmias, coronary heart disease proof, left ventricular hypertrophy, and bundle branch blocks. There are several levels in this. If the results are Normal then indicated as 0, 1 for ST-T wave abnormality, 2 for Left ventricular hypertrophy.

8. thalach: thalach means maximum heart rate obtained.

9. Exang: Exang means exercise induced angina. If it it is "yes" then indicated with 1 , 0 for "no".

10. old peak: old peak measures ST depression induced by exercise relative to rest.

11. slope: Slope is obtained from the graph. It is the peak ST segment. For up slopping it is set 0, 1 for falt and 2 for down slopping.

12. CA: When fluoroscope is done some major vessels are coloured. The number of major vessels coloured is indicated in the range of 0 t0 3.

13. Thal: This is Thallium - Stress Scintigraphy test. This test gives the ability of coronary arteries to deliver blood to the heart under stress. If it is normal indicated with 3, 6 for fixed defect and 7 for reversible defect.

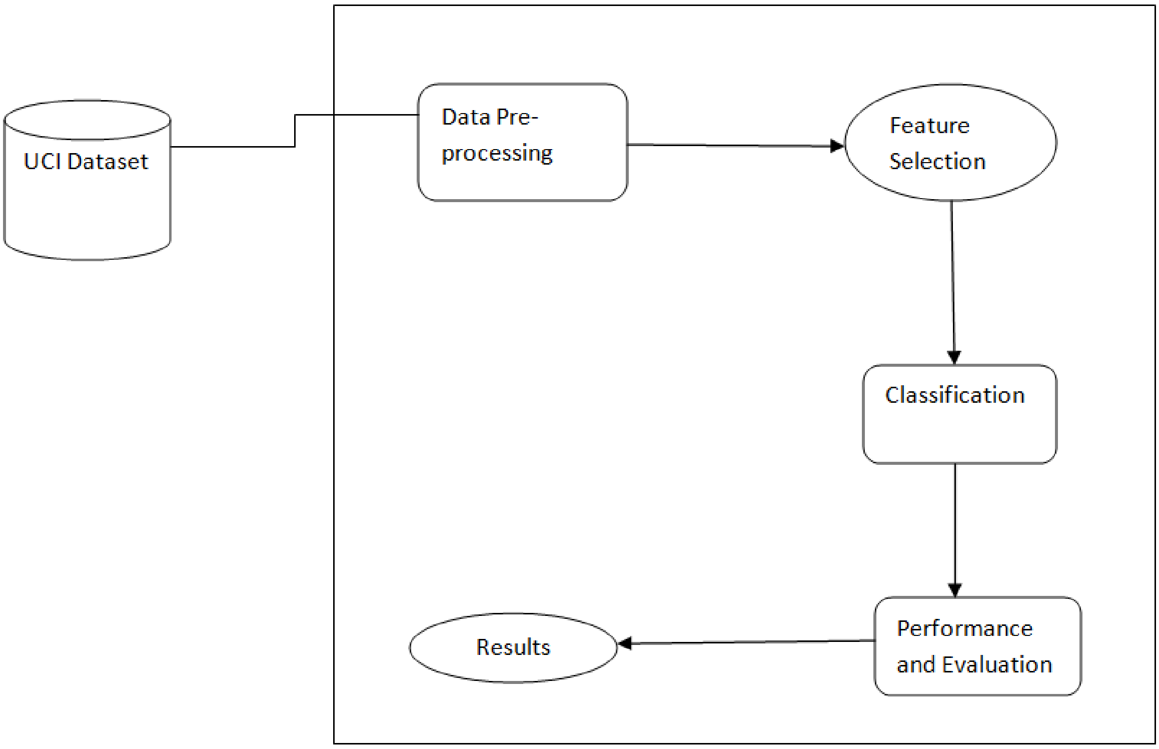


Figure 3.2: Architectural Diagram of the Project

3.1.2 Preprocessing and Feature Extraction

Information on heart disease is pre-treated with an array of various documents. The dataset includes an estimate of 303 records of patients, where 6 records have certain attributes lacking. Such 6 records were removed from the dataset and the remaining 297 patient records are used in pre-preparation. For the characteristics of the given data set the multiclass variable and binary classification are provided.

The multi-class variable is used to test whether coronary disease occurs or does not occur. If a patient has a coronary disease, the variable is set to 1, then the variable is set to 0 which indicates the patient's non-appearance of coronary disease. Preparation of information is achieved by shifting to diagnostic principles over clinical records. Results of pre-treatment information for 297 patient records indicate that 137 reports show the estimate of 1 indicating the closeness of coronary disease while the remaining 160 mirrored the estimate of 0 suggesting coronary disease non-attendance.

Two variables relating to age and sex are used from among the 13 information gathering characteristics to identify the patient's individual details. The remainder of the 11 variables are regarded as important as they comprise critical clinical information. Clinical records are important for assessing and understanding the nature of the coronary disease. Basic Linear Regression, Naive Bayes, Decision Tree, K-Nearest Neighbors (K-Nearest Neighbors (KNN)), Multinomial Naive Bayes, Support Vector Machine (SVM), XG- Boost are used as recently stated (Machine Learning (ML)) techniques. The exam was rehashed for all ML methods using each of the 13 properties.

3.1.3 Models

Accuracy Calculation

1. Recall: If the value of the Recall is high then the class is correctly recognized.
2. Precision: If the output of the testing example is positive and it is positive actually, then the precision is high.
3. High Recall, Low Precision: There are more false positives even though all the true positives are correctly recognized.
4. Low recall, High precision: Some of the true positives are correctly recognized while some of the positives are not recognized as positives.
5. F-Measure: UUses both Recall and Precision. Harmonic Mean is used in F-Measure.The value we get in F-Measure will always be close to the smaller value of Recall or Precision.
6. Numpy:Numpy is a python programming language library to add support for large, multi-dimensional arrays and matrices, along with a wide set of high-level mathematical functions for such arrays to run.
7. Pandas: Python is a great programming language for data processing, making it much easier to import, interpret and visualize data
8. Sklearn: Sci kit-learn is a python library that provides various unsupervised and supervised learning algorithms. The metrics module defines functions for different purposes which evaluate prediction error.
9. Confusion Matrix:

	Predicted Negatives	Predicted Positives
Actual Positives	34(TN)	7(FP)
Actual Negatives	2(FN)	48(TP})

10. Accuracy (or) Classification Rate:

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

3.1.4 Evaluation Process

Two parameters out of 13, are utilized to recognize the patient's data. The 11 parameters which are remaining are important. These 11 parameters are important for identification and knowing the condition of the heart. As forerly stated in the experiment, many machine learning techniques are used namely Linear Regression, Naive Bayes, Decision Tree, KNN, Multinomial Naive Bayes, SVM,XGBoost. The experiment was redone using many machine learning techniques with same attributes. Now various machine learning methods can be applied as our dataset is ready. Classification and Modelling is the important phase of the system, where the result of classification is obtained. Various algorithms are selected and their performance is compared. Out of all those algorithms XGBoost gives us the result with high accuracy. In user interface User Interface (UI) we will be having a web application. In this application the person can enter his/her details and check for their risk of heart attack. With the data entered by the person his risk rate will be shown on the screen. If it is showing 'High Risk Individual' then the person is highly prone to heart attack and can consult his/her physician. If it shows 'Low Risk Individual' then the person is less prone to get heart attack. To make this web application 'flask' is used.

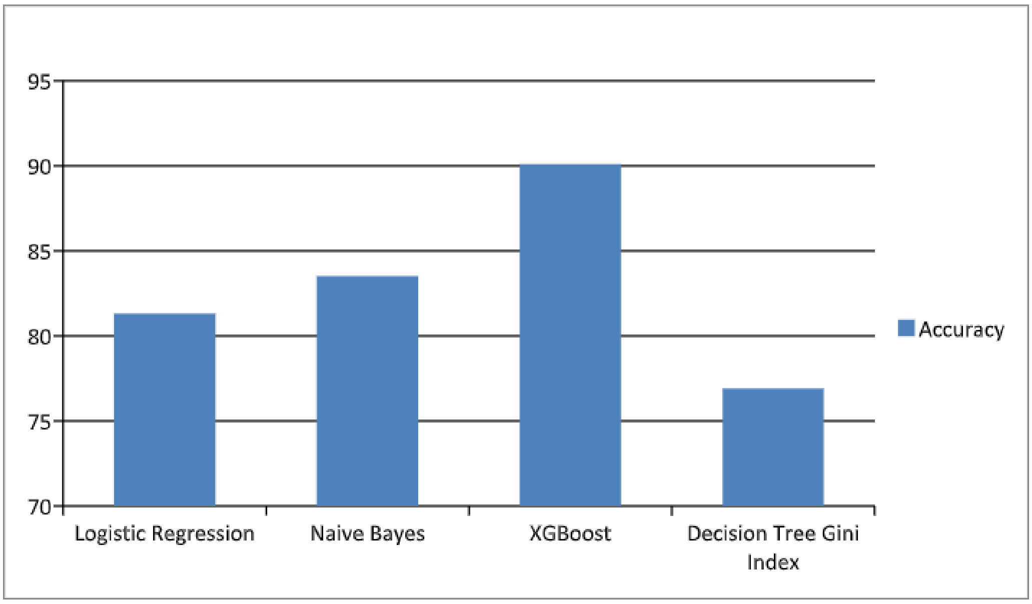


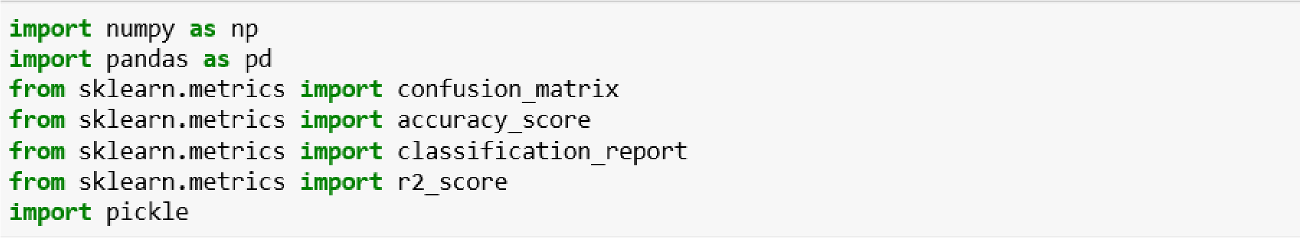
Figure 3.3: Comparison Graph

CHAPTER 4

IMPLEMENTATION

After gathering many files the data is processed. Various patient files are present in the dataset. There are 303 files in total. 6 files in the total have some values missing. The files with missing values have been taken out. Now for the purpose of pre-processing the left 297 record are used. A variable is set for the parameters of the dataset. This variable is helpful to detect whether the person is more/less likely to get heart attack. If the person is more likely to get heart attack, the variable is set to 1, otherwise it is set to 0. The results show that 137 records out of 297 with value 1 indicating the occurrence of the heart attack and the rest 160 columns have 0 as value indicating less chance of heart attack.

The proposed system aims to solve the issue by predicting that tells the risk of heart attack. We have used dataset from UCI's Machine learning repository. Among various Algorithms extreme gradient boosting algorithm yield us the better result. Chest pain is one of the most remarkable symptoms of heart attack. Predicting heart attack is one of the most important aspect where if there is any delay in detecting it may lead to damage to heart muscle. Myocardial dead tissue happens when there is blockage in coronary conduit that provisions rich oxygenated blood to heart



```
import numpy as np
import pandas as pd
from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score
from sklearn.metrics import classification_report
from sklearn.metrics import r2_score
import pickle
```

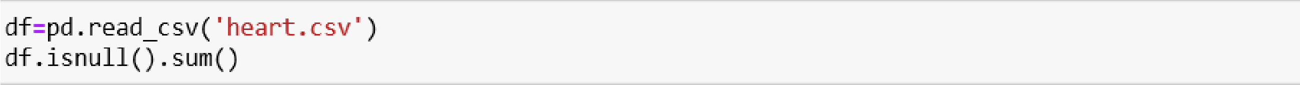
Figure 4.1: Importing Packages

In the above figure we are importing packages. Packages that includes Numpy, Pandas,SKlearn. We are importing Numpy for numerical Calculations and Pandas package for dataset Reading. The reason we are importing the SKlearn package is that it used to

Machine Learning and Deep Learning. Support Vector Machine modules are present in SKLearn. Also we will be importing Confusion Matrix , Accuracy Score , Classification Report , R2 Score. These modules are required in order to find the accuracy of the given model. We have used Pickle for converting object into byte stream. Sklearn has pre-processing techniques, Metix, Decision Tree, Naive Bayes.

4.1 Data Processing

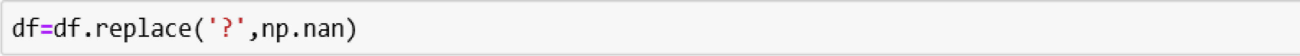
Data on heart disease is pre-treated after an array of various documents. The dataset includes an estimate of 303 records of patients, where 6 records have certain attributes lacking. These 6 records were removed from the registry, and the remaining 297 records of patients are used in pretreatment. For the qualities of the given data set the multi-class variable and binary classification are provided. The multi- variable is used to test whether coronary illness occurs or does not occur. For the case of a coronary disease patient, the variable is set to 1, then the variable is set to 0 indicating the patient's non- of coronary disease. Preparation of information is achieved by shifting to diagnostic principles over clinical records. The findings of pre-preparing details for 297 patient records indicate that 137 reports show an estimate of 1 indicating the closeness of coronary disease while the remaining 160 showed an estimate of 0 demonstrating the non-appearance of coronary disease.



```
df=pd.read_csv('heart.csv')
df.isnull().sum()
```

Figure 4.2: Reading the Data

In the above figure we are reading the dataset where we have cloned it from UCI ML Repo. The name of the dataset file is Heart.csv In the second step where we have df.isnull().sum() is for finding the different datatypes that are in the file. ? and string data types are replaced with NULL Value.



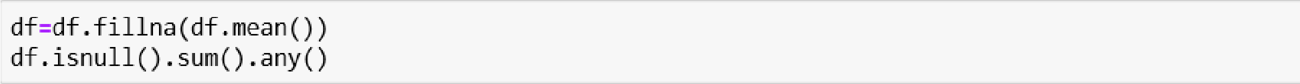
```
df=df.replace('?',np.nan)
```

Figure 4.3: Reading the Dataset

We are replacing the unsupported format and replacing it will NULL so that the algorithm runs efficiently.

4.2 Feature Selection and Reduction

From among the 13 properties of the informational collection, two factors relating to age and sex are utilized to recognize the individual data of the patient. The rest of 11 qualities are viewed as significant as they contain essential clinical records. Clinical records are crucial to analysis and learning the seriousness of coronary illness. As recently referenced right now, (ML) methods are utilized to be specific Linear Regression, Naive Bayes, Decision Tree, KNN, Multinomial Naive Bayes, SVM ,XGBoost. The examination was rehashed with all the ML strategies utilizing each of the 13 characteristics.



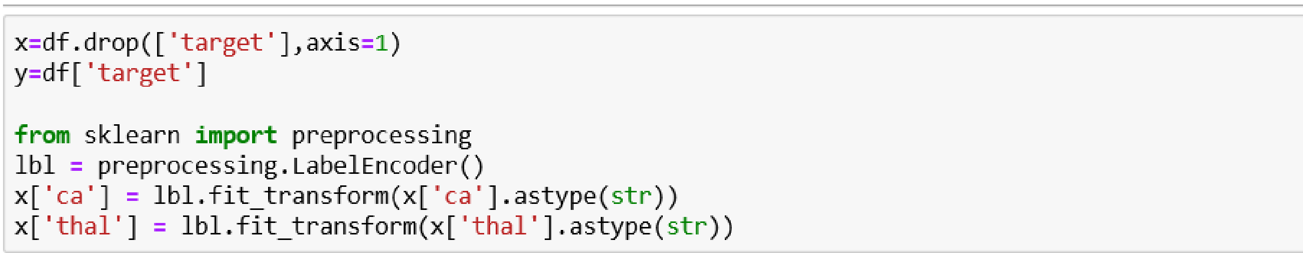
```
df=df.fillna(df.mean())
df.isnull().sum().any()
```

Figure 4.4: Replacing Nan

Here we are checking if NULL values are greater than 50 then we will be removing the whole column. Also if the NULL Values are less than 50 we are replacing it with Mean or Mode. That is the use of df.fillna. Here we are replacing it with Mean.

4.3 Classification and Modelling

Presently we have arranged the dataset and prepared for applying AI strategies. This is the center of the entire framework, which gives the characterization result from the highlight vector of the examinee. We select a few distinct calculations and look at their exhibition in our investigations. XGBoost presents an equal tree boosting (also referred to as GBDT, GBM) that solves numerous data technological ability to how to solve troubles in a quick and exact manner.

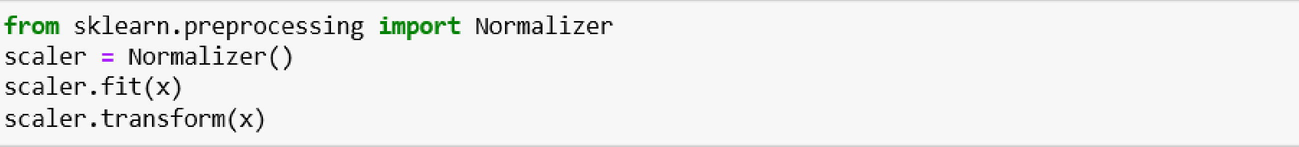


```
x=df.drop(['target'],axis=1)
y=df['target']

from sklearn import preprocessing
lbl = preprocessing.LabelEncoder()
x['ca'] = lbl.fit_transform(x['ca'].astype(str))
x['thal'] = lbl.fit_transform(x['thal'].astype(str))
```

Figure 4.5: Finding Mean

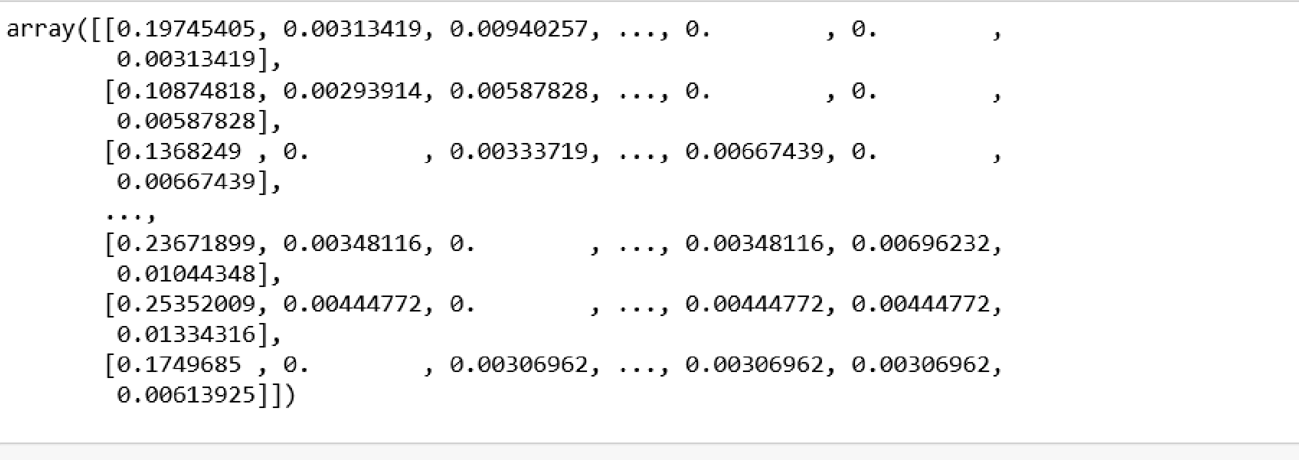
XG Boost Should have all the variables in the format of Int , Float. CA and Thal column from the data set are in the type of object so hence they need to be int and float were converting from Object to Int or Float. So here we are pre processing the data.

here as we can see the x and y variables. That means X variable contains all the columns that is 13 columns from the data set and the rest one column which is target column is Y. Where it defines the result whether the person gets heart attack or not. It is in the format of 0 and 1. Where 0 means he is not favorable to heart attack. While 1 means he is favorable to heart attack. In fit transform first we train and then normalization will be done.Also the reason we are using normalization because when the values are in between 0 and 1 the machine learning algorithm provides more accurate result.In the above figure we have taken 70% of X,Y as training and the rest 30% for testing.



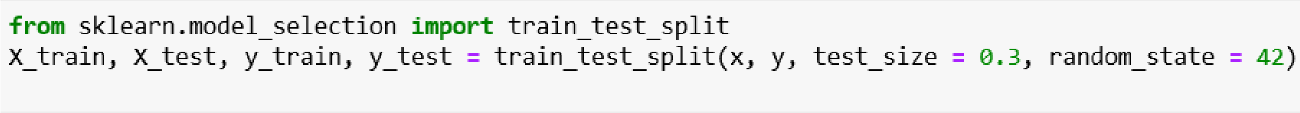
```
from sklearn.preprocessing import Normalizer
scaler = Normalizer()
scaler.fit(x)
scaler.transform(x)
```

Figure 4.6: Normalize



```
array([[0.19745405, 0.00313419, 0.00940257, ..., 0.        , 0.        ,
        0.00313419],
       [0.10874818, 0.00293914, 0.00587828, ..., 0.        , 0.        ,
        0.00587828],
       [0.1368249 , 0.        , 0.00333719, ..., 0.00667439, 0.        ,
        0.00667439],
       ...,
       [0.23671899, 0.00348116, 0.        , ..., 0.00348116, 0.00696232,
        0.01044348],
       [0.25352009, 0.00444772, 0.        , ..., 0.00444772, 0.00444772,
        0.01334316],
       [0.1749685 , 0.        , 0.00306962, ..., 0.00306962, 0.00306962,
        0.00613925]])
```

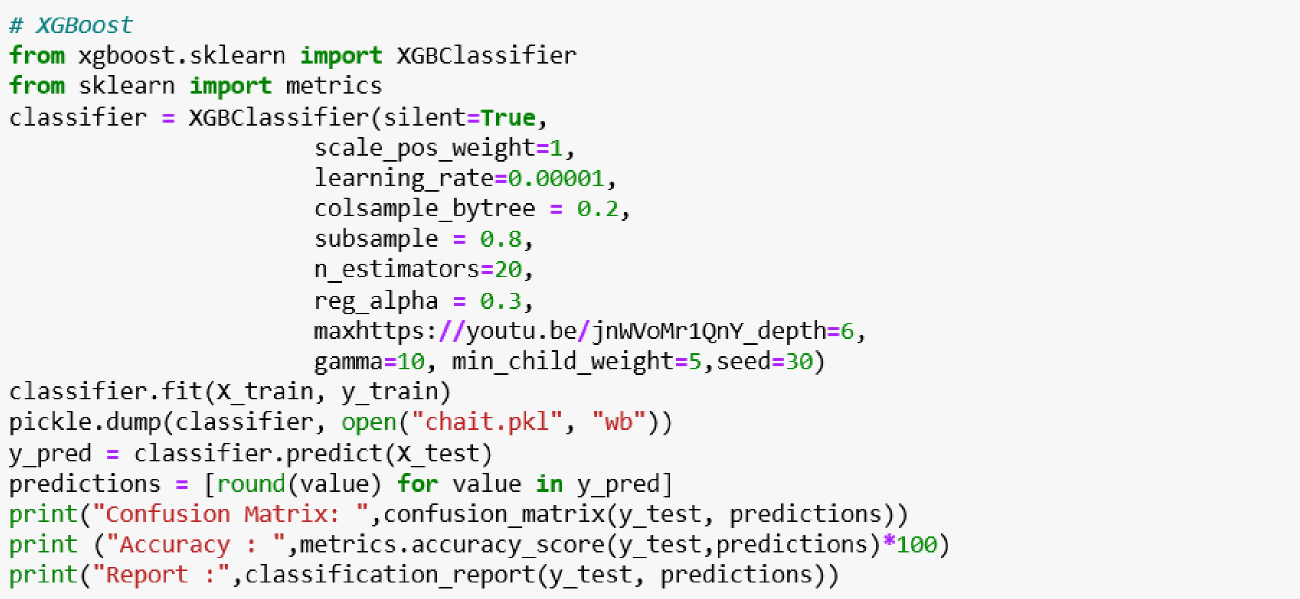
Figure 4.7: Matrix



```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(x, y, test_size = 0.3, random_state = 42)
```

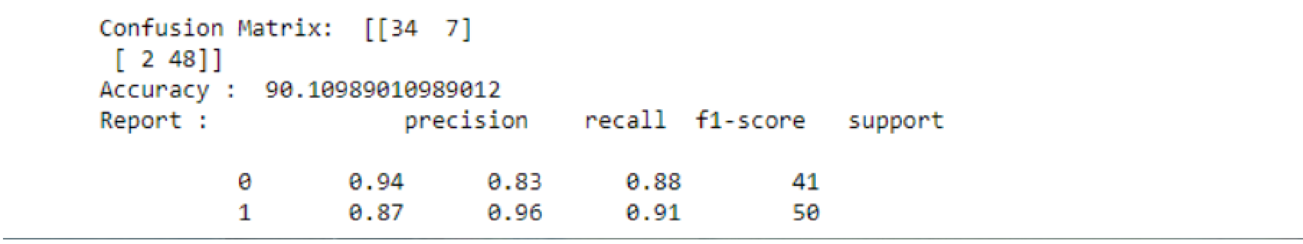
Figure 4.8: Split Data

The Data is trained and 70% of them are trained and 30% testing. The random spitted data is taken and it is tested. We are taking in random manner so that it doesn't train based on order.



```
# XGBoost
from xgboost.sklearn import XGBClassifier
from sklearn import metrics
classifier = XGBClassifier(silent=True,
                      scale_pos_weight=1,
                      learning_rate=0.00001,
                      colsample_bytree = 0.2,
                      subsample = 0.8,
                      n_estimators=20,
                      reg_alpha = 0.3,
                      maxhttps://youtu.be/jnWVoMr1QnY_depth=6,
                      gamma=10, min_child_weight=5,seed=30)
classifier.fit(X_train, y_train)
pickle.dump(classifier, open("chait.pkl", "wb"))
y_pred = classifier.predict(X_test)
predictions = [round(value) for value in y_pred]
print("Confusion Matrix: ",confusion_matrix(y_test, predictions))
print ("Accuracy : ",metrics.accuracy_score(y_test,predictions)*100)
print("Report :",classification_report(y_test, predictions))
```

Figure 4.9: XGBoost



```
Confusion Matrix:  [[34  7]
 [ 2 48]]
Accuracy :  90.10989010989012
Report :               precision    recall  f1-score   support

          0       0.94      0.83      0.88        41
          1       0.87      0.96      0.91        50
```

Figure 4.10: Accuracy

After finding the confusion matrix we get the accuracy. Accuracy is calculated by precision , recall , fl-score , support. TP is something like the data that is given is an heart attack patient and the result of prediction is high risk individual. TN Means the data that is given of heart attack patient is true and the result that we got is negative.

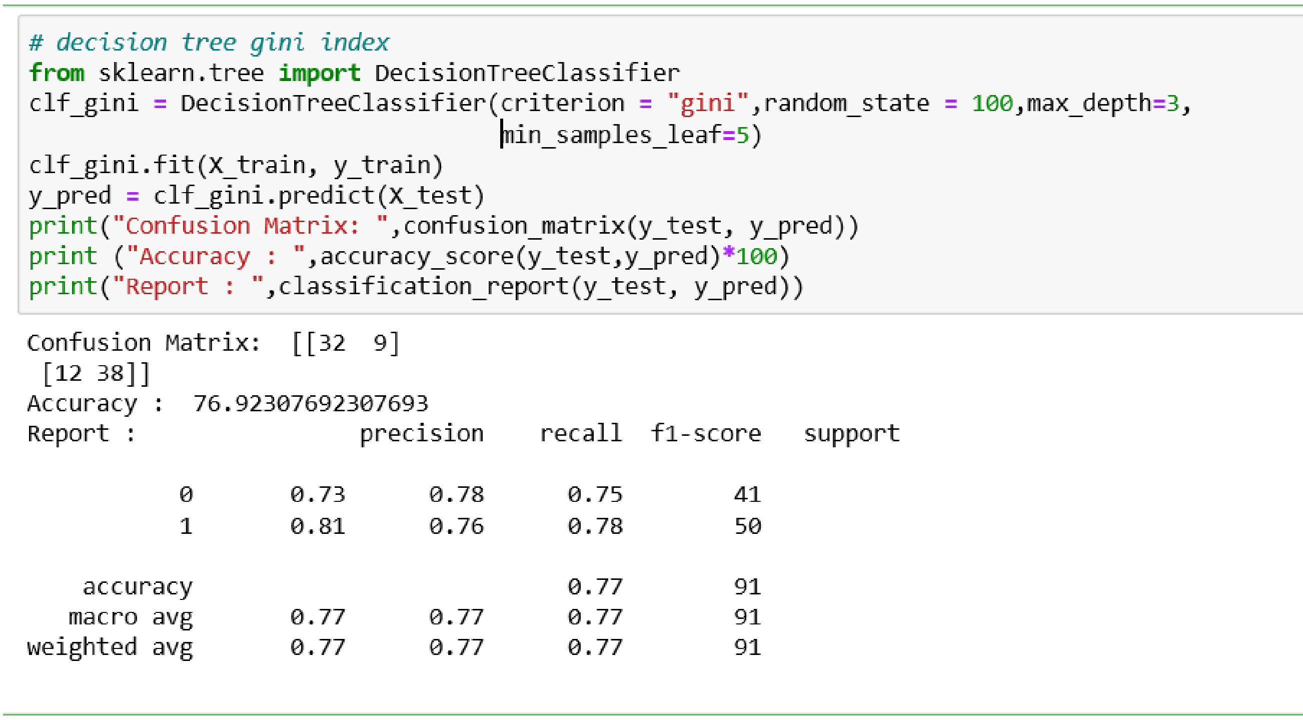
4.4 Algorithm

1. Collecting many files(files are taken from UCI's machine leaning repository)
2. After collecting the files data is processed(smoothing the outlier and missing data).
3. Adding a variable to the dataset(i.e. 1 for indicating more chance of heart attack and 0 for indicating less chance of heart attack)
4. Dividing the dataset into two parts. One part is testing dataset and the other part is training datase
5. The required parameters are selected for applying various machine learning algorithms.
6. Applying various machine learning algorithms to these parameters.
7. Finding the algorithm that gives the highest accuracy.
8. User Interface is created with the best algorithm and the person can know the risk of getting heart attack by entering the values.

CHAPTER 5

COMPARISON OF DIFFERENT ALGORITHM

5.1 Decision Tree Gini Index



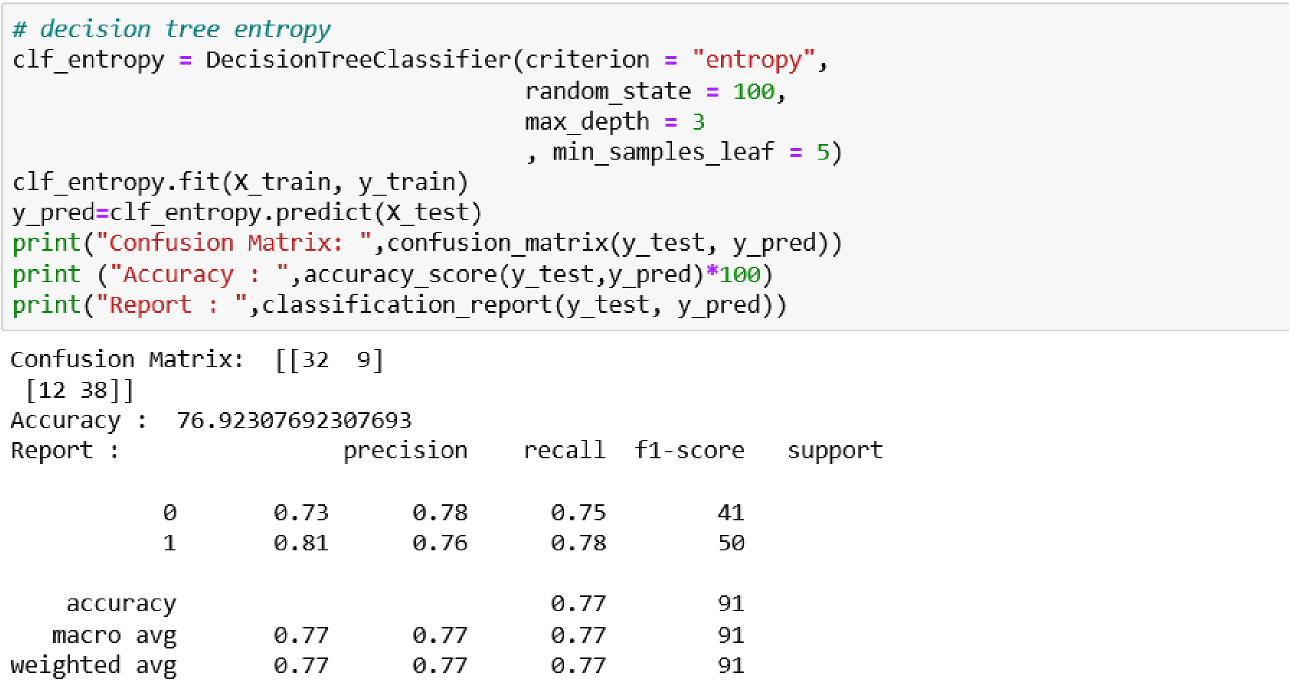
```
# decision tree gini index
from sklearn.tree import DecisionTreeClassifier
clf_gini = DecisionTreeClassifier(criterion = "gini",random_state = 100,max_depth=3,
                                  min_samples_leaf=5)
clf_gini.fit(X_train, y_train)
y_pred = clf_gini.predict(X_test)
print("Confusion Matrix: ",confusion_matrix(y_test, y_pred))
print ("Accuracy : ",accuracy_score(y_test,y_pred)*100)
print("Report : ",classification_report(y_test, y_pred))
```

```
Confusion Matrix:  [[32  9]
 [12 38]]
Accuracy :  76.92307692307693
Report :                precision    recall  f1-score   support

          0       0.73      0.78      0.75        41
          1       0.81      0.76      0.78        50

    accuracy                           0.77        91
   macro avg       0.77      0.77      0.77        91
weighted avg       0.77      0.77      0.77        91
```

Figure 5.1: Decision Tree Gini Index

The Decision Gini Index is calculated by subtracting from one the aggregates of each class ’squared probabilities. Bigger segments are preferred. Data Benefit increases the probability of class times the likelihood of the log (base=2) of that class. Data Benefit favors littler allotments with several unique characteristics. Eventually, you need to explore different avenues regarding your information and the parting measure.

5.2 Decision Tree Entropy



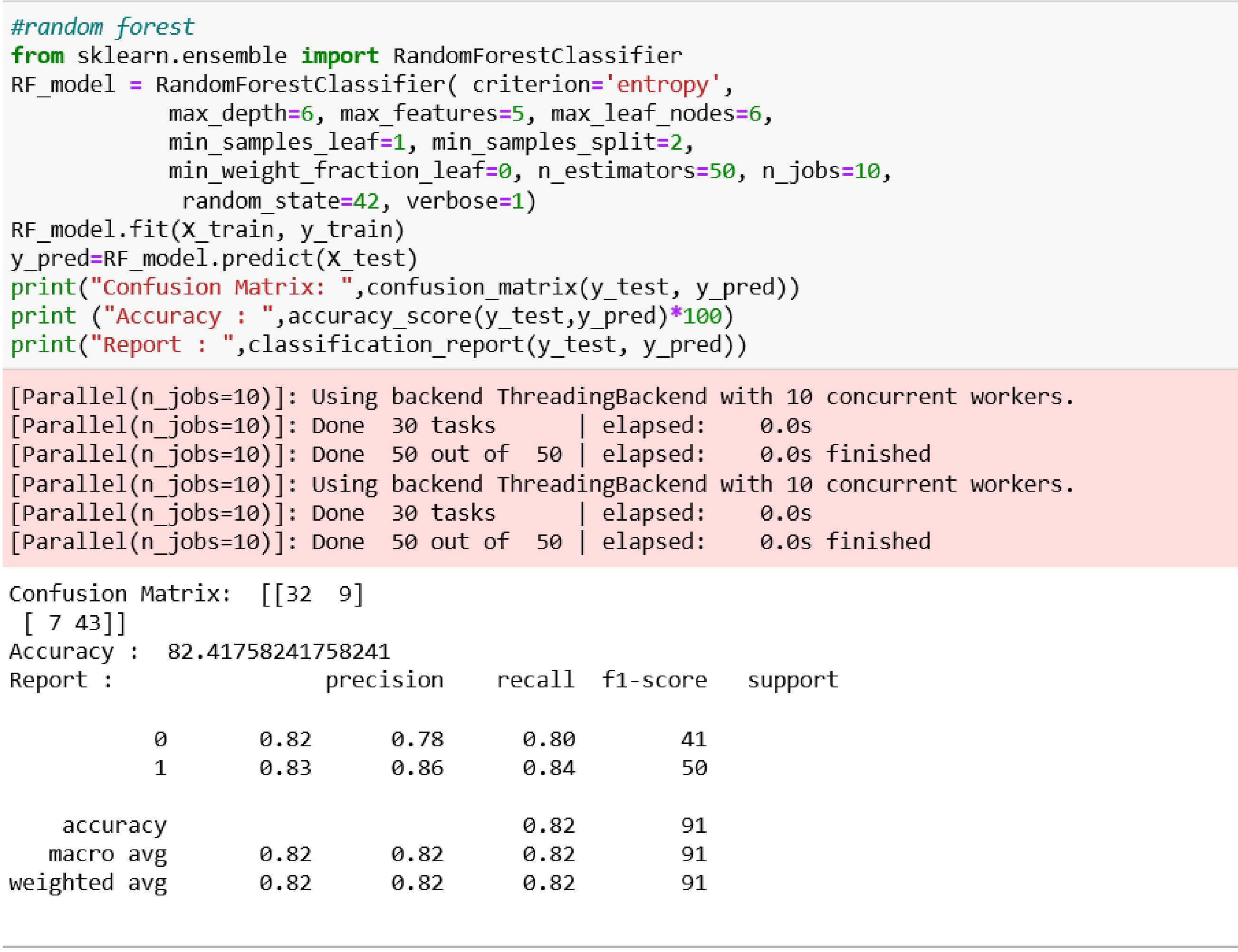
```
# decision tree entropy
clf_entropy = DecisionTreeClassifier(criterion = "entropy",
                                     random_state = 100,
                                     max_depth = 3
                                     , min_samples_leaf = 5)
clf_entropy.fit(X_train, y_train)
y_pred=clf_entropy.predict(X_test)
print("Confusion Matrix: ",confusion_matrix(y_test, y_pred))
print ("Accuracy : ",accuracy_score(y_test,y_pred)*100)
print("Report : ",classification_report(y_test, y_pred))
```

```
Confusion Matrix:  [[32  9]
 [12 38]]
Accuracy :  76.92307692307693
Report :                precision    recall  f1-score   support

           0       0.73      0.78      0.75        41
           1       0.81      0.76      0.78        50

    accuracy                           0.77        91
   macro avg       0.77      0.77      0.77        91
weighted avg       0.77      0.77      0.77        91
```

Figure 5.2: Decision Tree Entropy

Where ’Pi’ is just the frequent list likelihood of a component/class ’I’ in our information. For the good of simplicity suppose we just have two classes , a positive class and a negative class. Thusly ’I’ here could be either + or (-). So on the off chance that we had a sum of 100 information focuses in our dataset with 40 holding the +ve class and 60 holding the -ve class then ’P+’ would be 4/10 and ’P-’ would be 6/10. Entirely direct. Decision Tree Entropy has yield us the the accuracy of 76.9%

5.3 Random Forest



```
#random forest
from sklearn.ensemble import RandomForestClassifier
RF_model = RandomForestClassifier( criterion='entropy',
            max_depth=6, max_features=5, max_leaf_nodes=6,
            min_samples_leaf=1, min_samples_split=2,
            min_weight_fraction_leaf=0, n_estimators=50, n_jobs=10,
             random_state=42, verbose=1)
RF_model.fit(X_train, y_train)
y_pred=RF_model.predict(X_test)
print("Confusion Matrix: ",confusion_matrix(y_test, y_pred))
print ("Accuracy : ",accuracy_score(y_test,y_pred)*100)
print("Report : ",classification_report(y_test, y_pred))
```

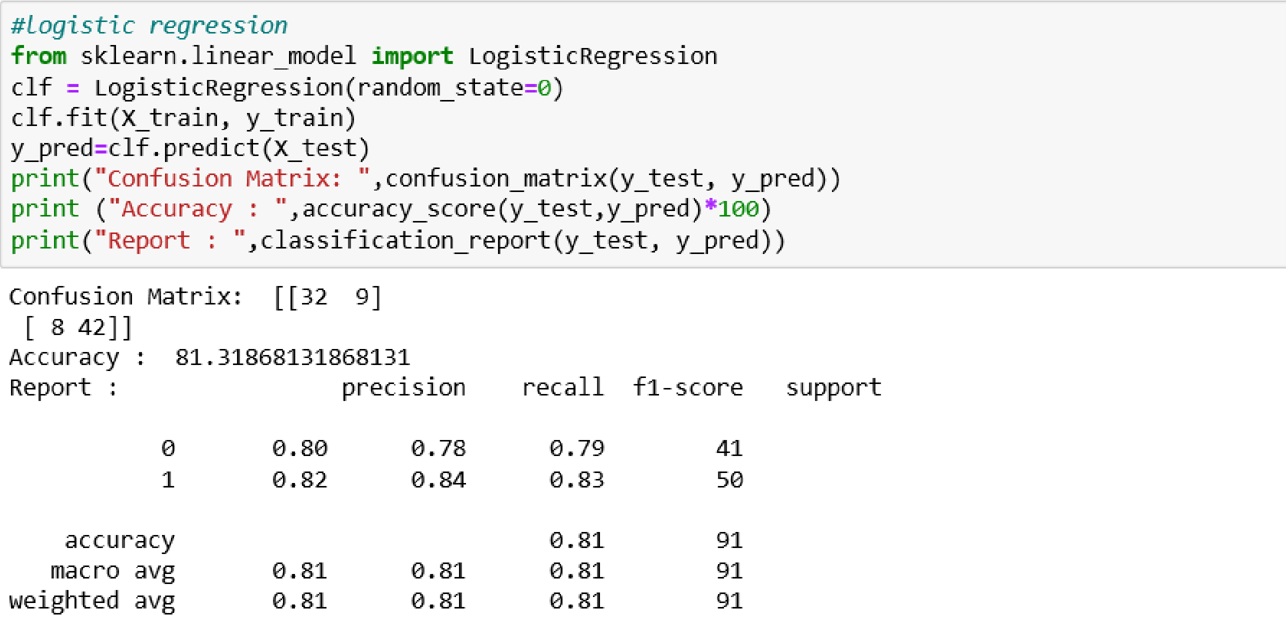
```
[Parallel(n_jobs=10)]: Using backend ThreadingBackend with 10 concurrent workers.
[Parallel(n_jobs=10)]: Done  30 tasks      | elapsed:    0.0s
[Parallel(n_jobs=10)]: Done  50 out of  50 | elapsed:    0.0s finished
[Parallel(n_jobs=10)]: Using backend ThreadingBackend with 10 concurrent workers.
[Parallel(n_jobs=10)]: Done  30 tasks      | elapsed:    0.0s
[Parallel(n_jobs=10)]: Done  50 out of  50 | elapsed:    0.0s finished
```

```
Confusion Matrix:  [[32  9]
 [ 7 43]]
Accuracy :  82.41758241758241
Report :               precision    recall  f1-score   support

           0       0.82      0.78      0.80        41
           1       0.83      0.86      0.84        50

    accuracy                           0.82        91
   macro avg       0.82      0.82      0.82        91
weighted avg       0.82      0.82      0.82        91
```

Figure 5.3: Random Forest

Random Forest algorithm is one among the supervised learning models. By name, we can tell the form of Random Forest. Which means it creates models in some way and makes it random. Also, there is a relationship between the number of models and the outcome of the model. The larger the modules the more accurate results you get. but we should make sure that having a large number of models is not the same as compared to the gain index.

5.4 Logistic Regression



```
#logistic regression
from sklearn.linear_model import LogisticRegression
clf = LogisticRegression(random_state=0)
clf.fit(X_train, y_train)
y_pred=clf.predict(X_test)
print("Confusion Matrix: ",confusion_matrix(y_test, y_pred))
print ("Accuracy : ",accuracy_score(y_test,y_pred)*100)
print("Report : ",classification_report(y_test, y_pred))
```

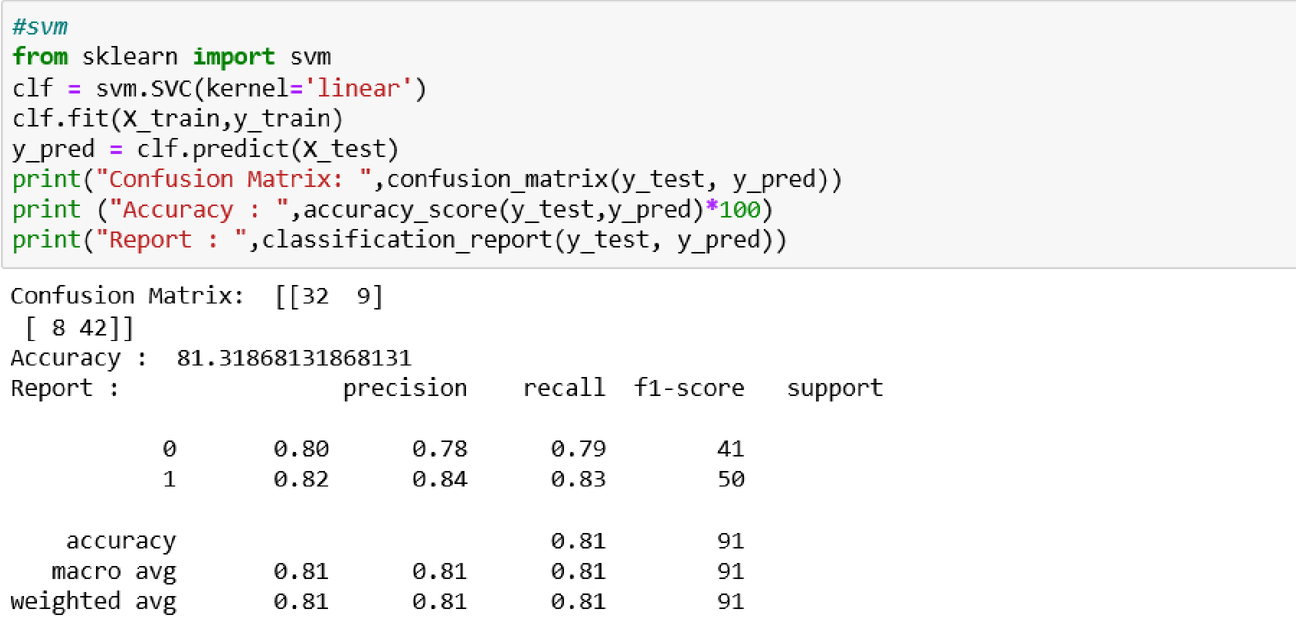
```
Confusion Matrix:  [[32  9]
 [ 8 42]]
Accuracy :  81.31868131868131
Report :                precision    recall  f1-score   support

           0       0.80      0.78      0.79        41
           1       0.82      0.84      0.83        50

    accuracy                           0.81        91
   macro avg       0.81      0.81      0.81        91
weighted avg       0.81      0.81      0.81        91
```

Figure 5.4: Logistic Regression

The strategic model is used in calculations to show the probability of an actual particular class or event, such as pass / come up short, win / lose, alive / dead or sound / wiped out. This can be extended out to include any occasional lessons. Decide, for example, if a image includes a feline, hound, lion and so on. Each object found in the picture will be relegated to a probability somewhere between the range of 0 and 1 and the entire addition to one.

Strategic relapse is a factual model that uses a measured capacity to demonstrate a double-reliant element in its critical structure, but there are a number of increasingly complex increases. For relapse analysis the parameters of a determined model are evaluated by strategic regression. Scientifically, a measured paired model has a dependable component with two possible qualities. For eg, pass / bomb to which a pointer variable addresses, where "0" and "1" are marked with the two qualities. The log-chances for the value called "1" in the measured model are a direct blend of at least one autonomous factors (indicators). Free factors may be either a parallel or a permanent variable.

5.5 Support Vector Machine



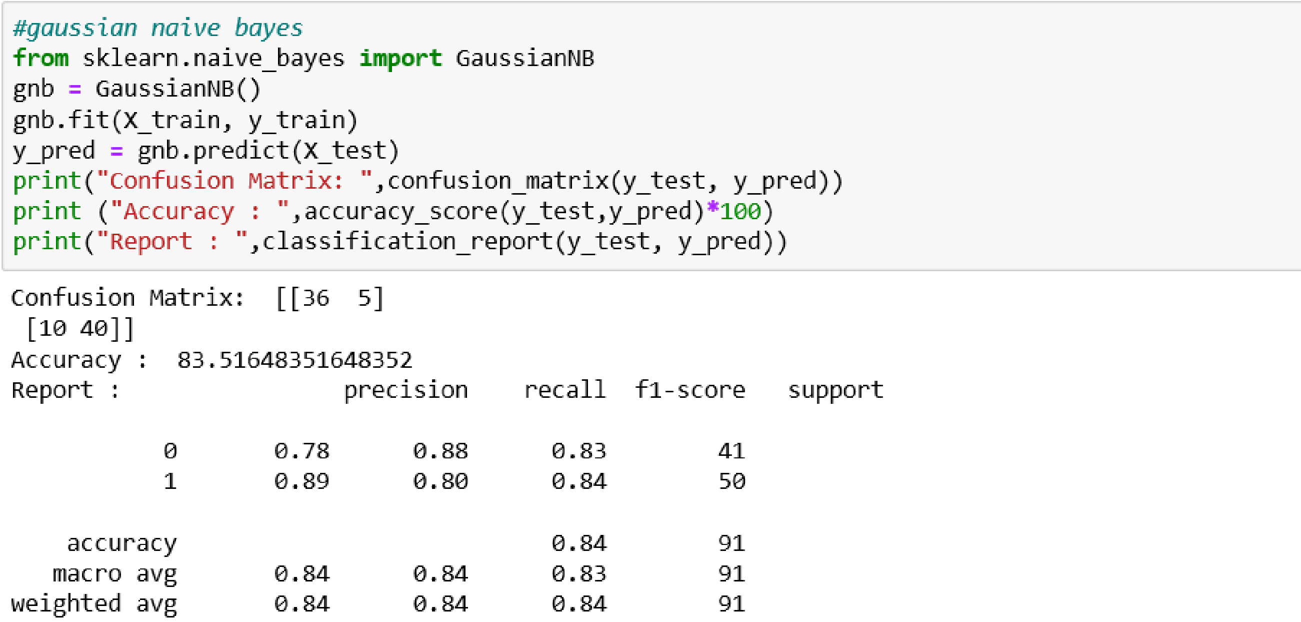
```
#svm
from sklearn import svm
clf = svm.SVC(kernel='linear')
clf.fit(X_train,y_train)
y_pred = clf.predict(X_test)
print("Confusion Matrix: ",confusion_matrix(y_test, y_pred))
print ("Accuracy : ",accuracy_score(y_test,y_pred)*100)
print("Report : ",classification_report(y_test, y_pred))
```

```
Confusion Matrix:  [[32  9]
 [ 8 42]]
Accuracy :  81.31868131868131
Report :                precision    recall  f1-score   support

           0       0.80      0.78      0.79        41
           1       0.82      0.84      0.83        50

    accuracy                           0.81        91
   macro avg       0.81      0.81      0.81        91
weighted avg       0.81      0.81      0.81        91
```

Figure 5.5: Support Vector Machine

In AI, support-vector machines are directed learning models with related learning calculations that dissect information utilized for arrangement and relapse examination. In AI, support-vector machines are driven learning models with related learning calculations which dissect information used for arrangement and relapse review. Provided a lot of preparing models each set apart as having a position with one of two classes, a calculation preparing SVM builds a model that doles out new guides to one or the other classification, making it a non-probabilistic straight classifier. Platt scaling, for example, exists for the use of SVM in a probabilistic setting. A SVM model is a representation of the models as space focuses, mapped with the intention of partitioning the instances of the different groups by an unmistakable hole that is as large as would be required under the situation.

5.6 Gaussian Naive Bayes



```
#gaussian naive bayes
from sklearn.naive_bayes import GaussianNB
gnb = GaussianNB()
gnb.fit(X_train, y_train)
y_pred = gnb.predict(X_test)
print("Confusion Matrix: ",confusion_matrix(y_test, y_pred))
print ("Accuracy : ",accuracy_score(y_test,y_pred)*100)
print("Report : ",classification_report(y_test, y_pred))
```

```
Confusion Matrix:  [[36  5]
 [10 40]]
Accuracy :  83.51648351648352
Report :                precision    recall  f1-score   support

           0       0.78      0.88      0.83        41
           1       0.89      0.80      0.84        50

    accuracy                           0.84        91
   macro avg       0.84      0.84      0.83        91
weighted avg       0.84      0.84      0.84        91
```

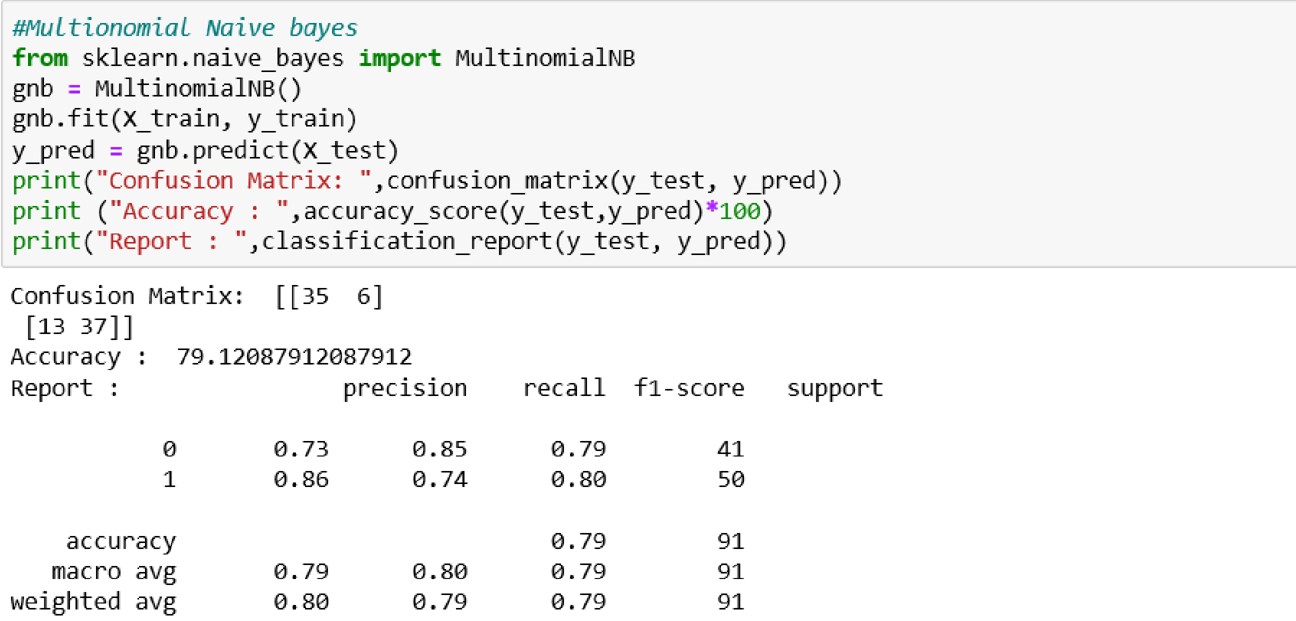
Figure 5.6: Gaussian Naive Bayes

Naive Bayes is an algorithm for the classification of binary and multi class problems. When represented using binary or categorical input values the technique is easiest to understand. This is a very strong conclusion that which actual data is most impossible. The method nonetheless performs remarkably well on data where this presumption does not hold.

It is possible to extend Naive Bayes to real-valued attributes, most generally by using a Gaussian distribution. This naive Bayes extension is called the Gaussian Naive Bayes. Certain functions can be used to approximate the data distribution, but the Gaussian is the simplest to work with, since you only need to measure the mean and standard deviation from your training statistics.

Above, we determined the probabilities for input esteems for each class utilizing a recurrence. With genuine esteemed information sources, we can compute the mean and standard deviation of information esteems (x) for each class to abridge the appropriation.This implies notwithstanding the probabilities for each class, we should likewise store the mean and standard deviations for each info variable for each class.

5.7 Multi nominal Naive Bayes



```
#Multionomial Naive bayes
from sklearn.naive_bayes import MultinomialNB
gnb = MultinomialNB()
gnb.fit(X_train, y_train)
y_pred = gnb.predict(X_test)
print("Confusion Matrix: ",confusion_matrix(y_test, y_pred))
print ("Accuracy : ",accuracy_score(y_test,y_pred)*100)
print("Report : ",classification_report(y_test, y_pred))
```

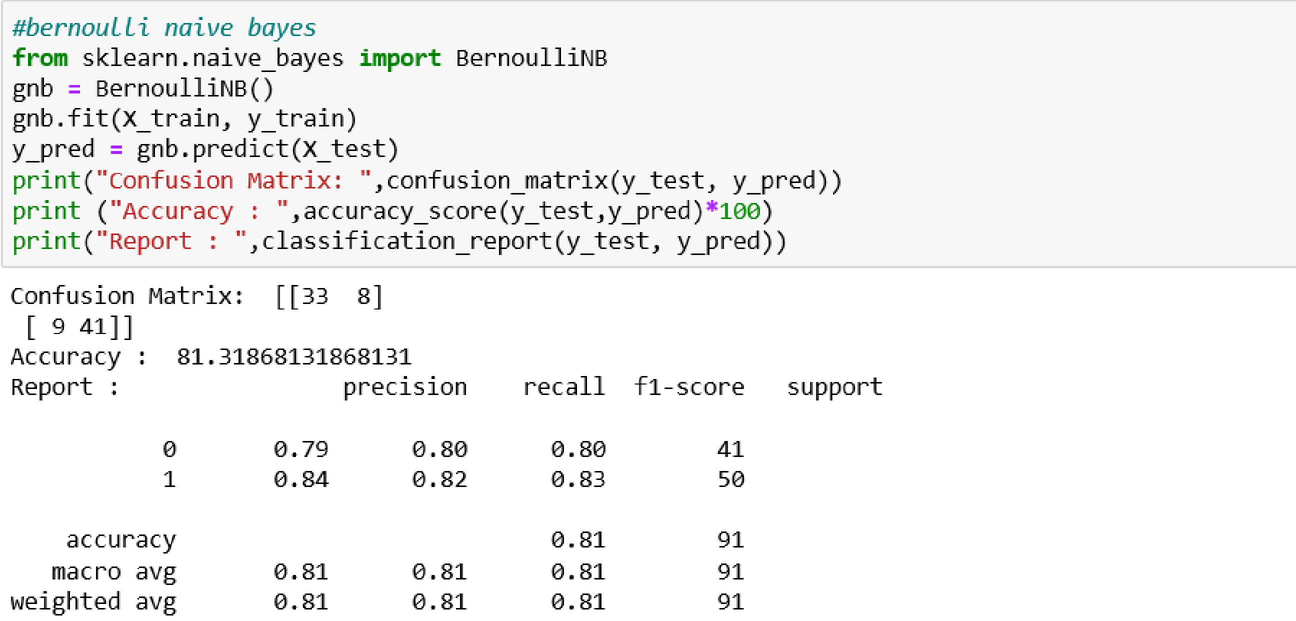
```
Confusion Matrix:  [[35  6]
 [13 37]]
Accuracy :  79.12087912087912
Report :                precision    recall  f1-score   support

           0       0.73      0.85      0.79        41
           1       0.86      0.74      0.80        50

    accuracy                           0.79        91
   macro avg       0.79      0.80      0.79        91
weighted avg       0.80      0.79      0.79        91
```

Figure 5.7: Multi nominal Naive Bayes

Multinomial Naive Bayes is a generalized variant of Naive Bayes which is more explicitly designed for text documents. Simple naive Bayes will model a text as the existence and absence of particular terms, multinomial naive Bayes directly model the word counts and change the underlying formulas to be discussed in.

It assesses the contingent likelihood of a specific word given a class as the general recurrence of term t in reports having a place with class(c). The variety considers the quantity of events of term t in preparing archives from class (c), including numerous events.

5.8 Bernoulli Naive Bayes



```
#bernoulli naive bayes
from sklearn.naive_bayes import BernoulliNB
gnb = BernoulliNB()
gnb.fit(X_train, y_train)
y_pred = gnb.predict(X_test)
print("Confusion Matrix: ",confusion_matrix(y_test, y_pred))
print ("Accuracy : ",accuracy_score(y_test,y_pred)*100)
print("Report : ",classification_report(y_test, y_pred))
```

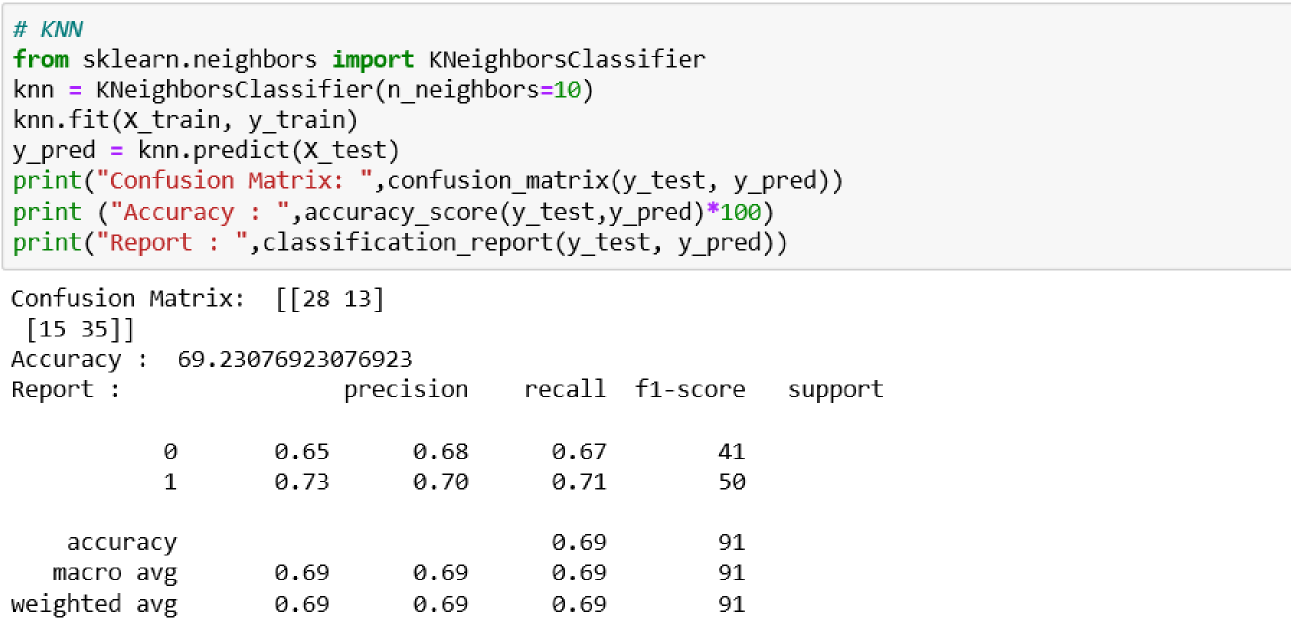
```
Confusion Matrix:  [[33  8]
 [ 9 41]]
Accuracy :  81.31868131868131
Report :                precision    recall  f1-score   support

           0       0.79      0.80      0.80        41
           1       0.84      0.82      0.83        50

    accuracy                           0.81        91
   macro avg       0.81      0.81      0.81        91
weighted avg       0.81      0.81      0.81        91
```

Figure 5.8: Bernoulli Naive Bayes

Bernoulli Naive Bayes is for oneway feature. Essentially, multinomial naive Bayes regards includes as occasion probabilities. Your model is given for nonbinary genuine esteemed highlights (x,y), which don't only lie in the interim [0,1], so the models don't make a difference to your highlights.

A normal model for either Bernoulli or multinomial NB is record order, where the highlights speak to the nearness of a term (in the Bernoulli case) or the likelihood of a term (in the multinomial case).

5.9 K-Nearest Neighbors (KNN)



```
# KNN
from sklearn.neighbors import KNeighborsClassifier
knn = KNeighborsClassifier(n_neighbors=10)
knn.fit(X_train, y_train)
y_pred = knn.predict(X_test)
print("Confusion Matrix: ",confusion_matrix(y_test, y_pred))
print ("Accuracy : ",accuracy_score(y_test,y_pred)*100)
print("Report : ",classification_report(y_test, y_pred))
```

```
Confusion Matrix:  [[28 13]
 [15 35]]
Accuracy :  69.23076923076923
Report :                precision    recall  f1-score   support

           0       0.65      0.68      0.67        41
           1       0.73      0.70      0.71        50

    accuracy                           0.69        91
   macro avg       0.69      0.69      0.69        91
weighted avg       0.69      0.69      0.69        91
```

Figure 5.9: K-Nearest Neighbors (KNN)

Calculation of the nearest neighbors (k-NN) is a non-parametric technique used for arrangement and regression. In the two instances, the data consists of the nearest k part space preparing models. The yield depends on whether k-NN is used for characterization or recurrence. The yield in k-NN order is class participation. An element is ordered by a majority vote of its neighbors, the article being doled out to its closest neighbors to the class generally basic. In case k = 1, the article is only allocated to the class of that closest solitary neighbour at that level.

CHAPTER 6

CODING

Listing 6.1: Code

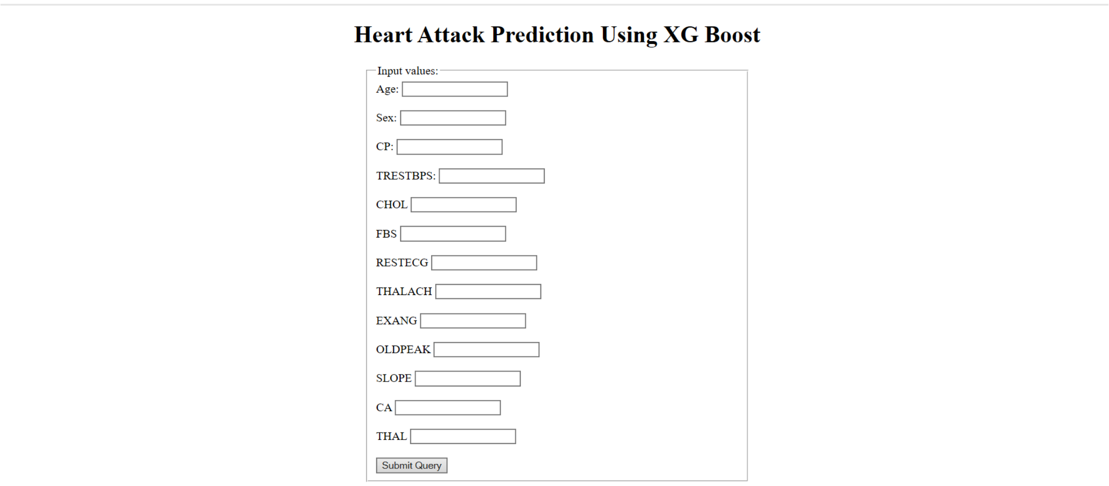
```
import numpy as np
import pandas as pd
from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score
from sklearn.metrics import classification_report
from sklearn.metrics import r2_score
import pickle
df=pd.read_csv('heart.csv')
df.isnull().sum()
df=df.replace('?',np.nan)
df=df.fillna(df.mean())
df.isnull().sum().any()
x=df.drop(['target'],axis=1)
y=df['target']

from sklearn import preprocessing
lbl = preprocessing.LabelEncoder()
x['ca'] = lbl.fit_transform(x['ca'].astype(str))
x['thal'] = lbl.fit_transform(x['thal'].astype(str))

from sklearn.preprocessing import Normalizer
scaler = Normalizer()
scaler.fit(x)
scaler.transform(x)
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(x, y, test_size = 0.3,
 random_state = 42)

# XGBoost
from xgboost.sklearn import XGBClassifier
from sklearn import metrics
classifier = XGBClassifier(silent=True,
scale_pos_weight=1,
learning_rate=0.00001,
colsample_bytree = 0.2,
subsample = 0.8,
n_estimators=20,
reg_alpha = 0.3,
max_depth=6,
gamma=10, min_child_weight=5,seed=30)
classifier.fit(X_train, y_train)
y_pred = classifier.predict(X_test)
predictions = [round(value) for value in y_pred]
print("Confusion␣Matrix:␣",confusion_matrix(y_test, predictions))
print ("Accuracy␣:␣",metrics.accuracy_score(y_test,predictions)*100)
print("Report␣:",classification_report(y_test, predictions))



###Front End Code
import flask
import pickle
import pandas as pd

# Use pickle to load in the pre-trained model
with open(f'model/chait.pkl', 'rb') as f:
model = pickle.load(f)

# Initialise the Flask app
app = flask.Flask(__name__, template_folder='templates')

# Set up the main route
@app.route('/', methods=['GET'])
def main_get():
if flask.request.method == 'GET':
# Just render the initial form, to get input
return(flask.render_template('main.html'))

@app.route("/", methods=['POST'])
```

```
def main_post():
if flask.request.method == 'POST':
# Extract the input
age = flask.request.form['age']
sex = flask.request.form['sex']
cp = flask.request.form['cp']
trestbps = flask.request.form['trestbps']
chol = flask.request.form['chol']
fbs = flask.request.form['fbs']
restecg = flask.request.form['restecg']
thalach = flask.request.form['thalach']
exang = flask.request.form['exang']
oldpeak = flask.request.form['oldpeak']
slope = flask.request.form['slope']
ca = flask.request.form['ca']
thal = flask.request.form['thal']

# Make DataFrame for model
input_variables = pd.DataFrame([[age, sex, cp, trestbps, chol, fbs, restecg,
 thalach, exang, oldpeak, slope, ca, thal]],
columns=['age', 'sex', 'cp', 'trestbps', 'chol', 'fbs', 'restecg', 'thalach',
 'exang', 'oldpeak', 'slope', 'ca', 'thal'],
dtype=float,
index=['input'])

# Get the model's prediction
prediction = model.predict(input_variables)[0]

# Render the form again, but add in the prediction and remind user
# of the values they input before
return flask.render_template('result.html',
original_input={'age':age,
'sex':sex,
'cp':cp,
'trestbps':trestbps,
'chol':chol,
'fbs':fbs,
'restecg':restecg,
'thalach':thalach,
'exang':exang,
'oldpeak':oldpeak,
'slope':slope,
'ca':ca,
'thal':thal},
result=prediction,
)


if __name__ == '__main__':
app.run()
```

CHAPTER 7

OUTPUT

7.1 User Interface

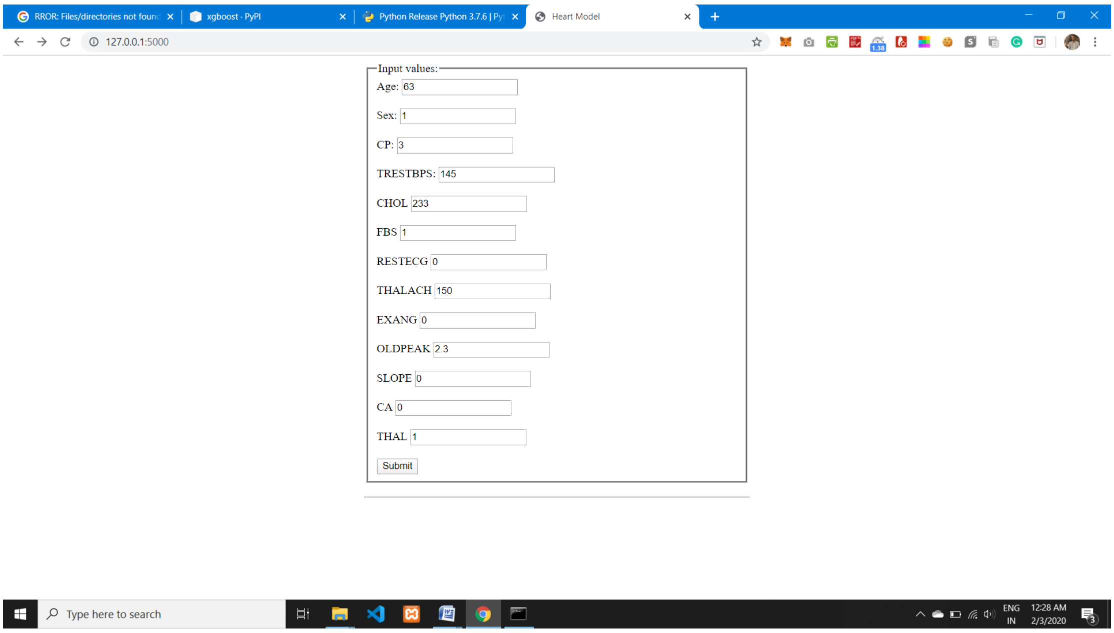


Heart Attack Prediction Using XG Boost

Figure 7.1: User Interface

In the User Interface the data is taken and there are 13 attributes that needs to be entered and then click on submit in order to view the result.

7.2 Input



Input values:

Age: 63

Sex: 1

CP: 3

TRESTBPS: 145

CHOL 233

FBS 1

RESTECG 0

THALACH 150

EXANG 0

OLDPEAK 2.3

SLOPE 0

CA 0

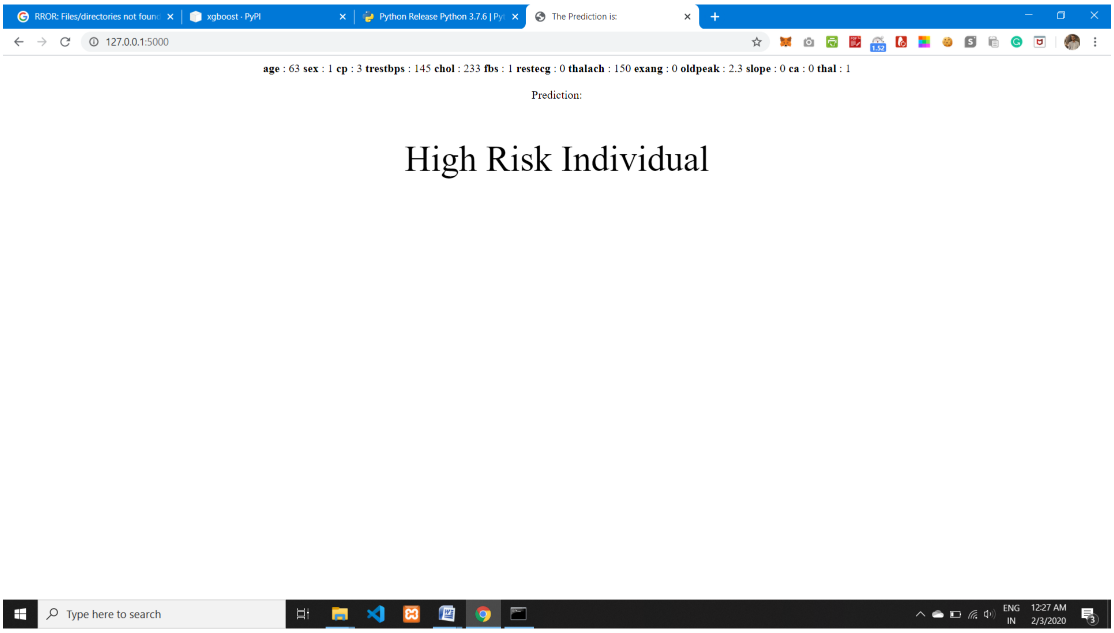
THAL 1

Submit

Figure 7.2: Input Data to the UI

Inputting the values here and this data is sent to the model. After that the result it sent to the output page.

7.3 Result



age : 63 **sex** : 1 **cp** : 3 **trestbps** : 145 **chol** : 233 **fbs** : 1 **restecg** : 0 **thalach** : 150 **exang** : 0 **oldpeak** : 2.3 **slope** : 0 **ca** : 0 **thal** : 1

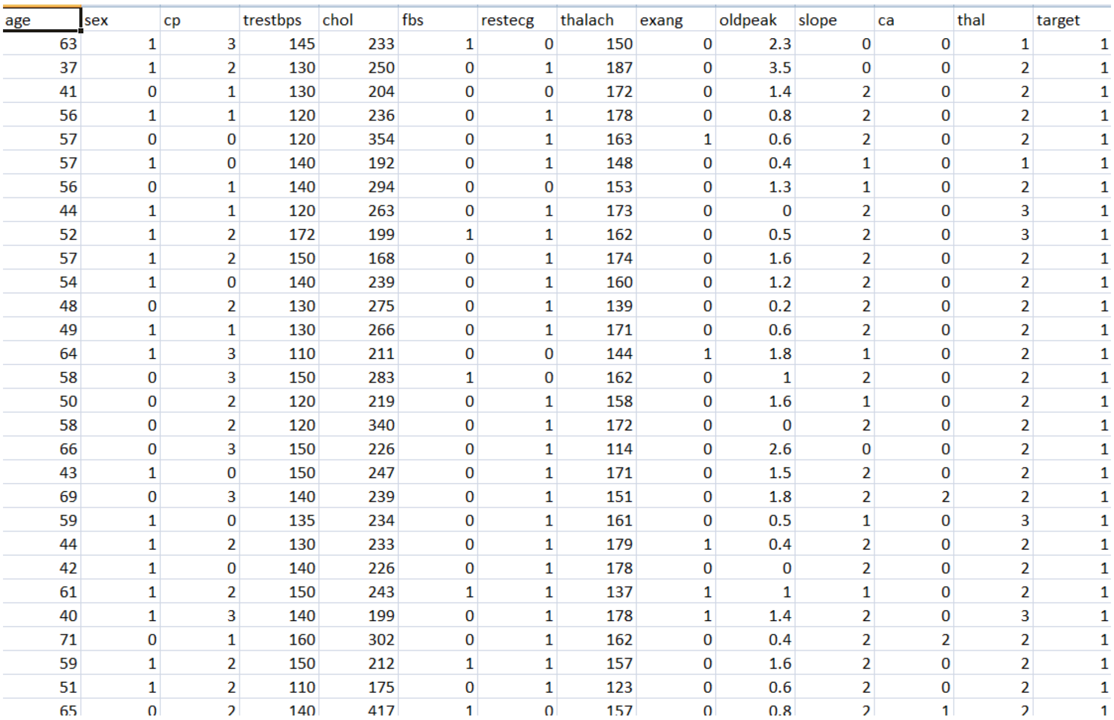
Prediction:

High Risk Individual

Figure 7.3: Result of the Prediction

Result is predicted if it is a high risk individual then its a heart attack and if it is a low risk individual then it is not an heart attack.

7.4 Data Set



age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
63	1	3	145	233	1	0	150	0	2.3	0	0	1	1
37	1	2	130	250	0	1	187	0	3.5	0	0	2	1
41	0	1	130	204	0	0	172	0	1.4	2	0	2	1
56	1	1	120	236	0	1	178	0	0.8	2	0	2	1
57	0	0	120	354	0	1	163	1	0.6	2	0	2	1
57	1	0	140	192	0	1	148	0	0.4	1	0	1	1
56	0	1	140	294	0	0	153	0	1.3	1	0	2	1
44	1	1	120	263	0	1	173	0	0	2	0	3	1
52	1	2	172	199	1	1	162	0	0.5	2	0	3	1
57	1	2	150	168	0	1	174	0	1.6	2	0	2	1
54	1	0	140	239	0	1	160	0	1.2	2	0	2	1
48	0	2	130	275	0	1	139	0	0.2	2	0	2	1
49	1	1	130	266	0	1	171	0	0.6	2	0	2	1
64	1	3	110	211	0	0	144	1	1.8	1	0	2	1
58	0	3	150	283	1	0	162	0	1	2	0	2	1
50	0	2	120	219	0	1	158	0	1.6	1	0	2	1
58	0	2	120	340	0	1	172	0	0	2	0	2	1
66	0	3	150	226	0	1	114	0	2.6	0	0	2	1
43	1	0	150	247	0	1	171	0	1.5	2	0	2	1
69	0	3	140	239	0	1	151	0	1.8	2	2	2	1
59	1	0	135	234	0	1	161	0	0.5	1	0	3	1
44	1	2	130	233	0	1	179	1	0.4	2	0	2	1
42	1	0	140	226	0	1	178	0	0	2	0	2	1
61	1	2	150	243	1	1	137	1	1	1	0	2	1
40	1	3	140	199	0	1	178	1	1.4	2	0	3	1
71	0	1	160	302	0	1	162	0	0.4	2	2	2	1
59	1	2	150	212	1	1	157	0	1.6	2	0	2	1
51	1	2	110	175	0	1	123	0	0.6	2	0	2	1
65	0	2	140	417	1	0	157	0	0.8	2	1	2	1

Figure 7.4: Datset

Here in the dataset we have 13 attributes which are the factors that causing the heart attack prediction and the last column of the dataset is the target which means the result of heart attack. 0 means no risk of heart attack. 1 means there is high risk of heart attack.

CHAPTER 8

CONCLUSION

The dataset was taken from UCI's machine learning repository [5] and pre-processed. The last dataset includes 13 indicator factors and one reaction variable named num. If that num is 0, implies under 50% of the veins are narrowing and estimation of 1 implies that over 50% of veins narrowing i.e., the expectation is 'high hazard person'. Since the information follows a typical circulation, the Gaussian Naïve Bayes calculation was utilized for the grouping. XGBoost has demonstrated to give the best exactness of 90.46%. A similar classifier can be gotten to with the assistance of a web interface for the comfort of the user.

CHAPTER 9

FUTURE ENHANCEMENT

In future if there is any newly improved algorithm is discovered then we can check with more efficient one, in the future. Silent heart attacks can be added as an attribute where Heart Attack Prediction can be helpful to detect before 6 hours by inflow of Cardiac Bio markers in the blood.

CHAPTER 10

REFERENCES

1. Manasa. K. N, Prince Kumar Gupta. (2017). Disease Prediction by Machine Learning with the help of Big Data from Healthcare Communities. International Journal of Engineering Science And Computing (IEEE).

2. Nitten S. Rajliwall, Rachel Davey, Girija Chetty. (2018), Cardiovascular Risk Prediction Using XGBoost. Institute of Electrical and Electronics Engineers (IEEE).

3. SenthilKumar Mohan, Chandrasegar Thirumalai, Gautam Srivatsava. (2019). Effective Heart Disease Prediction Using Hybrid Machine Learning Techniques, Institute of Electrical and Electronics Engineers (IEEE).

4. SenthilKumar Mohan, Chandrasegar Thirumalai, Gautam Srivatsava. (2019). Effective Heart Disease Prediction Using Hybrid Machine Learning Techniques, Institute of Electrical and Electronics Engineers (IEEE).

5. Shadman Nashif, Md. Rakib Raihan (2018), Heart Disease Detection by Machine Learning Algorithms and a Real-Time Cardiovascular Health Monitoring System, World Journal of Engineering and Technology.

6. Susmitha Manikandan (2017). Heart attack Prediction System, International Conference on Energy, Communication, Data Analytics and Soft Computing (IEEE).

7. Manasa. K. N, Prince Kumar Gupta. (2017). Disease Prediction by Machine Learning with the help of Big Data from Healthcare Communities. International Journal of Engineering Science And Computing (IEEE).

8. Nitten S. Rajliwall, Rachel Davey, Girija Chetty. (2018), Cardiovascular Risk Prediction Using XGBoost. Institute of Electrical and Electronics Engineers (IEEE).

9. SenthilKumar Mohan, Chandrasegar Thirumalai, Gautam Srivatsava. (2019). Effective Heart Disease Prediction Using Hybrid Machine Learning Techniques, Institute of Electrical and Electronics Engineers (IEEE).

10. SenthilKumar Mohan, Chandrasegar Thirumalai, Gautam Srivatsava. (2019). Effective Heart Disease Prediction Using Hybrid Machine Learning Techniques, Institute of Electrical and Electronics Engineers (IEEE).

11. Shadman Nashif, Md. Rakib Raihan (2018), Heart Disease Detection by Machine Learning Algorithms and a Real-Time Cardiovascular Health Monitoring System, World Journal of Engineering and Technology.

Project report M. Karthikeyan

ORIGINALITY REPORT

10%	3%	5%	9%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Senthilkumar Mohan, Chandrasegar Thirumalai, Gautam Srivastava. "Effective Heart Disease Prediction Using Hybrid Machine Learning Techniques", IEEE Access, 2019 Publication	1%
2	en.wikipedia.org Internet Source	1%
3	Submitted to SRM University Student Paper	1%
4	Submitted to AlHussein Technical University Student Paper	1%
5	Submitted to Higher Education Commission Pakistan Student Paper	1%
6	Submitted to University College London Student Paper	<1%
7	Submitted to Universiti Teknologi MARA Student Paper	<1%

"Twitter Sentiment Recognition using Support

Format - I

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

(Deemed to be University u/s 3 of UGC Act, 1956)

Office of Controller of Examinations

REPORT FOR PLAGIARISM CHECK ON THE DISSERTATION/PROJECT REPORTS FOR UG/PG PROGRAMMES
(To be attached in the dissertation/ project report)

1	Name of the Candidate **(IN BLOCK LETTERS)**	M.CHAITANYA RAJEEV
2	Address of the Candidate	PLOT 577 ROAD 32 JUBILEE HILLS HYDERABAD, TELANGANA **Mobile Number :** 8686730155
3	Registration Number	RA1611003010350
4	Date of Birth	26-05-1998
5	Department	COMPUTER SCIENCE AND ENGINEERING
6	Faculty	Mr. M. KARTHIKEYAN
7	Title of the Dissertation/Project	HEART ATTACK PREDICTION USING XGBOOST
8	Whether the above project/dissertation is done by	Individual or group : GROUP (Strike whichever is not applicable) a) If the project/ dissertation is done in group, then how many students together completed the project : 2 b) Mention the Name & Register number of other candidates : Ch.SAI CHAITANYA REDDY & RA1611003010374
9	Name and address of the Supervisor / Guide	MR.M.KARTHIKEYAN karthikm1@srmist.edu.in **Mail ID : Mobile Number :** 9994185313
10	Name and address of the Co-Supervisor / Co- Guide (if any)	**Mail ID : Mobile Number :**

11	Software Used	TURNITIN
12	Date of Verification	16-MAY-2020
13	**Plagiarism Details: (to attach the final report from the software)**	

Chapter	**Title of the Chapter**	**Percentage of similarity index (including self citation)**	**Percentage of similarity index (Excluding self citation)**	**% of plagiarism after excluding Quotes, Bibliography, etc.,**
1	INTRODUCTION	2%		
2	LITERATURE SURVEY	4%		
3	PROPOSED METHADOLOGY	2%		
4	IMPLEMENTATION	1%		
5	COMPARISION OF DIFFERENT ALGORITHMS	1%		
6	OUTPUT & RESULT	0%		
7				
8				
9				
10				
Appendices				

I / We declare that the above information have been verified and found true to the best of my / our knowledge.

Signature of the Candidate	MR.M.KARTHIKEYAN M. Karthikeyan **Name & Signature of the Staff (Who uses the plagiarism check software)**
MR.M.KARTHIKEYAN M. Karthikeyan **Name & Signature of the Supervisor/Guide**	**Name & Signature of the Co-Supervisor/Co-Guide**

Name & Signature of the HOD

Format - I

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

(Deemed to be University u/s 3 of UGC Act, 1956)

Office of Controller of Examinations

REPORT FOR PLAGIARISM CHECK ON THE DISSERTATION/PROJECT REPORTS FOR UG/PG PROGRAMMES
(To be attached in the dissertation/ project report)

1	Name of the Candidate **(IN BLOCK LETTERS)**	CH.SAI CHAITANYA REDDY
2	Address of the Candidate	DOOR NO:28-1169/2,GANGANAPALLI, KANNIAH NAIDU COLONY, CHITTOOR, ANDRA PRADESH **Mobile Number :** 9952918843
3	Registration Number	RA1611003010374
4	Date of Birth	07-01-1999
5	Department	COMPUTER SCIENCE AND ENGINEERING
6	Faculty	Mr. M. KARTHIKEYAN
7	Title of the Dissertation/Project	HEART ATTACK PREDICTION USING XGBOOST
8	Whether the above project/dissertation is done by	Individual or group : GROUP (Strike whichever is not applicable) a) If the project/ dissertation is done in group, then how many students together completed the project : 2 b) Mention the Name & Register number of other candidates : M.CHAITANYA RAJEEV & RA1611003010350
9	Name and address of the Supervisor / Guide	MR.M.KARTHIKEYAN karthikm1@srmist.edu.in **Mail ID : Mobile Number :** 9994185313
10	Name and address of the Co-Supervisor / Co- Guide (if any)	**Mail ID : Mobile Number :**

11	Software Used	TURNITIN
12	Date of Verification	16-MAY-2020
13	**Plagiarism Details: (to attach the final report from the software)**	

Chapter	**Title of the Chapter**	**Percentage of similarity index (including self citation)**	**Percentage of similarity index (Excluding self citation)**	**% of plagiarism after excluding Quotes, Bibliography, etc.,**
1	INTRODUCTION	2%		
2	LITERATURE SURVEY	4%		
3	PROPOSED METHADOLOGY	2%		
4	IMPLEMENTATION	1%		
5	COMPARISION OF DIFFERENT ALGORITHMS	1%		
6	OUTPUT & RESULT	0%		
7				
8				
9				
10				
Appendices				

I / We declare that the above information have been verified and found true to the best of my / our knowledge.

Ch.Sai Chaitanya Reddy **Signature of the Candidate**	MR.M.KARTHIKEYAN M. Karthikeyan **Name & Signature of the Staff (Who uses the plagiarism check software)**
MR.M.KARTHIKEYAN M. Karthikeyan **Name & Signature of the Supervisor/Guide**	**Name & Signature of the Co-Supervisor/Co-Guide**

Name & Signature of the HOD

Heart Attack Prediction Using XGBoost

[1]Karthikeyan.M*, [2]Chaitanya Rajeev Myakala, [3]Sai Chaitanya Chappidi
[1,2,3] Department of Computer Science and Engineering, SRM Institute of Science and Technology, Kattankulathur, Chennai, Tamilnadu, India-603 203.

[1]karthikm1@srmist.edu.in,[2]chaitanyakln@gmail.com,*
[3]saichaithanya904@gmail.com

Abstract

Heart disease has become more common these days. Heart attacks are the majority death cases in the current time. Lot of risk involved in people's life. Machine learning has been effective in taking decisions and predicting from huge amount of data set given by the medical healthcare industry. We have seen Machine Learning techniques are been used irrespective of the fields. There have been various factors that have been affecting the risk to the lives. Variation in Blood Pressure, sugar, pulse rate, shortness of breath. Etc. can lead to cardiovascular diseases that in turn block the blood vessels that carry rich oxygenated blood. This may cause coronary artery disease, heart failure, congenital heart disease. There are several attributes that are taken into account in order to predict the heart attack. There are many forms of heart diseases that can be predicted through various factors and can be diagnosed with various medical tests. The main aim of this project is to predict heart attack with at most accuracy. We have used Extreme gradient boosting algorithm for predication of heart attack. The pre research and reading obtained front this algorithm is used in detection of heart attack at early level and can be cured by proper diagnosis. We have produced an efficient performance level with an accuracy level over 90% though the prediction model for predicting of heart attack. Using machine learning to predict this kind of diseases is a major necessity for brining the healthcare industry to great heights.

Keywords: *machine learning; prediction; user interface; Artificial Neural Network.*

1. Introduction

There has been so many diseases which affect us badly and one of them is Heart Diseases. The heart is one of the most precious organs of the body. It pumps blood in the cardiovascular system which consists of arteries and veins through which blood reaches all the parts of the body. The heart is a solid organ which is situated between the lungs, in the center compartment of the chest. Cardiovascular patients with heart diseases live at home and seek for healthcare service when they feel abnormal. The primary however, these symptoms don't effect until the very late stage of the disease and it's important to know that the damage is already done, nothing can be done and most of the patients die before they get access to treatment the main aim is to improve heart disease and reduce the death rate. Machine learning is truly creating a revolution in any field to that matter. Healthcare industry is slowly adopting the machine learning models. There has been massive improvement mainly in the field of medical and healthcare services. The Integration of sensors with the communication system has helped the patients to be under observation from anywhere irrespective of where they are what they are doing. Healthcare today is shifting from a clinic-centric to a patient-centric level. Inappropriate diet/hypertension causes dysfunction of heart which in turn leads to heart failure which is commonly known as a heart attack. Nowadays we see increasing hypertension and bad dieting, the numbers of heart attacks is a increasing. In this fast-paced world, it is easy to create an automated health check-up for person so that a person is known before any of the factors leading to a heart attack. It is hard to recognize the disease because of numerous factors that sum to a disease. Picking up individual factors and predicting is inaccurate. The proposed system aims to solve the issue by predicting that tells the risk of heart attack. We have used dataset from UCI's Machine learning repository. Among various Algorithms extreme gradient boosting algorithm yield us the better result. Chest pain is one of the most remarkable symptoms of heart attack. Predicting heart attack is one of the most important aspect where if there is any delay in detecting it may lead to damage to heart muscle. Myocardial dead tissue happens when there is blockage in coronary conduit that provisions rich oxygenated blood to heart. This blockage is brought about by the cholesterol and cell squander items. When there is a blockage the cells that are present after the blockage. Medical sector has humongous data. Identifying the required data and taking the appropriate data in order to predict.

2. Literature Surevey

Heart Attack Prediction Using Machine Learning Methods is very useful and proved its importance in the past few years. Manasa. K. N, Prince Kumar Gupta presented a system which is suitable for real-time heart diseases prediction and can be used by the users who have coronary disease.

Nitten S. Rajliwall, Rachel Davey, Girija Chetty. In this paper they have proposed a framework which is based on supervised learning algorithms and processing based on batch level involving cohort separation and they have used filtering based on gender , education level and age. SenthilKumar Mohan, Chandrasegar Thirumalai, Gautam Srivatsava proposed hybrid HRFLM approach is used combining the characteristics of Random Forest (RF) and Linear Method (LM).

Shadman Nashif, Md. Rakib Raihan proposed a model which is a cloud based on heart diseases prediction model which is been proposed to identify impending heart diseases with the use of machine learning algorithms.Susmitha Manikandan In this paper sample module of the system consist of binary classification model which is used to predict the risk factor of a patient based on their medical data.

Aditi Gavhane, GowtamiKokkula, Isha Pandya. In this paper they proposed a system where they used the NN Algorithm and multi layered perceptron for training and testing the dataset.D. K. Ravish, K.J. Shanthi, Nayana R Shenoy, S. Nisargh In this paper they have developed an efficient way to acquire the clinical and ECG data. For training the ANN to accurately diagnose and predict heart abnormalities if found any.

C. M ChethanMalode, K. Bhargavi, B. G Gunasheela In this paper they have used fuzzy rule and set theory which is concatenated with SVM classifier to identify and differentiate heart attack risk among adolescents.Kwenbing Chang, Yinglai Liu, Xueyi Wu, YiyongXaio, Shenghan Zhou, Wen Cao. In this paper they have used XGBSVM which means Xgboost plus SVM hybrid model to predict heart diseases within three years.

Procheta Nag, SaikatMondal, Foysal Ahmed, Arun More, M.Raihanused. In this paper they have used classification techniques of data mining and decision tree to predict whether the chest pain is for heart attack or any other. Boshra Bahrami, Mirsaeid Hosseini Shirvani In this paper they have provided an intuition about data mining technique used to forecast cardiovascular diseases.

3. Related Work

In previous works there has been several state of the art classification models that have been used..[1] Manasa. K. N, Prince Kumar Gupta used Random Forest algorithm which gave an accuracy of 89%. [3] SenthilKumar Mohan, ChandrasegarThirumalai, GautamSrivatsava used Random Forest With Linear Model gave an accuracy of 88.7%. [4] ShadmanNashif, Md. RakibRaihan used algorithms namely Naïve Bayes, Decision Tree, K- nearest Neighbour, Random Forest in 10-fold Cross-Validation which gave an accuracy of 80%. [5] SusmithaManikandanused Naïve Bayes algorithm which gave an accuracy of 81.25%. [8] C. M ChethanMalode, K. Bhargavi, B. G Gunasheela used set thoery and Fuzzy theory enabled SVM Approach which gave an accuracy of 85.6%.

4. XGBoost

XGBoost is an efficient implementation of gradient boosting techniques. Although there is no new mathematical break through. It is one of the well-built versions of Gradient boosting which used to optimal and to improve accuracy. It contains both a linear model and a tree learning algorithm.

Boosting is a procedure that utilizes a lot of AI calculations to join frail learners to frame solid learners so as to increase the accuracy of the model. Boosting is a kind of ensemble learning. It comprises of Sequential learning (Boosting) and Parallel Learning (Bagging) eg: Random Forest. Ensemble learning is a strategy that is utilized to upgrade the exhibition of the AI model with improved proficiency and exactness.

To create various weak learners and consolidate their predictions from one in number standard. Presently their weak learners are created by applying base Machine Learning algorithms on various distributions of the dataset. By and large, base AI calculations are decision trees. so what these base algorithms do is that they produce weak guidelines for every iteration. so after numerous cycles, the weak learners are consolidated and they structure solid learners that will anticipate the more exact result.

There are three sorts of Boosting procedures. 1. Adaptive Boosting 2.Gradient Boosting 3. Extreme gradient boosting. Among this XG Boosting is a propelled rendition of gradient boosting strategy that is intended to concentrate on computational speed and model productivity. It really falls under the classification of dispersed machine learning community most advanced version of gradient boosting. Some of the advantages of XGBoost algorithm are it's highly flexible which means we can set custom evaluation criteria and optimization objectives. Processing is faster than gradient boosting..

It has built-in methods to handle missing data. Tree pruning: In Gradient boosting algorithm stops when it encounters -ve loss in the split but in case of XGBoost it digs to maximum depth and starts pruning the splits without any positive gain. It is a decision tree based algorithms which is considered best for small or medium structured date.Building models using XGBoost is quite easy. But improving it’s efficiency is really hard. We have various parameters in XGBoost, which requires tuning.

4. Proposed Work

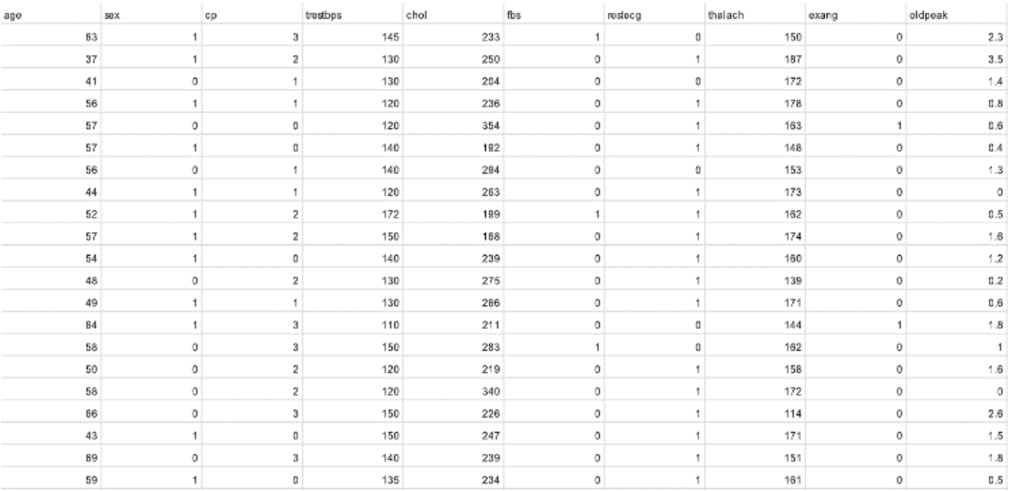
In this paper, we have built a prototype model which has a binary classification to measure and to intimate the risk of causing heart attack based on the medical data of the individual data. The dataset that we have used is obtained from University of California, Irvine’s machine learning repository. The unstructured dataset are converted to a structured dataset. The dataset has 14 attributes from which 13 are predictor variables. One is a binary response variable. Extreme Gradient Boosting was used for the classification process.

5. Implementation

5.1. Data Preprocessing

After gatheringmanyfiles the data is processed. Various patient files are present in the dataset. There are 303 files in total. 6 files in the totalhave some values missing. The files with missing values have been taken out. Now for the purpose of pre-processing the left 297 record are used. A variable is set for the parameters of the dataset. This variable is helpful to detect whether the person is more/less likely to get heart attack. If the patient is more likely to get heart attack, the value is set to 1, otherwise it is set to 0. The results show that 137 records out of 297 with value 1 indicating the occurrence of the heartattack and the rest 160 columns have0 as value indicating less chance of heart attack.

Table 1. Sample Of The Dataset



age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak
63	1	3	145	233	1	0	150	0	2.3
37	1	2	130	250	0	1	187	0	3.5
41	0	1	130	204	0	0	172	0	1.4
56	1	1	120	236	0	1	178	0	0.8
57	0	0	120	354	0	1	163	1	0.6
57	1	0	140	192	0	1	148	0	0.4
56	0	1	140	294	0	0	153	0	1.3
44	1	1	120	263	0	1	173	0	0
52	1	2	172	199	1	1	162	0	0.5
57	1	2	150	168	0	1	174	0	1.6
54	1	0	140	239	0	1	160	0	1.2
48	0	2	130	275	0	1	139	0	0.2
49	1	1	130	266	0	1	171	0	0.6
64	1	3	110	211	0	0	144	1	1.8
58	0	3	150	283	1	0	162	0	1
50	0	2	120	219	0	1	158	0	1.6
58	0	2	120	340	0	1	172	0	0
66	0	3	150	226	0	1	114	0	2.6
43	1	0	150	247	0	1	171	0	1.5
69	0	3	140	239	0	1	151	0	1.8
59	1	0	135	234	0	1	161	0	0.5

The following parameters are present in the final numerical dataset. The dataset is in .csv format. There are a total of 14 parameters. They are:

1. Age
2. Sex
3. Chestpain
4. BloodPressure atRest
5. Heart rateachieved during peak
6. Electrocardiographic results at Rest
7. Fasting blood sugar
8. Cholesterol
9. ST depression induced by exercise
10. Induced Engina
11. Slope of STsegment during peak exercise
12. Count of majorvessels

13. Thal
14. Num – responsevariable

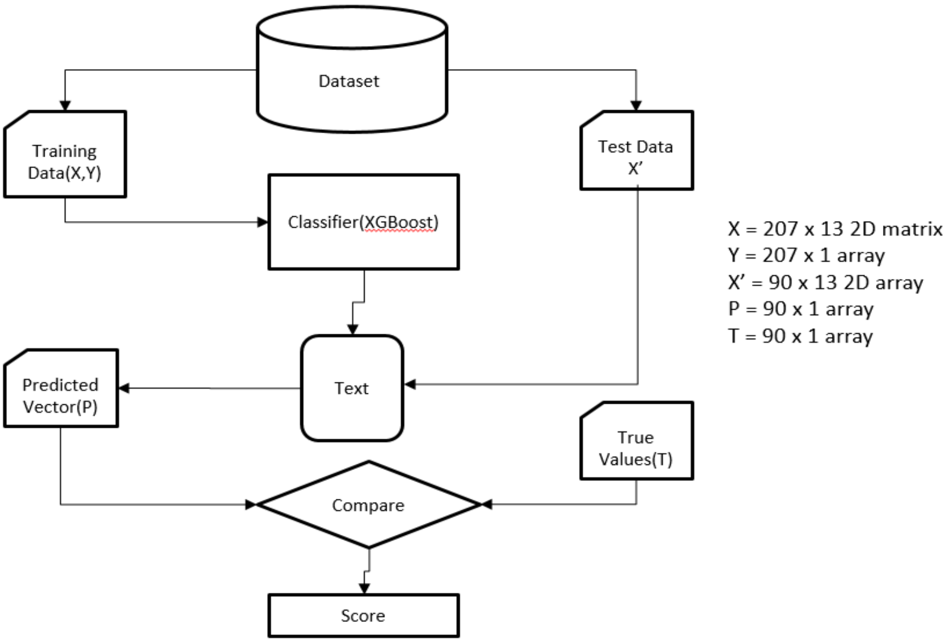


Figure 1. Data Procesing

5.2.Feature Selection and Reduction

Two parameters out of 13,are utilized to recognize the patient's data. The 11 parameters which are remaining are important. These 11 parameters are important for identification and knowing the condition of the heart. As forerly stated in the experiment, many machine learning techniques are used namely Linear Regression, Naive Bayes, Decision Tree, KNN, Multinomial Naive Bayes, SVM,XGBoost. The experiment was re done using many machine learning techniques with same attributes.

5.3.Classification and Modelling

Now various machine learning methods can be applied as our dataset is ready. Classification and Modelling is the important phase of thesystem, where the result of classification is obtained. Various algorithms are selected and their performance is compared. Out of all those algorithms XGBoost gives us the result with high accuracy.

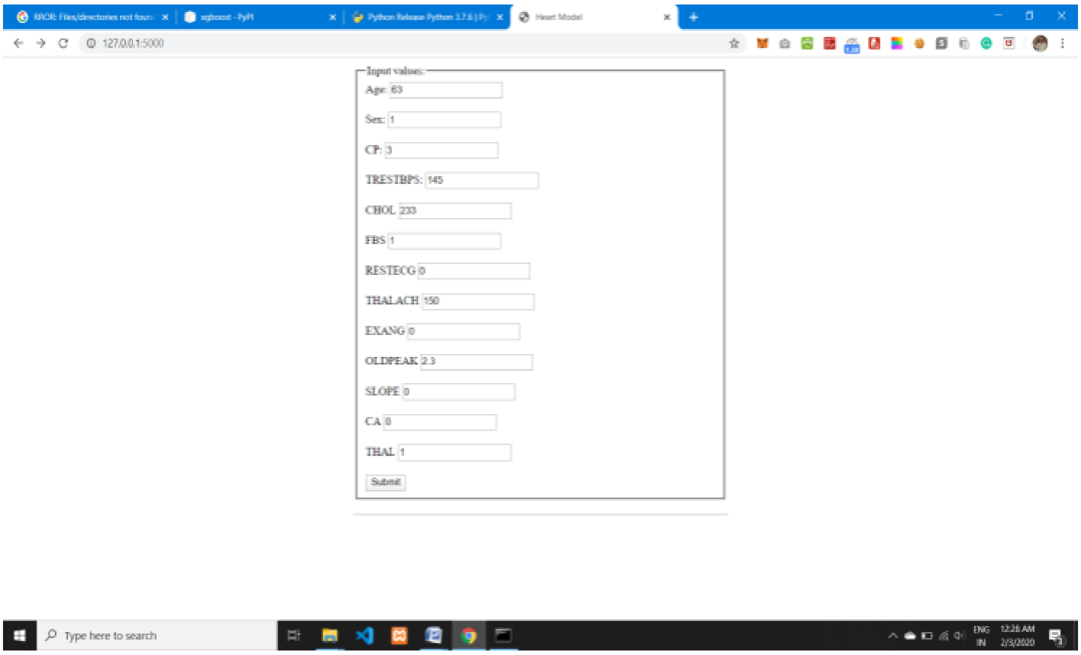
5.4.User Interface

In user interface we will be having a web application. In this application the person can enter his/her details and check for their risk of heart attack. With the data entered by the person his risk rate will be shown on the screen. If it is showing 'High Risk Individual' then the person is highly prone to heart attack and can consult his/her physician. If it shows 'Low Risk Individual' then the person is less prone to get heart attack. To make this web application 'flask' is used.

6. Results and Discussion

Below are the images of the user interface. Figure 2 shows the page of the application where the patient has to enter his/her data. Figure 3 shows the page of application where the risk of getting heart attack is shown after the user enters the details.

In Table II shows the confusion matrix based on which we can calculate the accuracy and many other parameters.



Input values:

Age: 63

Sex: 1

CP: 3

TRESTBPS: 145

CHOL 233

FBS 1

RESTECG 0

THALACH 150

EXANG 0

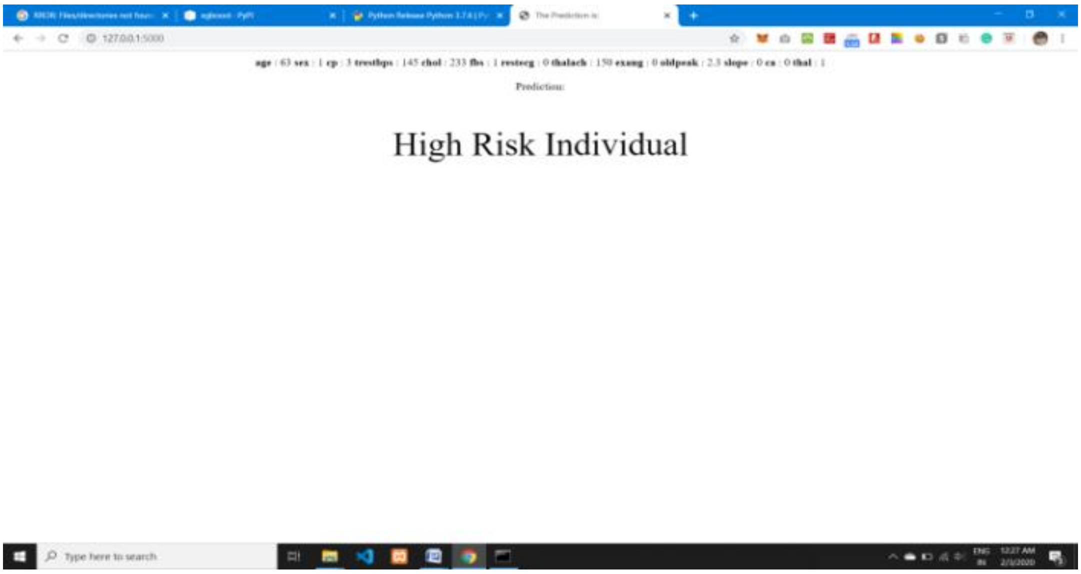
OLDPEAK 2.3

SLOPE 0

CA 0

THAL 1

Figure 2: The Web Interface where the input is given



age : 63 sex : 1 cp : 3 trestbps : 145 chol : 233 fbs : 1 restecg : 0 thalach : 150 exang : 0 oldpeak : 2.3 slope : 0 ca : 0 thal : 1

Prediction:

High Risk Individual

Figure 3: The predicted result after the prediction.

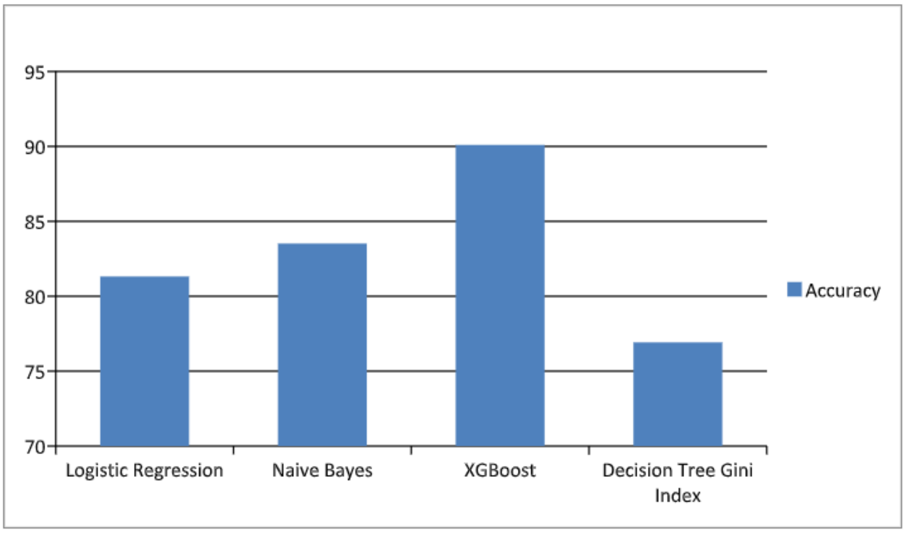
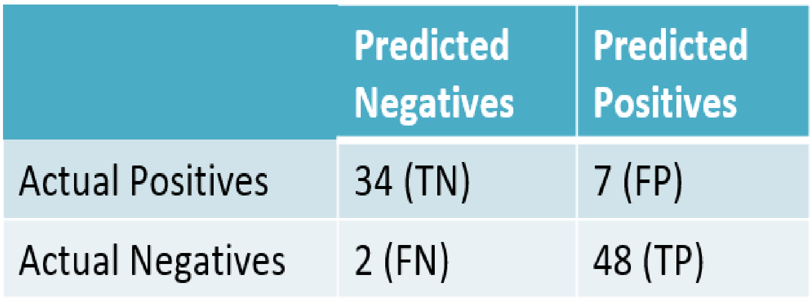


Figure 4: The accuracy compared with different algorithms

Table2. Confusion Matrix



	Predicted Negatives	Predicted Positives
Actual Positives	34 (TN)	7 (FP)
Actual Negatives	2 (FN)	48 (TP)

Confusion matrix is shown in Table 2. The following measures are calculated using on the confusion matrix. There are four valuesTN, FP,TP and FN. The calculated measures are very important for any classification system.

i) Accuracy = (TP+TN)/total = 0.9010
= 90.1%

ii) TPR=Recall=TP/(TP+FN) = 0.96
= 96%

iii) FPR=FP/(FP+TN)
= 0.1701
= 17.01%

iv) Specificity =TN/ActualNegatives
= 0.8292
= 82.92%

v) Precision =TP/PredictedPositives
= 0.8727

= 87.27%

vi) F-Measure=1/(1/recall)+(1/precision)

= 1/2.188

= 0.4570

Here TPR means True Positive Rate, FPR means False Positive Rate, TN means True Negatives, FP means False Positives, TP means True Positives, FN means False Negatives.

7. Conclusion

The dataset used in this experiment was taken from UCI's machine learning repository [5]. After taking the dataset from the repository it is pre-processed. There are 13 predictor parameters and 1 response variable in the dataset. If the value of response is 1, then the likelihood of the danger of heart attack is high and if the value of responseis 0, then the likelihood of the danger of heart attack is less.

By using XG Boost an accuracy of 90.46% is obtained. The same algorithm is used to predict and show the risk level of the individual on the user interface.

8. References

[1] Manasa. K. N, Prince Kumar Gupta. (2017). Disease Prediction by Machine Learning with the help of Big Data from Healthcare Communities. International Journal of Engineering Science And Computing (IEEE).

[2] Nitten S. Rajliwall, Rachel Davey, Girija Chetty. (2018), Cardiovascular Risk Prediction Using XGBoost. Institute of Electrical and Electronics Engineers (IEEE).

[3] SenthilKumar Mohan, Chandrasegar Thirumalai, Gautam Srivatsava. (2019). Effective Heart Disease Prediction Using Hybrid Machine Learning Techniques, Institute of Electrical and Electronics Engineers (IEEE).

[4] SenthilKumar Mohan, Chandrasegar Thirumalai, Gautam Srivatsava. (2019). Effective Heart Disease Prediction Using Hybrid Machine Learning Techniques, Institute of Electrical and Electronics Engineers (IEEE).

[5] Shadman Nashif, Md. Rakib Raihan (2018), Heart Disease Detection by Machine Learning Algorithms and a Real-Time Cardiovascular Health Monitoring System, World Journal of Engineering and Technology.

[6] Susmitha Manikandan (2017). Heart attack Prediction System, International Conference on Energy, Communication, Data Analytics and Soft Computing (IEEE).

[7] Manasa. K. N, Prince Kumar Gupta. (2017). Disease Prediction by Machine Learning with the help of Big Data from Healthcare Communities. International Journal of Engineering Science And Computing (IEEE).

[8] Nitten S. Rajliwall, Rachel Davey, Girija Chetty. (2018), Cardiovascular Risk Prediction Using XGBoost. Institute of Electrical and Electronics Engineers (IEEE).

[9] SenthilKumar Mohan, Chandrasegar Thirumalai, Gautam Srivatsava. (2019). Effective Heart Disease Prediction Using Hybrid Machine Learning Techniques, Institute of Electrical and Electronics Engineers (IEEE).

[10] SenthilKumar Mohan, Chandrasegar Thirumalai, Gautam Srivatsava. (2019). Effective Heart Disease Prediction Using Hybrid Machine Learning Techniques, Institute of Electrical and Electronics Engineers (IEEE).

[11] Shadman Nashif, Md. Rakib Raihan (2018), Heart Disease Detection by Machine Learning Algorithms and a Real-Time Cardiovascular Health Monitoring System, World Journal of Engineering and Technology.