PREDICTION OF CARDIAC DISEASE USING SUPERVISED MACHINE LEARNING ALOGORITHMS

Project Report

Submitted in partial fulfillment for the award of the degree of

MASTER OF COMPUTER APPLICATIONS (MCA)

Submitted By

Ms. GOKEDA LAHARI SAI DURGA

(Regd.No.22L31F0012)

Under the Guidance of

Guide: Mrs.G.Jyothi

Assistant Professor

DEPARPMENT OF MASTER OF COMPUTER APPLICATIONS

VIGNAN'S INSTITUTE OF INFORMATION TECHNOLOGY

(Autonomous)

Affiliated to JNTUGV, Vizianagaram & Approved by AICTE, New Delhi

Re-Accredited by NAAC (CGPA of 3.41/ 4.00)

ISO 9001:2008, ISO 14001:2004, OHSAS 18001:2007 Certified Institution

VISAKHAPATNAM – 530049

VIGNAN's INSTITUTE OF INFORMATION TECHNOLOGY
(AUTONOMOUS)
(Approved by AICTE-New Delhi & Affiliated to JNTU-GV, Vizianagaram)
Beside VSEZ, Duvvada, Vadlapudi Post, Gajuwaka, Visakhapatnam - 530 049.

CERTIFICATE

This is to certify that the project report entitled **"PREDICTION OF CARDIAC DISEASE USING SUPERVISED MACHINE LEARNING ALOGORITHMS"** is a bonafide record of project work carried out under my supervision by **GOKEDA LAHARI SAI DURGA (22L31F0012)** during the academic year 2023-24 in partial fulfilment of the requirements for the award of the degree of MASTER OF COMPUTER APPLICATIONS in VIGNAN'S INSTITUTE OF INFORMATION TECHNOLOGY (Autonomous). The results embodied in this project report have not been submitted to any other University or Institute for the award of any Degree or Diploma.

Signature of Project Guide
Mrs.G.Jyothi
Assistant Professor
Department of IT, VIIT

Head of the Department
Dr.G.Neelima
Associate Professor
Department of MCA,VIIT

EXTERNAL EXAMINER

VIGNAN's INSTITUTE OF INFORMATION TECHNOLOGY
(AUTONOMOUS)
(Approved by AICTE-New Delhi & Affiliated to JNTU-GV, Vizianagaram)
Beside VSEZ, Duvvada, Vadlapudi Post, Gajuwaka, Visakhapatnam - 530 049.

DECLARATION

We hereby declare that this project report entitled “**PREDICTION OF CARDIAC DISEASE USING SUPERVISEDMACHINE LEARNING ALOGORITHMS“** has undertaken by us for the fulfillment of Degree in Master of Computer Applications. We declare that this project report has not been submitted anywhere in the part of fulfillment for any degree of any other University.

PLACE: Visakhapatnam

DATE:

GOKEDA LAHARI SAI DURGA

(22L31F0012)

MASTER OF COMPUTER APPLICATIONS

Vision of the department:

➢ We aim to generate groomed, technical competent and skilled intellectual professionals.

➢ We serve as a valuable resource for modern industry and current society

Mission of the department:

➢ Providing strong theoretical and practical knowledge in computer science discipline with an emphasis on software development.

➢ To provide need-based quality training in the field of information technology.

➢ Impart quality education to meet global standards and achieve excellence in teaching-learning and research.

➢ To provide students with the tools to become productive, participating global citizens and life-long learners

MASTER OF COMPUTER APPLICATIONS

Vision of the Institute (VIIT):

We envision being recognized leader in technical education. We shall aim at national excellence by creating competent and socially conscious technical manpower for the current and future Industrial requirements and development of the nation.

Mission of the Institute (VIIT):

- Introducing innovative practices of Teaching and Learning.
- Undertaking research and development in thrust areas.
- Continuously collaborating with industry.
- Promoting a strong set of ethical values.
- Serving the surrounding region and nation at large.

VIGNAN's INSTITUTE OF INFORMATION TECHNOLOGY
(AUTONOMOUS)
(Approved by AICTE-New Delhi & Affiliated to JNTU-GV, Vizianagaram)
Beside VSEZ, Duvvada, Vadlapudi Post, Gajuwaka, Visakhapatnam - 530 049.

MASTER OF COMPUTER APPLICATIONS

PROGRAMOUTCOMES	
PO1	**Engineering Knowledge:** Apply the knowledge of mathematics science engineering fundamentals and mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems and engineering problems.
PO2	**Problem analysis:** Identify, formulate, review research Literature, and analyze complex engineering problems reaching substantiated conclusions using the first principles of mathematics,natural sciences, and engineering sciences
PO3	**Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for public health and safety, and the cultural societal, and environmental considerations
PO4	**Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments,analysis and interpretation of data, and synthesis of the information to provide valid conclusions
PO5	**Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
PO6	**The engineer and society:** Applyreasoninginformedbythecontextualknowledgetoassesssocietal,health,

	safety,legal and cultural issues and the consequent responsibilities legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice
PO7	**Environment and sustainability:** Understand the impact of professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development and need for sustainable development.
PO8	**Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
PO9	**Individual and teamwork:** Function effectively as an individual and as a member or leader in diverse teams and individual, and as a member or leader in diverse teams,and in multi disciplinary settings.
PO10	**Communication:** Communicate effectively on complex engineering activities with engineering community and with society at large, such as being able to comprehend and write effective reports and design documentation, and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
PO11	**Project management and finance**: Demonstrate knowledge and understanding of the engineering and knowledge and understanding of the engineering and management principles and apply these to one's work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
PO12	**Life-long learning:** Recognize the need for and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

MASTER OF COMPUTER APPLICATIONS

Program Educational Objectives (PEOS):

The post graduations of Master of Computer Application will be able:

PEO1: To successful professional in industry, government sector, academia, research, Entrepreneurial pursuit and consulting firms.

PEO2: To contribute to society as broadly educated, expressive, ethical and responsible citizens with proven expertise.

PEO3: To thrive to pursue life-long learning to fulfill their goals.

Program Specific Outcomes (PSOs):

MCA program has been designed to prepare graduates for attaining the following program Specific outcomes:

PSO1: They can identity, critically analyze, formulate and develop computer applications.

PSO2: Function competently as an individual and as a leader in multidisciplinary projects.

ACKNOWLEDGEMENT:

An endeavour over a long period can be successfully with the advice and support of many well-wishers. I take this opportunity to express our gratitude and appreciation to all of them.

I express my sincere gratitude to my internal guide, **Mrs.G.Jyothi** for her encouragement and cooperation in completion of my project. I am very fortunate in getting the generous help and constant encouragement from her.

I would be very grateful to our project coordinator **Mrs.A.Sirisha, Assistant Professor** for the continuous monitoring of my project work.I truly appreciate for her time and effort spent helping me.

I would like to thank our Head of the Department, **Dr.G.Neelima, Associate Professor** and all other teaching and non-teaching staff of the department for their cooperation and guidance during my project.

I sincerely thank to **Dr. J.Sudhakar, Principal** of VIGNAN'S INSTITUTE OF INFORMATION TECHNOLOGY (A) for his inspiration to undergo this project.

I wanted to convey my sincere gratitude to **Dr. V. Madhusudhan Rao**, **Rector** of VIGNAN'S INSTITUTE OF INFORMATION TECHNOLOGY (A) for allocating the required resources and for the knowledge sharing during my project work.

I extended my grateful thanks to our honorable **Chairman Dr. L. Rathaiah** for giving me an opportunity to study in his esteemed institution.

GOKEDA LAHARI SAI DURGA
Regd.No.:22L31F0012

Abstract

Nowadays, there is a scarcity of cardiovascular specialists in many countries and the number of cases that are misdiagnosed is rising. With the use of digital health information, this problem can be solved by establishing an efficient and accurate early-stage heart disease prediction algorithm. Data mining and machine learning are used in the medical industry for detecting and predicting heart disease would be beneficial. In this paper several data mining algorithms like KNN, Logistic Regression, Random Forest, Adaboost etc. were compared by performance and accuracy to predict heart disease. The study includes a dataset with 13 different attributes like cholesterol, blood pressure, blood sugar etc. and feature importance scores were estimated for each attribute to find which attributes have more impact on heart disease prediction which can later be used during diagnosis. In this paper, firstly the data was preprocessed and later several algorithms were applied on the dataset and were compared using different performance measures. Based on the results, an optimal algorithm was selected to be used for heart disease prediction.

KEYWORDS: Cardiovascular disease, Feature importance score, data mining techniques, Confusion Matrix, Classification Report, KNN, Decision Tree, Logistic Regression, Random Forest, Adaboost, Multilayer Perceptron.

INDEX

LIST OF FIGURES

1.Introduction

A popular saying goes that we are living in an "information age". Terabytes of data are produced every day. Data mining is the process which turns a collection of data into knowledge. The health care industry generates a huge amount of data daily. However, most of it is not effectively used. Efficient tools to extract knowledge from these databases for clinical detection of diseases or other purposes are not much prevalent. The aim of this paper is to summarize some of the current research on predicting heart diseases using data mining techniques, analyses the various combinations of mining algorithms used and conclude which technique(s) are effective and efficient. Also, some future directions on prediction systems have been addressed. The essential pumping organ of the human body requires proper care. Unforeseen development may inculcate confinement of random disease With the help of proper datasets from the validated repository, and utilization of supervised techniques would facilitate monitoring otherwise prediction to cease any unfortunate occurrence. One such technique possibly market-favoured is the machine learning techniques.Machine learning is based on testing and training the data where the system takes data from experience, and once the model has been trained, the predictions are made on the test dataset. Supervised learning like Logistic regression, Naïve Bayes, Decision Tree, SVM Model, KNN and Random can be defined as learning in the presence of a teacher .Machine-learning techniques are useful for vast data to be incorporated into the development of robust predictive analytics, often without restrictions of standard modelling techniques .For the proposed work in this paper, the dataset is split in the ratio of 7:3(train: test) where the 'train' is the teacher for prediction on the 'test' .In recent times, those techniques aid in making medical aid software for the early diagnosis. The risk of a fatality can be reduced at the primary stage by identifying any heart-related illness. For the implemented project as discussed further in the paper, the biological parameters used are – age, cholesterol and blood pressure, chest pain type, sex, exercise induced angina, and others. On their basis, the comparison has been performed in the terms of accuracy and other performance metrics of the algorithms used.

The suggested Records of large set of medical data created by medical experts are available for analyzing and extracting valuable knowledge from it. Data mining techniques are the means of extracting valuable and hidden information from the large amount of data available. Mostly the medical database consists of discrete information. Hence, decision making using discrete data becomes complex and tough task. Machine Learning (ML) which is subfield of data mining handles large scale well-formatted dataset efficiently. In the medical field, machine learning can be used for diagnosis, detection and prediction of various diseases. The main goal of this paper is to provide a tool for doctors to detect heart disease as early stage . This in turn will help to provide effective treatment to patients and avoid severe consequences. ML plays a very important role to detect the hidden discrete patterns and thereby analyze the given data. After analysis of data ML techniques help in heart disease prediction and early diagnosis. This paper presents performance analysis of various ML techniques such as Naive Bayes, Decision Tree, Logistic Regression and Random Forest for predicting heart disease at an early stage.

1.1 MOTIVATION

The main challenge in today's healthcare is provision of best quality services and effective accurate diagnosis. Even if heart diseases are found as the prime source of death in the world in recent years, they are also the ones that can be controlled and managed effectively. The whole accuracy in management of a disease lies on the proper time of detection of that disease. The proposed work makes an attempt to detect these heart diseases at early stage to avoid disastrous consequences.

2.Literature survery

To design a model for heart disease prediction, researchers used various methods of data mining include association rules, categorization, and clustering.

S.K. Dehkordi et al. proposed "Prediction of disease based on prescription using data mining methods". This paper deals with an algorithm called skating which is similar to boosting and bagging to improve the system's accuracy later it was compared with algorithms like Naïve bayes and KNN to show the difference in accuracy thus proving it a good approach.

M.S. Khalid et al. proposed "Ensemble approach for developing a smart heart disease prediction system using classification algorithms". This paper discusses a sample smart heart disease forecast model based on an ensemble technique and uses SVM, Randomforest trees, Naive Bayesian model, neural networks, and LR analysis-based classifiers.

C.B.C Latha et al. proposed "Enhancing the accuracy of prediction of heart disease risk based on ensemble classification techniques". In this study, the feature selection strategy is applied to improve the accuracy of the results. the classifier using Ensembled Majority Vote using RF,MP, Bayesian Networks, and Nave Bayes. M.

Tarawneh et al. proposed "Hybrid approach for predicting heart disease using data mining techniques". This paper deals with a dataset containing 12 features and a hybrid approach. The hybrid approach generated 89.62 accuracy compared to other algorithms like DT, KNN, J48, ANN and etc.

Chitra et al. proposed "prediction of heart disease system using supervised learning classifiers". This study investigates the use of a CNN classifier to increase the predictability of heart disease. A neural network employs a cascade design, in which the network is replenished one by one with neurons that have been cached and do not alter after being introduced to a concealed network. The suggested method's results .

Bardhwaj et al., (2017), Shailaja et al., (2018), Sun et al., (2019), and Lee & Yoon, (2017) studied a broad overview of machine learning techniques used in healthcare for various diseases.

They provided insights into the potential value of medical big data that can be used for clinical decision support, diagnostics, treatment decisions, fraud detection, and prevention. They briefly summarized the nine-step data mining process along with focusing on why efficient decision support was required by the healthcare system. The results from their experiment showed that machine learning models can be used for the early diagnosis of diseases. Their research is applicable to this project to an extent; however, their research is less focused on the diagnosis of heart diseases. Therefore, we move forward to review the literature that aligns with our project objective which is how machine learning algorithms can be used in the diagnosis of heart disease.

A comprehensive review by **Tripoliti et al**., (2017) focused on machine learning methodologies evaluating heart failure. They researched severity estimation of heart failure and the prediction of re-hospitalization, mortality, and destabilizations. They performed an extensive study on related works of heart failure.

A study by **J. & S**., (2019) used two supervised classifiers called Naïve Bayes Classifier and Decision Tree Classifiers to predict heart diseases on a dataset. 8 Their Decision Tree model predicted the heart disease patients with an accuracy of 91 percent and the Naïve Bayes Classifier had an accuracy of 87 percent.

A study by **Kamal kant et al**.(2014) proposed a model using the Naïve Bayes algorithm to predict heart diseases. The naïve Bayes algorithm is used to assign no dependency between the features. Their study concluded that the Naïve Bayes algorithm is the most effective for heart disease prediction after that Neural Networks and Decision Trees.

Nidhi Bhatla et al., (2012) used different data mining techniques to predict heart diseases. Their study revealed that the Neural Networks algorithm has performed with higher accuracy than Decision Trees. Their research project included two additional features such as obesity and smoking other than the common attributes.

The literature review reveals emerging and advanced machine learning and data mining algorithms involved in predicting heart diseases. It is evident from the above literature review that data mining algorithms have effectively predicted heart diseases. The trustworthiness of the model for predicting heart diseases with different risk factors is a high concern, however, SVM, Naïve Bayes, Decision Trees, Bagging and Boosting, and RandomForest have achieved reliable results in the diagnosis of heart disease (Jan et al., 2018). Numerous models using different algorithms have been proposed in the past, producing unique ways to talk about reliability and accuracy for heart disease. In the above literature review, many different data mining prediction models have been introduced such as SVM, Naïve Bayes, Decision Trees, Bagging and Boosting, and RandomForest for heart disease. The models using these algorithms to predict heart disease produced very high accuracy. Therefore, based on these data mining algorithms, we move forward with our research objective in this project to explore these machine learning algorithms and build an optimized model.

3.SYSTEM ANALYSIS

3.1 Existing Method

Clinical decisions are often made based on doctors' intuition and experience rather than on the knowledge rich data hidden in the database. This practice leads to unwanted biases, errors and excessive medical costs which affects the quality of service provided to patients. There are man y ways that a medical misdiagnosis can present itself. Whether adoctor is at fault, or hospital staff, a misdiagnosis of a serious illness can have very extremeand harmful effects. The National Patient Safety Foundation cites that 42% of medical patients feel they have had experienced a medical error or missed diagnosis. Patient safety issometimes negligently given the back seat for other concerns, such as the cost of medicaltests, drugs, and operations. Medical Misdiagnoses are a serious risk to our healthcare profession. If they continue, then people will fear going to the hospital for treatment. We can put an end to medical misdiagnosis by informing the public and filing claims and suits againstthe medical practitioners at fault.

Disadvantages:

- Prediction is not possible at early stages.
- In the Existing system, practical use of collected data is time consuming.
- Any faults occurred by the doctor or hospital staff n predicting would lead to fatalincidents.
- Highly expensive and laborious process needs to be performed before treating the patientto find out if he/she has any chances to get heart disease in future.

3.2 PROPOSED SYSTEM

This section depicts the overview of the proposed system and illustrates all of thecomponents, techniques and tools are used for developing the entire system. To develop anintelligent and user-friendly heart disease prediction system, an efficient software tool isneeded in order to train huge datasets and compare multiple machine learning algorithms.After choosing the robust algorithm with best accuracy and performance measures, it will beimplemented on the development of the and predicting heart disease risk level.Hardware components like Arduino/Raspberry Pi,differe nt biomedical sensors, display monitor, buzzer etc. are needed to build the continuous patient monitoring system.

Our proposed strategy focuses on a novel machine learning procedures for Heart disease (DD) classification and prediction, thus overcoming the existing problem. By utilizing Random Forest algorithms we will make our model in order to increase the performance and accuracy. According to literature - 3, by utilizing information mining methods An Intelligent Heart Disease Prediction System (IHDPS) is created, Sellappan Palaniappan proposed Neural

Network, and Decision Trees. To manufacture this strategy shrouded examples and connection between them is utilized. It is electronic, easy to use and expandable. To predict the heart attack disease.it helps in reducing treatment costs by providing effective treatments.To find the parameters values in prediction like accuracy,elapsed time and energy consumption.To overcome these difficulties proposed a prediction of cardiac disease using supervised learning algorithms. For this study, data set was obtained from Kaggle . There are 14 attributes and 1025 records in the data set.Table 1.3.1 illustrates all of the features in detail.

Serial Number	**Name of Attribute**	**Description**
1	age	Age
2	sex	female=0, male=1
3	trestbps	blood pressure
4	cp	Type of Chest pain
5	chol	Cholesterol level
6	Fbs	Fasting blood sugar > 120 mg/dl
7	Restecg	Electro cardiograph results
8	Thalach	Max heart rate
9	Exang	Gain induced due to exercise (1=true, 0=false)
10	Oldpeak	ST depression induced exercise
11	Slope	St segments peak slope
12	Ca	count of major vessels
13	Thal	1=normal 2= fixed defect 3= reversible defect
14	Target	0- not diseased 1- diseased person

Advantages

- Six supervised ML algorithms were compared for better analysis and understanding.
- Feature Importance scores were calculated which helps us to better take care of health.
- Prediction of heart disease could be useful for early diagnosis thus reducing mortality rate.
- Proves that simple supervised algorithms can show great impact in reality.

Disadvantages

- Ensemble approach can be used for improving accuracy.
- More number of records if taken may give better result or yield.

Block Diagram

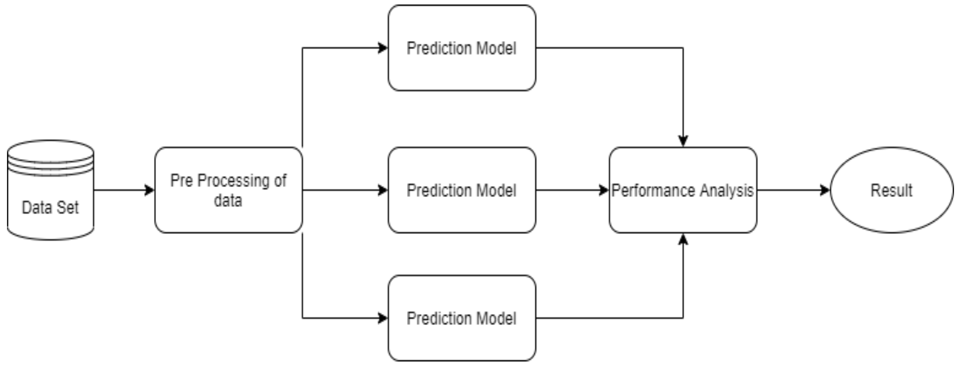


Fig3.2: Block Diagram

3.3 FEASIBILITY STUDY

A Feasibility Study is a preliminary study undertaken before the real work of a projectstarts to ascertain the likely hood of the projects success. It is an analysis of possiblealternative solutions to a problem and a recommendation on the best alternative.

Economic Feasibility

It is defined as the process of assessing the benefits and costs associated with thedevelopment of project. A proposed system, which is both operationally and technically feasible, must be a good investment for the organization. With the proposed system the usersare greatly benefited as the users can be able to detect the fake news from the real news andare aware of most real and most fake news published in the recent years. This proposedsystem does not need any additional software and high system configuration. Hence the proposed system is economically feasible.

Technical Feasibility

The technical feasibility infers whether the proposed system can be developedconsidering the technical issues like availability of the necessary technology, technicalcapacity, adequate response and extensibility. The project is decided to build using Python.Jupyter Note Book is designed for use in distributed environment of the internet and for the professional programmer it is easy to learn and use effectively. As the developingorganizati on has all the resources available to build the system therefore the proposed systemis technically feasible.

Operational Feasibility

Operational feasibility is defined as the process of assessing the degree to which a proposed system solves business problems or takes advantage of business opportunities. Thesystem is self-explanatory and doesn't need any extra sophisticated training. The system has built-in methods and classes which are required to produce the result. The application can behandled very easily with a novice user. The overall time that a user needs to get trained is 14less than one hour. As the software that is used for developing this application is veryeconomical and is readily available in the market. Therefore the proposed system isoperationally feasible.

TYPES OF CARDIOVASCULAR DISEASES

Heart diseases or cardiovascular diseases (CVD) are a class of diseases that involve the heart and blood vessels. Cardiovascular disease includes coronary artery diseases (CAD) like angina and myocardial infarction (commonly known as a heart attack). There is another heart disease, called coronary heart disease (CHD), in which a waxy substance called plaque develops inside the coronary arteries. These are the arteries which supply oxygen-rich blood to heart muscle. When plaque begins to build up in these arteries, the condition is called atherosclerosis. The development of plaque occurs over many years. With the passage of time, this plaque can harden or rupture (break open). Hardened plaque eventually narrows the coronary arteries which in turn reduces the flow of oxygen-rich blood to the heart. If this plaque ruptures, a blood clot can form on its surface. A large blood clot can most of the time completely block blood flow through a coronary artery. Over time, the ruptured plaque also hardens and narrows the coronary arteries. If the stopped blood flow isn't restored quickly, the section of heart muscle begins to die. Without quick treatment, a heart attack can lead to serious health problems and even death. Heart attack is a common cause of death worldwide. Some of the common symptoms of heart attack are as follows**.**

Chest pain

It is the most common symptom of heart attack. If someone has a blocked artery or is having a heart attack, he may feel pain, tightness pressure in the chest.

Nausea, Indigestion, Heartburn and Stomach Pain

These are some of the often over looked symptoms of heart attack.

Pain in the Arms

The pain often starts in the chest and then moves towards the arms, especially in the left side.

Fatigue

Simple chores which begin to set a feeling of tiredness should not be ignored.

Sweating

Some other cardiovascular diseases which are quite common are stroke, heart failure, hypertensive heart disease, rheumatic heart disease, Cardiomyopathy, Cardiac arrhythmia, Congenital heart disease, Valvular heart disease, Aortic aneurysms, Peripheral artery disease

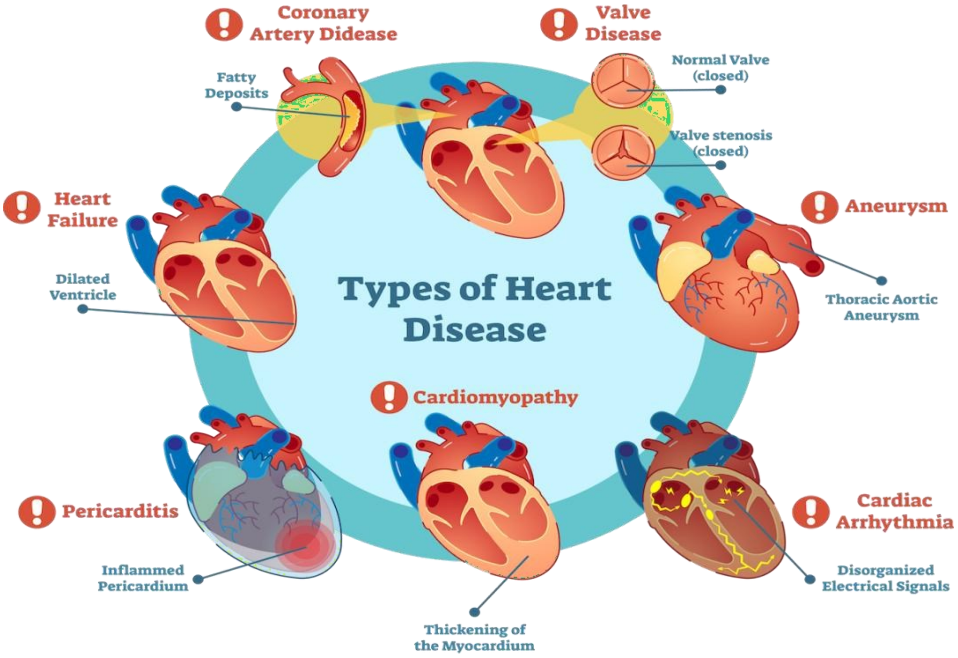


Fig 3.3: Types of Heart Diseases

Prevalence of Cardiovascular Diseases

An estimated 17.5 million deaths occur due to cardiovascular diseases worldwide. More than 75% deaths due to cardiovascular diseases occur in the middle-income and low- income countries. Also, 80% of the deaths that occur due to CVDs are because of stroke and heart attack. India too has a growing number of CVD patients added every year. Currently, the number of heart disease patients in India is more than 30 million. Over two lakh open heart surgeries are performed in India each year. A matter of growing concern is that the number of patients requiring coronary interventions has been rising at 20% to 30% for the past few years. The rest of the paper is organized as follows. Section 2 describes some of the well-known data mining algorithms used for heart disease prediction. Section 3 describes some of the popular data mining tools used for the data analysis purpose. Section 4 summarizes the methodologies and results of previous research on heart disease diagnosis and prediction. Section 5 discusses the pros and cons on literature survey. Finally, Section 6 concludes the paper along with future scope.

Supervised Machine Learning Algorithms

Different kinds of supervised machine learning algorithms were employed in this study. In supervised machine learning algorithms, the labeled training dataset is employed first of all to practice the fundamental algorithm. This qualified model is then loaded into a non- labeled testing research dataset to categorize it into related categories. A quick overview of these proposed supervised machine learning formulas for disease detection is given in the correspondingsubsection.

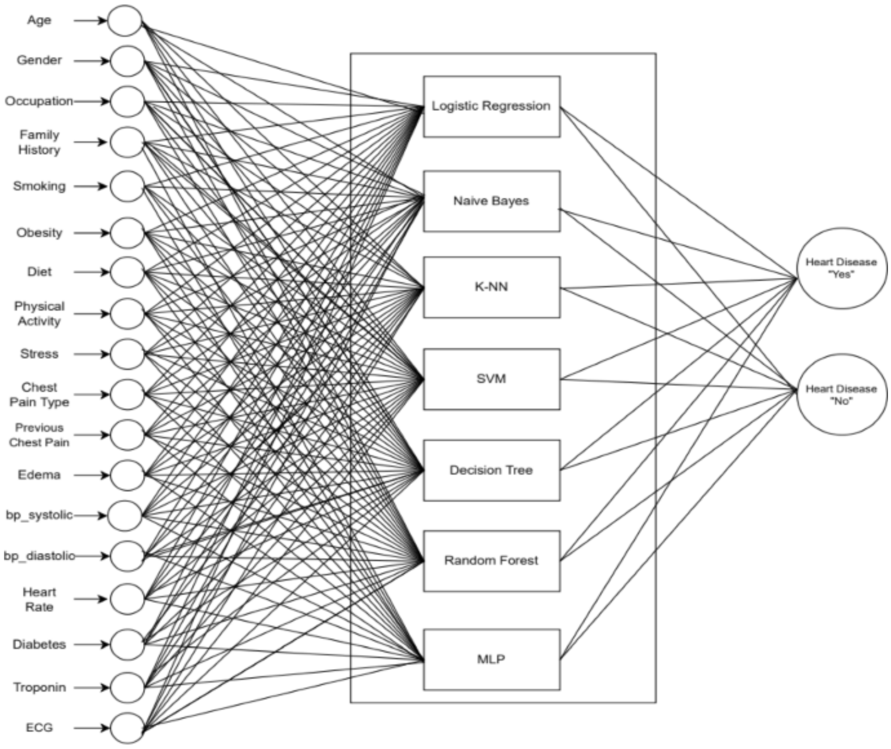


Fig 3.3.1: Supervised Machine Learning Algorithms

K-nearest neighbor (KNN)

KNN is one of the oldest and easiest classification algorithms [26,27] or statistical learning techniques [28]. K refers to the number of the nearest neighbors used, which can be defined directly in the object builder or simply calculated using the upper limit provided by the stated value [24]. Similar cases are thus subject to similar classifications [29] and a new instance is categorized by measuring its similarity to each of the current instances [30]. When an unidentified sample is received, the nearest neighbor algorithm will scan the pattern space for the k training samples next to the unfamiliar sample. From the test instance based on their distance, predictions from several neighbors can be calculated and two distinct methodologies are introduced to transform the distance into a weight [28,31]. The algorithm has a number of advantages such as it is analytically tractable and very easy to implement [28]. The classifier is very effective and performs well in disease prediction especially in HD prediction since it works with a single instance. In this study, the value of neighbors 2 and leaf size 40 were the best fit parameter for the dataset.

Random forest

RF is a method for classifying data by ensemble learning based on DT. It creates a large number of trees and also produces a forest of decision trees, while it is under the training stage. Every tree, a member of the forest, forecast class label for every single instance at the testing period. When a class label is predicted by each tree, then majority voting is used to decide the final decision for each test data. The class label that obtains the largest number of votes is considered as the most appropriate label applied to the test data. For every data in the data collection, this cycle is replicated. The best fit random state value for this study was 123, which gave the best performance for the applied dataset.

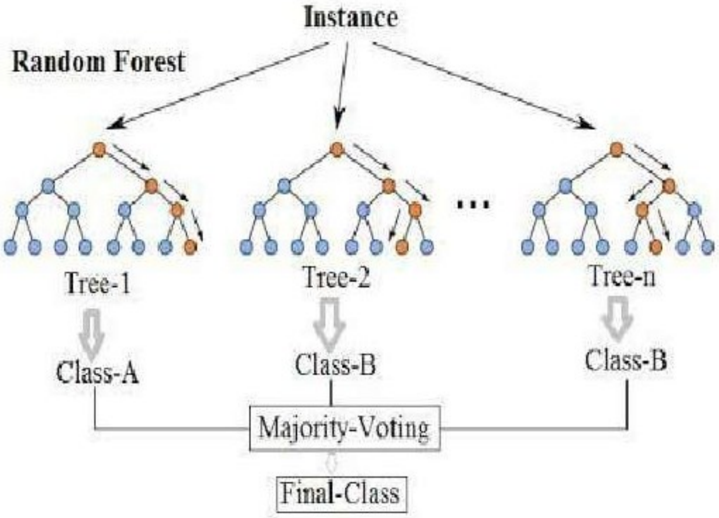


Fig 3.3.2: Random Forest

Decision tree (DT)

DT is one of the oldest and most common machine learning algorithms. A DT designs the logic of the decision in such a way that evaluates and matches results for the classification of data items into a structure as like a tree [25]. Usually a DT has multiple levels of nodes, the topmost level is known as root or parent node and others are child nodes. Evaluation of input variables or features is represented by all internal nodes that contain at least one child node. Depending on the evaluation outcome, the classification techniques branch to the correct child node, where the evaluation and branching process continues before the leaf node is reached [34]. The leaf or terminal nodes refer to the outcomes of the decision. DT is recognized easy to understand and learn and is a basic component of many protocols for medical diagnosis The maximum depth for this classification algorithm was defined 7 and the classifier by this maximum depth value produced the best result for the applied dataset in this study.

AdaboostM1 (ABM1)

ABM1 is one kind of ensemble learning based supervised machine learning classifier, which is widely used. It employs an adaptive enhancement approach and produces improved classification results by integrating multiple weak classifiers into a strong classifier [36]. In the initial stage, the same weight is allocated to all the observations. The weights of the observations change with the coefficient of weak classifiers, and the coefficient of the applied

classifiers is estimated using the value of the estimation error. So, the value of error generated by a massifier is considered as the coefficient of the classifier. Consequently, the weight of misclassified observations can be raised by the ABM1 algorithm and the weight of correctly identified observations can be reduced. In the subsequent iterations, it will enforce higher weight on the incorrectly classified observations more. Finally, all the weak classifiers developed are combined to form a stronger classifier using a linear combination method [37] to produce accurate classification performance. The value of estimators was defined as 200, with this classifier providing the best performance in this study.

Logistic regression (LR)

LR is a strong classifier among supervised machine learning algorithm's and is an extension of the general regression modeling applied to a dataset, represents the probability of occurrence or nonoccurrence of a particular instance. LR identifies the chances of a new observation belonging to a certain class, the result lying between 0 and 1 since it is a probability. Consequently, a threshold is assigned that defines this parathion into two classes to implement the LR as binary classification. For instance, a probability value calculated higher than 0.5 is designated as ‘class A’ otherwise ‘class B’. To design a categorical variable, which contains more than two values, the LR model can be generalized as a multinomial logistic regression. The best fit random state value 1234, and the best fit maximum iteration number 100 were found in this study for the applied dataset.

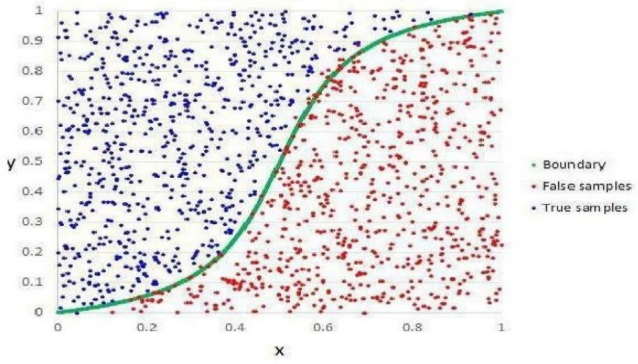


Fig 3.3.3: Logistic regression

Multilayer perceptron (MLP):

MLP is a well-established neural network-based classification algorithm, which consists of three or more types of layers: an input layer, output layer and one or more hidden layers between input and output layers. Every layer contains a number of ‘neurons’ connecting all the layers with each other. MLP is a universal multivariate non-linear mappings calculator that results from the capacity of training data to learn and generalize from training data using backpropagation learning methods. The construction of MLP classifiers consists of adequate input variables and specification of the type of network.

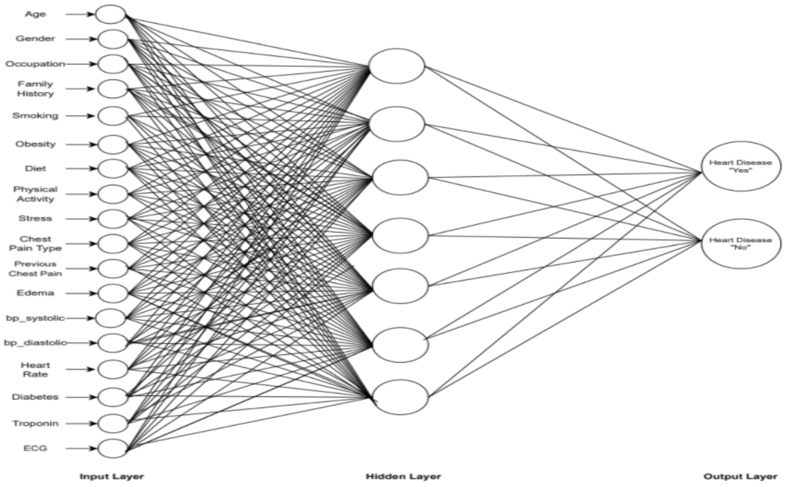


Fig 3.3.4: Multilayer perceptron

Data Mining Algorithms

Research on data mining has led to the formulation of several data mining algorithms. These algorithms can be directly used on a dataset for creating some models or to draw vital conclusions and inferences from that dataset. Some popular data mining algorithms are Decision tree, Naïve Bayes, k-means, artificial neural network etc. They are discussed in the follows section.

Decision Tree

A Decision tree is a decision support tool that uses a tree-like graph or model of decisions and their possible consequences including chance event outcomes and utility. It is one of the ways to display an algorithm. Decision trees are commonly used in operations research, specifically in decision analysis to help and identify a strategy that will most likely reach the goal. It is also a popular tool in machine learning. A Decision tree can easily be transformed to a set of rules by mapping from The root node to the leaf nodes one by one. Finally, by following these rules, appropriate conclusions can be reached.

K-means Algorithm

K-means creates k groups from a set of given objects so that the members of a group are more similar. Other than specifying the number of clusters, k-means also "learns" the clusters on its own without any information about which cluster a particular observation should belong to. That's why k-means can be called as semi-supervised learning method. K-means is especially effective over large datasets.

Support Vector Machine (SVM)

It is a supervised learning method which classifies data into two classes over a hyperplane. Support vector machine performs a similar task like C4.5 except that it doesn't use Decision trees at all. Support vector machine attempts to maximize the margin (distance between the hyper plane and the two closest data points from each respective class) to decrease any chance of misclassification. Some popular implementations of support vector machine are scikit-learn, MATLAB and of LIBSVM.

Naive Bayes (NB)

It is a simple technique for constructing classifiers. It is a probabilistic classifier based on Bayes' theorem. All Naive Bayes classifiers assume that the value of any particular feature is independent of the value of any other feature, given the class variable. Bayes theorem is given as follows: P(C|X) = P(X|C) * P(C)/P(X), where X is the data tuple and C is the class such that P(X) is constant for all classes. Though it assumes an unrealistic condition that attribute values are conditionally independent, it performs surprisingly well on large datasets where this condition is assumed and holds.

Artificial Neural Network (ANN)

An artificial neural network (ANN) is a computational model based on the structure and functions of biological neural networks. Information which flows through the network affects the structure of the artificial neural network because a neural network changes or learns in a sense-based on input and output, for that particular stage and consequently for each stage. ANN's are considered nonlinear statistical data modelling tools where the complex relationships between inputs and outputs are modelled or patterns are found .ANNs have layers that are interconnected. Artificial neural networks are fairly simple mathematical models to enhance existing data analysis technologies.

Random Forest

Random Forests are an ensemble learning method (also thought of as a form of nearest neighbor predictor) for classification and regression techniques. It constructs a number of Decision trees at training time and outputs the class that is the mode of the classes output by individual trees. It also tries to minimize the problems of high variance and high bias by averaging to find a natural balance between the two extremes. Both R and Python have robust packages to implement this algorithm.

Regression

Regression is a statistical concept which is used to determine the weight of relationship between one dependent variable (usually denoted by Y) and a series of other changing variables (known as independent variables). Two basic types of regression are linear regression and multiple linear regression. Also, there are several non-linear regression methods that are used for more complicated data analysis.

A-Priori Algorithms

It is an algorithm for frequent item set mining and association rule learning. A-priori uses breadth-first search algorithm and a hash structure to count candidate item sets efficiently. generates candidate item sets of length k from item sets of length k-1 .Then it prunes the candidates which have an infrequent sub pattern.

MATLAB

It is the short form for matrix laboratory. It supports a multi-paradigm numerical computing environment. It is a fourth-generation programming language. MATLAB provides matrix manipulations, plotting of functions and data, algorithm implementations, creation of user interfaces and interfacing with programs written in other languages including C, C++,C#, Java, Fortran and Python.

4. SYSTEM SPECIFICATION

4.1Functional Requirements

In this system have two actors are there admin and user Admin has to login using his username and password After admin login they can upload a dataset, they can train the machine using machine learning approach. In user part they have to register themselves. User has to login using user id and password Then user has to input patient data, and then based on trained model user input data will check and it will give the output.

4.2 Non-functional requirements

Nonfunctional necessities describe however a system should behave and establish constraints of its practicality. This type of requirements is also known as the system's quality attributes. Attributes such as performance, security, usability, compatibility are not the feature of the system, they are a required characteristic. They are "developing" properties that emerge from the whole arrangement and hence we can't compose a particular line of code to execute them. Any attributes required by the customer are described by the specification. We must include only those requirements that are appropriate for our project. Some Non-Functional Requirements are as follows:

1) Reliability

The structure must be reliable and strong in giving the functionalities. The movements must be made unmistakable by the structure when a customer has revealed a couple of enhancements. The progressions made by the Programmer must be Project pioneer and in addition the Test designer.

2) Maintainability

The system watching and upkeep should be fundamental and focus in its approach. There should not be an excess of occupations running on diverse machines such that it gets hard to screen whether the employments are running without lapses.

3) Performance

The framework will be utilized by numerous representatives all the while. Since the system will be encouraged on a single web server with a lone database server outside of anyone's ability to see, execution transforms into a significant concern. As[8]The structure should not capitulate when various customers would use everything the while. It should allow brisk accessibility to each and every piece of its customers. For instance, if two test specialists are all the while attempting to report the vicinity .

4.3 H/W requirements

Operating system	Windows 7 or 7+
RAM	8 GB
Hard disc or SSD	More than 500 G
Processor	Intel 3rd generation or high or Ryzen with 8 GB Ram

4.4 S/W requirements

Software’s	Python 3.6 or high version
IDE	Jupyter
Framework	Matplot ,imbalance,numpy and Scikit-Learn

5.SOFTWARE ENVIRONMENT

5.1 Introduction to Python

Python is an interpreted high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace. It p rovides constructs that enable clear programming on both
small and large scales. Pythonfeatures a dynamic type system and automatic memory management.It supports multiple programming paradigms, including object-oriented, imperative,functional and procedural, and has a large and comprehensive standard library. Pythoninterpreters are available for many operating systems. C Python, the referenceimplementation of Python, is open source software and has a community-based developmentmodel, as do nearly all of its variant implementations. C Python is managed by the non-profitPython Software Foundation.

Why Python?

- Python works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc).
- Python has a simple syntax similar to the English language.
- Python has syntax that allows developers to write programs with fewer lines than some other programming languages.
- Python can be treated in a procedural way, an object-orientated way or a functional way.

Managing Packages:

Pandas

Pandas is an open-source Python Library providing high-performance datamanipulation and analysis tool using its powerful data structures. The name Pandas is derivedfrom the word Panel Data an Econometrics from Multidimensional data.In 2008, developer Wes McKinney started developing pandas when in need of high performance, flexible tool for analysis of data. Prior to Pandas, Python was majorly used fordata mining and preparation. It had very little contribution towards data analysis. Pandassolved this problem.Using Pandas, we can accomplish five typical steps in the processing and analysis of data,regardless of the origin of data load, prepare, manipulate, model, and analyze. Python with Pandas is used in a wide range of fields including academic and commercial domainsincluding finance, economics, Statistics, analytics, etc.

Key Features of Pandas

-

- Fast and efficient Data Frame object with default and customized indexing.
- Tools for loading data into in-memory data objects from different file formats.
- Data alignment and integrated handling of missing data.
- Reshaping and pivoting of date sets.
- Label-based slicing, indexing and sub setting of large data sets.
- Columns from a data structure can be deleted or inserted.
- Group by data for aggregation and transformations.
- High performance merging and joining of data.
- Time Series functionality.

NumPy

NumPy is a general-purpose array-processing package. It provides a high- performance multidimensional array object, and tools for working with these arrays. It is thefundamental package for scientific computing with Python.It contains various features including these important ones:

- A powerful N-dimensional array object
- Sophisticated (broadcasting) functions
- Tools for integrating C/C++ and Fortran code
- Useful linear algebra, Fourier transform, and random number capabilities 24
- Besides its obvious scientific uses, NumPy can also be used as an efficient multi-dimensional container of generic data. Arbitrary data-types can be defined using Numpywhich allows NumPy to seamlessly and speedily integrate with a wide variety ofdatabases.

Sckit-Learn

- Simple and efficient tools for data mining and data analysis\
- Accessible to everybody, and reusable in various contexts
- Built on NumPy, SciPy, and matplotlib

- Open source, commercially usable - BSD license.

Matplot lib

- Matplotlib is a python library used to create 2D graphs and plots by using pythonscripts.
- It has a module named pyplot which makes things easy for plotting by providingfeature to control line styles, font properties, formatting axes etc.
- It supports a very wide variety of graphs and plots namely - histogram, bar charts, power spectra, error charts etc.

Jupyter Notebook

- The Jupyter Notebook is an incredibly powerful tool for interactively developing and presenting data science projects.
- The Jupyter Notebook is an open-source web application that allows you to createand share documents that contain live code, equations, visualizations and narrativetext.
- The Notebook has support for over 40 programming languages, including Python, R,Julia, and Scala.
- Notebooks can be shared with others using email, Drop box, Git Hub and the Jupyter Notebook.
- Your code can produce rich, interactive output: HTML, images, videos, LATEX, andcustom MIME types.
- Leverage big data tools, such as Apache Spark, from Python, R and Scala. Explorethat same data with pandas, scikit-learn, ggplot2, Tensor Flow.

Fig 5.2: Jupyter notebook

6. SYSTEM DESIGN

6.1 SYSTEM ARCHITECTURE

The below figure shows the process flow diagram or proposed work. First we collected theCleveland Heart Disease Database from UCI website then pre-processed the dataset andselect 16 important features.

For feature selection we used Recursive feature Elimination Algorithm using Chi2 methodand get 16 top features. After that applied ANN and Logistic algorithm individually andcompute the accuracy. Finally, we used proposed Ensemble Voting method and compute bestmethod for diagnosis of heart disease.

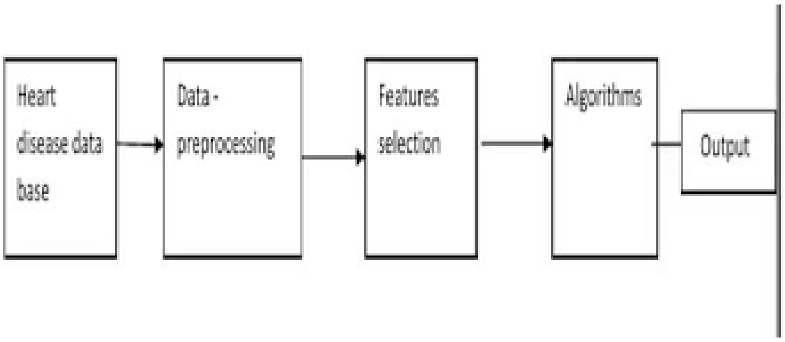


Fig 6.1 :SYSTEM ARCHITECTURE

6.2 MODULES

The entire work of this project is divided into 4 modules.They are:

a.Data Pre-processing

b.Feature

c.Classification

d.Prediction

a.Data Pre-processing:

This file contains all the pre-processing functions needed to process all input documents andtexts. First we read the train, test and validation data files then performed some preprocessinglike tokenizing, stemming etc. There are some exploratory data analysis is performed likeresponse variable distribution and data quality checks like null or missing values etc.

b.Feature:

Extraction In this file we have performed feature extraction and selection methods from sci-kit learn python libraries. For feature selection, we have used methods like simple bag-of-words and n-grams and then term frequency like tf-tdf weighting. We have also used AlgorithmsFeaturesselectionData -preprocessingHeartdisease database word2vec and POS tagging to extract the features, though POS tagging and word2vec has not been used at this point in the project.

c.Classification:

Here we have built all the classifiers for the breast cancer diseases detection. The extractedfeatures are fed into different classifiers. We have used Naive-bayes, Logistic Regression,Linear SVM, Stochastic gradient decent and Random forest classifiers from sklearn. Each ofthe extracted features was used in all of the classifiers. Once fitting the model, we comparedthe f1 score and checked the confusion matrix.After fitting all the classifiers, 2 best performing models were selected as candidatemodels for heart diseases classification. We have performed parameter tuning byimplementing GridSearchCV methods on these candidate models and chosen best performing parameters for these classifier.Finally selected model was used for heart disease detection with the probability oftruth. In Addition to this, we have also extracted the top 50 features from our term-frequencytfidf Vectorizer to see what words are most and important in each of the classes.We have also used Precision-Recall and learning curves to see how training and testset performs when we increase the amount of data in our classifiers.

d.Prediction:

Our finally selected and best performing classifier was algorithm which was thensaved on disk with name final_model.sav. Once you close this repository, this model will becopied to user's machine and will be used by prediction.py file to classify the Heart diseases. It takes a news article as input from user then model is used for final classification outputthat is shown to user along with probability of truth.

6.3DATA FLOW DIAGRAM

The data flow diagram (DFD) is one of the most important tools used by system analysis.Data flow diagrams are made up of number of symbols, which represents system components.Most data flow modeling methods use four kinds of symbols: Processes, Data stores, Dataflows and external entities.These symbols are used to represent four kinds of system components. Circles in DFDrepresent processes. Data Flow represented by a thin line in the DFD and each data store hasa unique name and square or rectangle represents external entities.

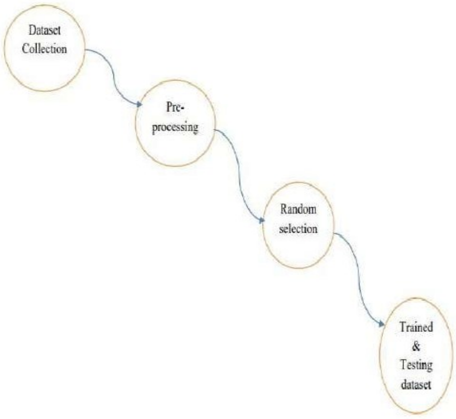


Fig 6.3: Data Flow diagram Level

UML DIAGRAMS

Use-Case Diagram

A use case diagram is a diagram that shows a set of use cases and actors and theirrelationships. A use case diagram is just a special kind of diagram and shares the samecommon properties as do all other diagrams,i.e a name and graphical contents that are a projection into a model. What distinguishes a use case diagram from all other kinds ofdiag rams is its particular content.

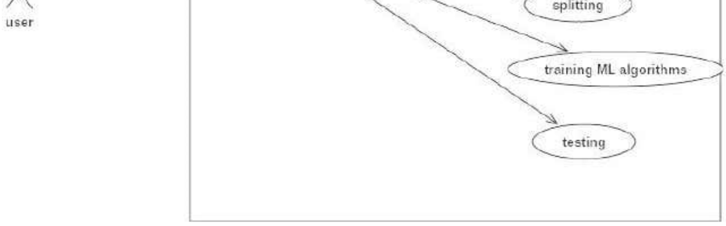


Fig6.3.1: Use case Diagram

Activity Diagram

An activity diagram shows the flow from activity to activity. An activity is an ongoing non-atomic execution within a state machine. An activity diagram is basically a projection of theelements found in an activity graph, a special case of a state machine in which all or moststates are activity states and in which all or most transitions are triggered by completion ofactivities in the source.

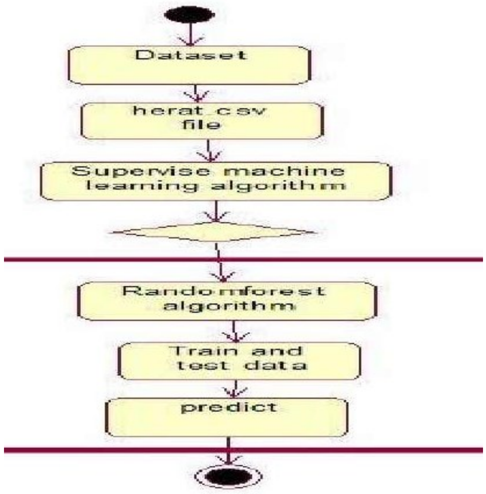


Fig6.3.2:activity diagram

Sequence Diagram

A sequence diagram is an interaction diagram that emphasizes the time ordering of messages.A sequence diagram shows a set of objects and the messages sent and received by thoseobjects. The objects are typically named or anonymous instances of classes, but may alsorepresent instances of other things, such as collaborations, components, and nodes. We usesequence diagrams to illustrate the dynamic view of a system.

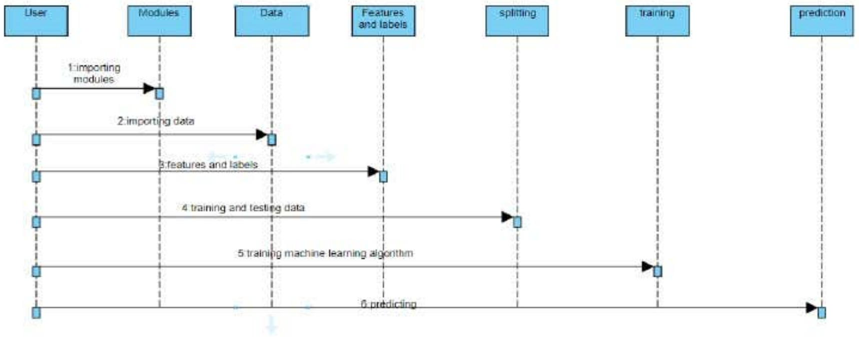


Fig 6.3.3: Sequence Diagram

Class diagram

A Class diagram in the Unified Modeling Language (UML)is a type of static structurediagram that describes the structure of a system by showing the system's classes, theirattributes, operations (or methods), and the relationships among objects.

It provides a basicnotation for other structure diagrams prescribed by UML. It is helpful for developers andother team members too.

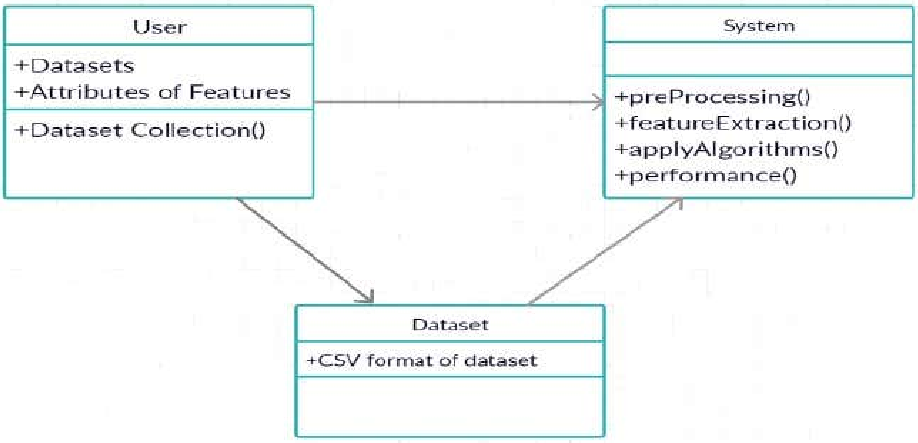


Fig 6.3.4. Class Diagram

7.SYSTEM IMPLEMENTATION

7.1 METHODOLOGY:

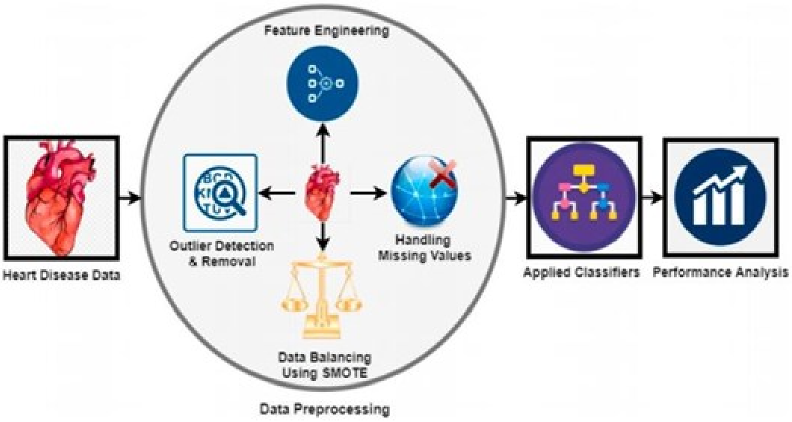


Fig 7.1 Experimental Methodology

Performance Evaluation Metrics

Six (06) classification algorithms were applied to the dataset to find the best performer algorithm comparing the accuracy and other statistical variables by 10-fold cross-validation. The algorithms applied were multilayer perceptron (MP), K-nearest neighbours (KNN), random forest (RF), decision tree (DT), logistic regression (LR) and AdaboostM1 (ABM1). These algorithms were compared based on their performance evaluation metrics.

A brief overview of these performance evaluations is described in this subsection. A confusion matrix was obtained to calculate the sensitivity, specificity and accuracy of the result for each algorithm. The below mentioned formulas were used to calculate all the parameters [11,13]:

Sensitivity = TP/TP + FN (3)
Specificity = TN/TN + FP (4)

Specificity = (TP + TN)/ (TP + FP + TN + FN) (5)
TPR = TP (TP + FN) (6)

FPR = FP/ (FP + TN) (7)

Here, TP and TN represent true positive and true negative respectively and FP and FN demonstrate false positive and false negative; TPR represents the true positive rate and FPR the false positive rate. Sensitivity relates to the percentage of actual positives that the classifier accurately defines as data and reflects the number of positive predictions that the classifier correctly identifies [20]. Specificity is the ability of the classifier to correctly distinguish negative outcomes [20]. Accuracy is the percentage of correctly classified instances by a classifier [11,13,20]. Different statistical values were used to compare the efficiency of different algorithms such as kappa statistics, precision, recall, f-measure, Matthew's correlation coefficient (MCC), receiver operating characteristic (ROC) and precision-recall (PRC). Kappa statistics estimate interrater agreement from identified and expected accuracy for qualitative attributes [21].

Precision is a valid evaluation metric particularly when the proposed ML model is required to validate based on the predicted and actual result [20,21]. It calculates the percentage of expected positive's that are actual positives. As a result, it is reliant on TP and FP values. When it is required to determine the number of positives that may fairly be predicted, recall is another useful evaluation metric [20,21], representing the proportion of positives successfully categorized. Recall is measured using TP and FP values. F-Measure maintains a balance between precision and recall for a classifier. The F-Measure score is a number between 0 and 1 that represents the statistically significant measures of precision and recall [20,21].

In machine learning, the MCC is used to assess the validity of binary and multiclass classifications. It accounts for true and false positives and negatives and is often recognized as a balanced metric that may be applied even when the classes are of considerably differents sizes. The MCC is essentially a correlation coefficient number ranging from −1 to +1 [22]. These parameters estimate their values using the following equations [22–24]:

Kappa, $k = PrPr\ (a) - Pr(e)/1 - Pr(e)$ (8)

$\text{Precision} = TP/TP+FP$(9)

$\text{Recall} = TP/TP+FN$(10)

$\text{F-measure}=2*\text{precision recall}\ /\text{precision}+\text{recall}$(11)

$MCC = TP*TN - FP*FN\ (TP + FP)\ (TP + FN)\ (TN + FP)\ (TN + FN)\ \sqrt{}$ (12)

ROC is used to determine how much a model is capable of distinguishing classes. PRC is the ratio of precision and recall. In this study, K-fold cross-validation was used to train and test the model. In this approach, the data set is divided into a number of groups. K refers to the number of groups, also known as 'fold'. At the same time, cross-validation is an approach to evaluate a machine learning model. K-fold cross-validation is such a technique, where the data set is split into k number of groups and the model is trained by (k-1) groups and the other group participates to test or evaluate the trained model. In this approach, the model is trained k number of times and each time, different fold participates to evaluate the model. It indicates that each fold participates to train and test a model in K-fold cross-validation.

Fig. 2 represents a 5-fold cross-validation approach. The figure depicts that the dataset is split into 5 folds or groups, where 4 groups participate in model training and another one-fold participates to evaluate the training in each iteration. In our study, 10-fold cross-validation

is employed. Cross-validation is performed to protect against overfitting in a predictive model.

Classifier	Accuracy	Sensitivity	Specificity	Precision	Recall	Kappa	FMeasure	MCC
LR	87.36	0.82	0.91	0.87	0.88	0.742	0.87	0.744
ABM1	74.71	0.63	0.97	0.80	0.75	0.518	0.75	0.576
MLP	92.52	0.86	0.98	0.92	0.92	0.848	0.92	0.853
KNN	100	1	1	1	1	1	1	1
DT	97	0.92	1	1	1	1	1	1
RF	100	1	1	1	1	1	1	1

Feature	**LR**	**ABM1**	**DT**	**RF**
Age	0.044	0.176	0.102	0.103
Sex	-0.921	0.027	0.024	0.045
Cp	0.929	0.034	0.104	0.140
Trestbps	-0.328	0.120	0.038	0.072
Chol	-0.408	0.171	0.167	0.076
Fb	0.099	0.00	0.000	0.012
Restcg	0.075	0.00	0.007	0.016
Thalach	0.540	0.161	0.087	0.105
Exang	-0.336	0.005	0.016	0.037
Oldpeak	-0.401	0.155	0.109	0.099
Slope	0.589	0.018	0.000	0.054
Ca	-1.208	0.054	0.112	0.103
Thal	-0.733	0.074	0.229	0.132

Also, Feature importance scores for all the features were calculated with respect to algorithms like LR, ABM1, DT and RF. These scores are important as they tell which feature has more impact on result and thuscan be used for better diagnosis. The below table consists of feature importance scores of all attributes.

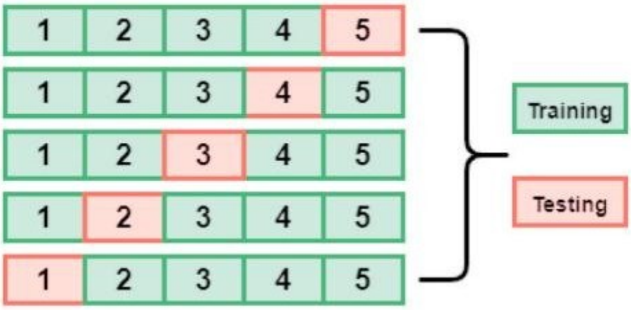


Fig. 2. Graphical representation of K-fold cross-validation.

8.SYSTEM TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the

Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

8.1Testing Method

Unit testing

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

Integration testing

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

Functional test

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

Functional testing is centered on the following items:

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

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

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures

interfacing systems or procedures must be invoked.

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

8.2Test cases

SYSTEM TEST

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

White Box Testing

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

Black Box Testing

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

Unit Testing

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

Test strategy and approach

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

Test objectives

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

Features to be tested

- Verify that the entries are of the correct format
- No duplicate entries should be allowed
- All links should take the user to the correct page.

Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

Test Results:

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

Acceptance Testing

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

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

SAMPLE CODE

```
 import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
from imblearn.over_sampling import SMOTE
from sklearn.linear_model import LogisticRegression
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import classification_report,cohen_kappa_score,matthews_corrcoef
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import confusion_matrix
from sklearn.tree import DecisionTreeClassifier
from sklearn.neural_network import MLPClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.ensemble import AdaBoostClassifier
from sklearn.metrics import roc_curve,precision_recall_curve
from sklearn.metrics import matthews_corrcoef
from sklearn.metrics import roc_curve
import matplotlib.pyplot as plt3
from sklearn.metrics import cohen_kappa_score
```

DATA ANALYSIS

```
df = pd.read_csv('C:\\Users\\LAHARI\\OneDrive\\Desktop\\Dataset.csv')
#reading dataset using pandas
#Df
#showing details
df.info()
#to find count of null values
df.isnull().sum()
```

```
#size of data
df.shape
#count of male and female
df.sex.value_counts()
#count of diseased and non-diseased
df.target.value_counts()
pd.crosstab(df.target,df.sex)
#standard measures for every row
df.describe().T
```

DATA PREPROCESSING

```
#Box Plot for Finding Outliers
col1=[df['restecg'],df['oldpeak'],df['slope'],df['ca']]
fig,ax=plt.subplots()
ax.boxplot(col1,patch_artist=True,vert=False)
plt.yticks([1,2,3,4],['restecg','oldpeak','slope','ca'])
plt.show()
col2=[df['thalach'],df['chol'],df['trestbps'],df['age']]
fig,ax=plt.subplots()
ax.boxplot(col2,patch_artist=True,vert=False)
plt.yticks([1,2,3,4],['thalach','chol','trestbps','age'])
plt.show()
```

IQR FILTER

```
def removeOutliers(data, col):
    q3=data[col].quantile(0.75)
    q1=data[col].quantile(0.25)
    IQR=q3-q1
    Lowr=q1-(1.5*IQR)
    upr=q3+(1.5*IQR)
    return upr,Lowr
u1, l1=removeOutliers(df,"chol")
print(u1,l1)
```

```
df=df[(df['chol']>l1)&(df['chol']<u1)]
print("after change:",df.shape)
u2,l2=removeOutliers(df,"trestbps")
print(u2,l2)
df=df[(df['trestbps']>l2)&(df["trestbps"]<u2)]
print(df.shape)
u3,l3=removeOutliers(df,'thalach')
print(u3,l3)
df=df[(df['thalach']>l3)&(df['thalach']<u3)]
print(df.shape)
u4,l4=removeOutliers(df,'ca')
print(u4,l4)
df=df[(df['ca']>l4)&(df['ca']<u4)]
print(df.shape)
u5,l5=removeOutliers(df,'oldpeak')
print(u5,l5)
df=df[(df['oldpeak']>l5)&(df['oldpeak']<u5)]
print(df.shape)
col2=[df['thalach'],df['chol'],df['trestbps'],df['age']]
fig,ax=plt.subplots()
ax.boxplot(col2,patch_artist=True,vert=False)
plt.yticks([1,2,3,4],['thalach','chol','trestbps','age'])
plt.show()
col1=[df['restecg'],df['oldpeak'],df['slope'],df['ca']]
fig,ax=plt.subplots()
ax.boxplot(col1,patch_artist=True,vert=False)
plt.yticks([1,2,3,4],['restecg','oldpeak','slope','ca'])
plt.show()
df.shape
```

SMOTE

```
x=df.drop("target",axis=1).values
```

```
y=df["target"].values
X_train, X_test, y_train, y_test = train_test_split(x, y, test_size = 0.2, random_state = 0)
X_train=StandardScaler().fit_transform(X_train)
X_test=StandardScaler().fit_transform(X_test)
print("Number of transactions in X_train dataset: ", X_train.shape)
print("Number of transactions in y_train dataset: ", y_train.shape)
print("Number of transactions in X_test dataset: ", X_test.shape)
print("Number of transactions in y_test dataset: ", y_test.shape)
sam1 = LogisticRegression()
sam1.fit(X_train, y_train)
sam1pred = sam1.predict(X_test)
print(classification_report(y_test, sam1pred))
bos=sum(y_train==1)
print("Before OverSampling, counts of label '1':", bos)
aos= sum(y_train == 0)
print("Before OverSampling, counts of label '0':",aos)
smt = SMOTE(random_state = 2)
X_train_res, y_train_res = smt.fit_resample(X_train, y_train)
print("\nAfter OverSampling, counts of label '1': ",bos)
print("After OverSampling, counts of label '0': ",aos)
logre1 = LogisticRegression()
logre1.fit(X_train_res, y_train_res)
sample = logre1.predict(X_test)
print(classification_report(y_test, sample))
sns.heatmap(df.corr(),annot=True,annot_kws={'size':10},linewidth=0,linecolor="white")
sns.set(rc={"figure.figsize":(10,5)})
fig1=sns.FacetGrid(df,hue="target")
fig1.map(sns.kdeplot,'age',fill=True)
fig1.set(xlim=(20,80))
plt.legend(labels=['Diseased','Non-Diseased'])
fig1.set(ylabel=('Density'))
```

```
warnings.filterwarnings("ignore")
```

FEATURE IMPORTANCE SCORE

```
logr = LogisticRegression()
logr.fit(X_train, y_train)
importance = logr.coef_[0]
x = ['restecg', 'sex', 'cp', 'trestbps', 'chol', 'thalach', 'exang', 'oldpeak', 'age', 'fbs', 'slope', 'ca', 'thal']
for i, v in enumerate(importance):
   print("Feature: %s, Score: %.5f" % (x[i], v))
plt.bar(x, importance)
plt.show()
dt = DecisionTreeClassifier()
dt.fit(X_train, y_train)
imp2 = dt.feature_importances_
x = ['thalach', 'exang', 'trestbps', 'chol', 'fbs', 'restecg', 'oldpeak', 'age', 'sex', 'cp', 'slope', 'ca', 'thal']
for a, b in enumerate(imp2):
   print("Feature: %s, Score: %.5f" % (x[a], b))
plt.bar(x, imp2)
plt.show()

rf = RandomForestClassifier()
rf.fit(X_train, y_train)
imp3 = rf.feature_importances_
x = ['slope', 'ca', 'age', 'trestbps', 'chol', 'fbs', 'sex', 'cp', 'restecg', 'thalach', 'exang', 'oldpeak', 'thal']
for x_idx, y in enumerate(imp3):
   print("Feature: %s, Score: %.5f" % (x[x_idx], y))

plt.bar(x, imp3)
plt.show()
classifier = RandomForestClassifier()
classifier.fit(X_train, y_train)
imp4 = classifier.feature_importances_
```

```
for p, q in enumerate(imp4):
    print("Feature: %s, Score: %.5f" % (x[p], q))

plt.bar(x, imp4)
plt.show()
```

ANALYSIS

```
logistic_regression_model = LogisticRegression()
logistic_regression_model.fit(X_train, y_train)
# Make predictions on the test set
logr_pred = logistic_regression_model.predict(X_test)
lr_fpr,lr_tpr,lr_th =roc_curve(y_test,logr_pred)
random_forest_model = RandomForestClassifier()
random_forest_model.fit(X_train, y_train)

# Make predictions on the test set
rf_pred = random_forest_model.predict(X_test)
rf_fpr1,rf_tpr1,rf_th = roc_curve(y_test,rf_pred)
adaboost_model = AdaBoostClassifier()
adaboost_model.fit(X_train, y_train)

# Make predictions on the test set
adb_pred = adaboost_model.predict(X_test)
adb_fpr2,adb_tpr2,adb_th =roc_curve(y_test,adb_pred)
knn_model = KNeighborsClassifier()
knn_model.fit(X_train, y_train)

# Make predictions on the test set
knn_pred = knn_model.predict(X_test)
knn_fpr3,knn_tpr3,knn_th = roc_curve(y_test,knn_pred)
decision_tree_model = DecisionTreeClassifier()
decision_tree_model.fit(X_train, y_train)
```

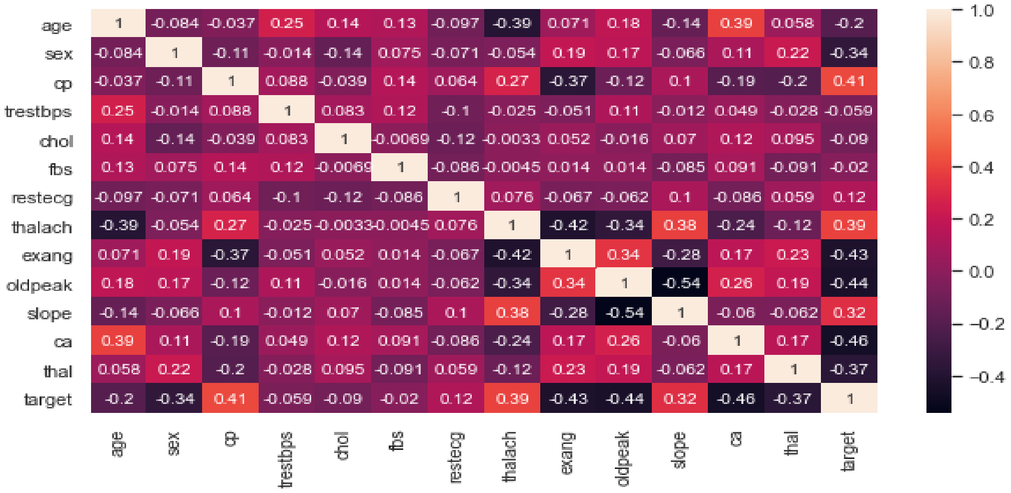
```
# Make predictions on the test set
dt_pred = decision_tree_model.predict(X_test)
dt_fpr4,dt_tpr4,dt_th = roc_curve(y_test,dt_pred)
mlp_model = MLPClassifier()
mlp_model.fit(X_train, y_train)

# Make predictions on the test set
mlp_pred = mlp_model.predict(X_test)
mlp_fpr5,mlp_tpr5,mlp_th =roc_curve(y_test,mlp_pred)

plt.figure(figsize=(10,5))
plt1= plt.subplot()
plt1.plot(lr_fpr,lr_tpr,label='Logistic Regression')
plt1.plot(rf_fpr1,rf_tpr1,label='Random Forest')
plt1.plot(adb_fpr2,adb_tpr2,label='AdaBoost Classifier')
plt1.plot(knn_fpr3,knn_tpr3,label='K-Nearest Neighbor')
plt1.plot(dt_fpr4,dt_tpr4,label='Desion Tree')
plt1.plot(mlp_fpr5,mlp_tpr5,label='Multilayer Perceptron')
plt1.plot([0,1],[0,1],ls='--',color='grey')
plt1.set_ylabel('True positive rate')
plt1.set_xlabel('False positive rate')
plt1.set_title('ROC Curve')
plt1.legend()
plt.grid(True)
plt.show()

logistic_regression_model = LogisticRegression()
logistic_regression_model.fit(X_train, y_train)

# Make predictions on the test set
lr_pred = logistic_regression_model.predict(X_test)
```

```
lr_prec,lr_rec,lr_thld = precision_recall_curve(y_test,lr_pred)
rf_prec,rf_rec,rf_thld = precision_recall_curve(y_test,rf_pred)
adb_prec,adb_rec,adb_thld = precision_recall_curve(y_test,adb_pred)
knn_prec,knn_rec,knn_thrld = precision_recall_curve(y_test,knn_pred)
dt_prec,dt_rec,dt_thld = precision_recall_curve(y_test,dt_pred)
mlp_prec,mlp_rec,mlp_thld = precision_recall_curve(y_test,mlp_pred)
#sns.set_style('whitegrid')
plt.figure(figsize=(10,5))
plt3.title('Precision Recall cuurve')
plt3.plot(lr_prec,lr_rec,label='Logistic Regression')
plt3.plot(rf_prec,rf_rec,label='Random Forest')
plt3.plot(adb_prec,adb_rec,label='AdaBoost Classifier')
plt3.plot(knn_prec,knn_rec,label='K-Nearest Neighbor')
plt3.plot(dt_prec,dt_rec,label='Desion Tree')
plt3.plot(mlp_prec,mlp_rec,label='Multilayer Perceptron')
plt3.plot([0,1],[0,1],ls='--')
plt3.ylabel('Precision')
plt3.xlabel('Recall')
plt3.legend()
plt3.show()
df
df.to_csv('modified1.csv')
# Assuming you have the ground truth labels y_test and predictions lr_pred, knn_pred, etc.
lr_kappa = cohen_kappa_score(y_test, lr_pred)
knn_kappa = cohen_kappa_score(y_test, knn_pred)
dt_kappa = cohen_kappa_score(y_test, dt_pred)
rf_kappa = cohen_kappa_score(y_test, rf_pred)
adb_kappa = cohen_kappa_score(y_test, adb_pred)
mlp_kappa = cohen_kappa_score(y_test, mlp_pred)
y=[lr_kappa,knn_kappa,dt_kappa,rf_kappa,adb_kappa,mlp_kappa]
x=['logistic regression','KNN','Decision Tree','Random Forest','Adaboost','MLP']
```

```
plt.plot(x,y)
plt.ylabel('Kappa Values',size=15)
plt.title("Kappa Metric for various Classifiers",size=15)
#plt.rcParams['figure.figsize']=[10,5]
plt.show()
lr_MCC = matthews_corrcoef(y_test, lr_pred)
knn_MCC = matthews_corrcoef(y_test, knn_pred)
dt_MCC = matthews_corrcoef(y_test, dt_pred)
rf_MCC = matthews_corrcoef(y_test, rf_pred)
adb_MCC = matthews_corrcoef(y_test, adb_pred)
mlp_MCC = matthews_corrcoef(y_test, mlp_pred)
y=[lr_MCC,knn_MCC,dt_MCC,rf_MCC,adb_MCC,mlp_MCC]
x=['logistic regression','KNN','Decision Tree','Random Forest','Adaboost','MLP']
plt.plot(x,y,color='green')
plt.ylabel('MCC Values',size=15)
plt.title("MCC Metric for various Classifiers",size=15)
plt.show()
y=[87.36,100.00,97.00,100.00,74.71,92.52]
x=['logistic regression','KNN','Decision Tree','Random Forest','Adaboost','MLP']
plt.bar(x,y,color=['grey'])
plt.ylabel('Accuracy',size=15)
plt.title("Accuracy for various Classifiers",size=15)
plt.show()
x=['logistic regression','Adaboost','MLP','KNN','Decision Tree','Random Forest']
y1=[0.87,0.80,0.82,1,1,0.99]
y2=[0.88,0.78,0.93,1,1,0.99]
y3=[0.87,0.75,0.92,1,1,0.99]
plt.plot(x,y1,label='Precision',color='red')
plt.plot(x,y2,label='Recall',color='green')
plt.plot(x,y3,label='F-Measure',color='blue')
plt.legend(loc='best')
plt.title("Classification Metrics",size=15)

plt.ylabel('Metrics')

plt.show()

plt.rcParams['figure.figsize']=[10,5]
```

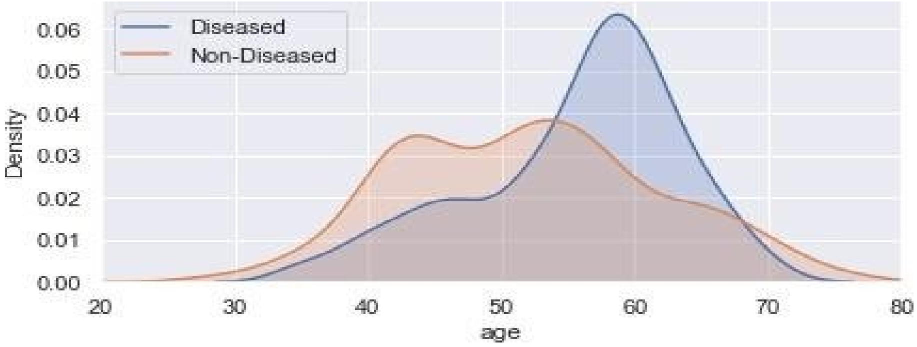
9.EXPERIMENTAL RESULTS

Heat Map for Correlation Matrix

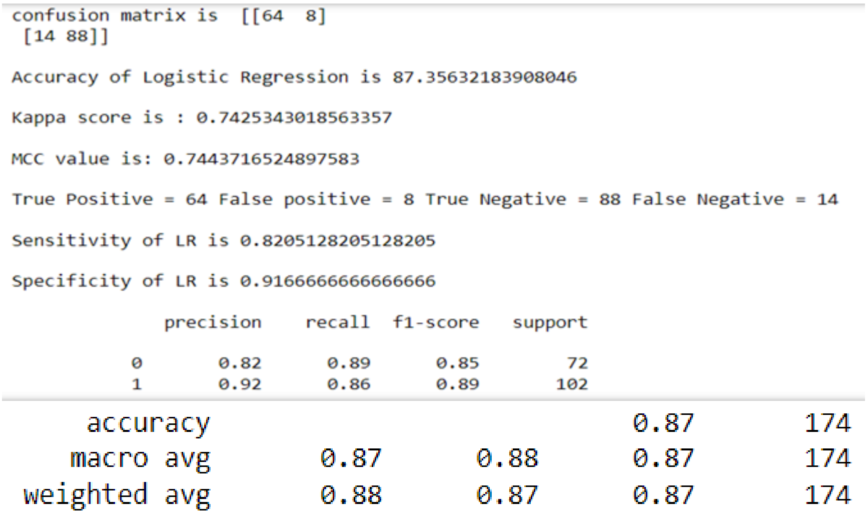


	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
age	1	-0.084	-0.037	0.25	0.14	0.13	-0.097	-0.39	0.071	0.18	-0.14	0.39	0.058	-0.2
sex	-0.084	1	-0.11	-0.014	-0.14	0.075	-0.071	-0.054	0.19	0.17	-0.066	0.11	0.22	-0.34
cp	-0.037	-0.11	1	0.088	-0.039	0.14	0.064	0.27	-0.37	-0.12	0.1	-0.19	-0.2	0.41
trestbps	0.25	-0.014	0.088	1	0.083	0.12	-0.1	-0.025	-0.051	0.11	-0.012	0.049	-0.028	-0.059
chol	0.14	-0.14	-0.039	0.083	1	-0.0069	-0.12	-0.0033	0.052	-0.016	0.07	0.12	0.095	-0.09
fbs	0.13	0.075	0.14	0.12	-0.0069	1	-0.086	-0.0045	0.014	0.014	-0.085	0.091	-0.091	-0.02
restecg	-0.097	-0.071	0.064	-0.1	-0.12	-0.086	1	0.076	-0.067	-0.062	0.1	-0.086	0.059	0.12
thalach	-0.39	-0.054	0.27	-0.025	-0.0033	-0.0045	0.076	1	-0.42	-0.34	0.38	-0.24	-0.12	0.39
exang	0.071	0.19	-0.37	-0.051	0.052	0.014	-0.067	-0.42	1	0.34	-0.28	0.17	0.23	-0.43
oldpeak	0.18	0.17	-0.12	0.11	-0.016	0.014	-0.062	-0.34	0.34	1	-0.54	0.26	0.19	-0.44
slope	-0.14	-0.066	0.1	-0.012	0.07	-0.085	0.1	0.38	-0.28	-0.54	1	-0.06	-0.062	0.32
ca	0.39	0.11	-0.19	0.049	0.12	0.091	-0.086	-0.24	0.17	0.26	-0.06	1	0.17	-0.46
thal	0.058	0.22	-0.2	-0.028	0.095	-0.091	0.059	-0.12	0.23	0.19	-0.062	0.17	1	-0.37
target	-0.2	-0.34	0.41	-0.059	-0.09	-0.02	0.12	0.39	-0.43	-0.44	0.32	-0.46	-0.37	1

Density plot for diseased and non-diseased according to age distribution

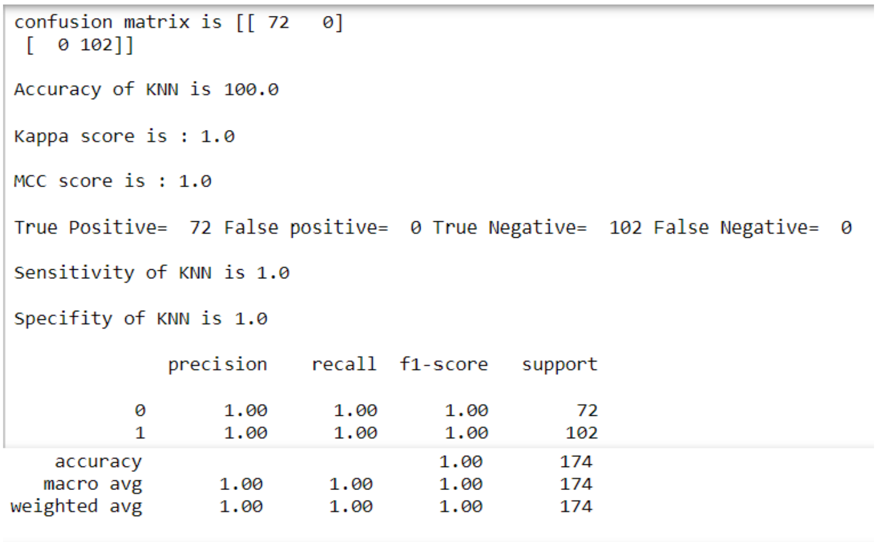


Result of Logistic Regression



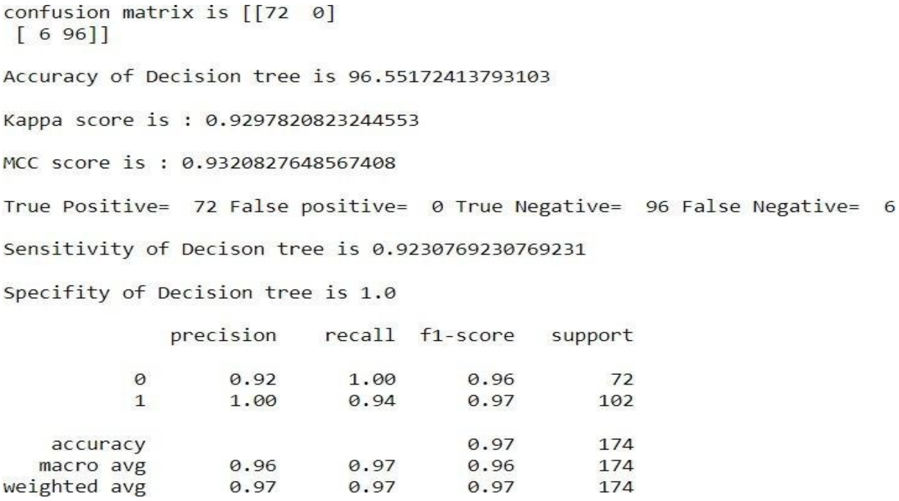
```
confusion matrix is  [[64  8]
 [14 88]]

Accuracy of Logistic Regression is 87.35632183908046

Kappa score is : 0.7425343018563357

MCC value is: 0.7443716524897583

True Positive = 64 False positive = 8 True Negative = 88 False Negative = 14

Sensitivity of LR is 0.8205128205128205

Specificity of LR is 0.9166666666666666

              precision    recall  f1-score   support

           0       0.82      0.89      0.85        72
           1       0.92      0.86      0.89       102

    accuracy                           0.87       174
   macro avg       0.87      0.88      0.87       174
weighted avg       0.88      0.87      0.87       174
```

Result of KNN



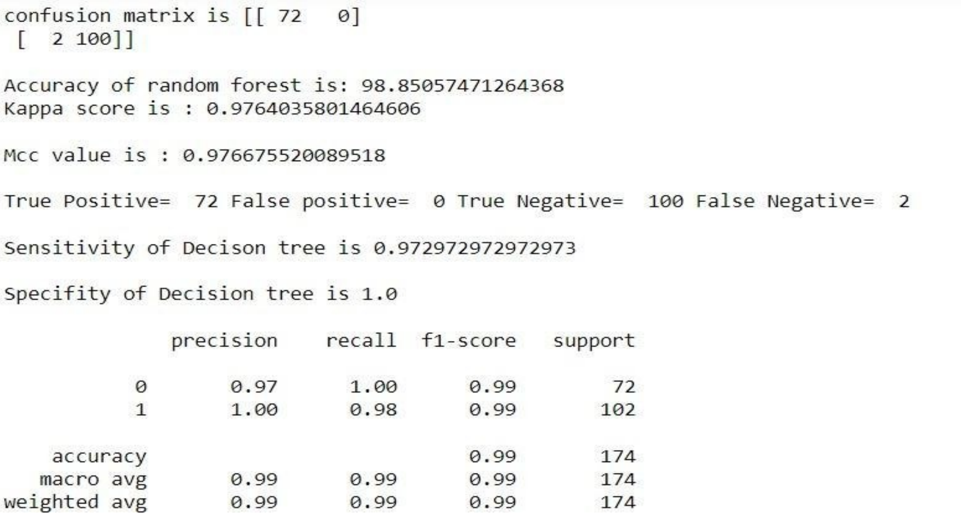
```
confusion matrix is [[ 72   0]
 [  0 102]]

Accuracy of KNN is 100.0

Kappa score is : 1.0

MCC score is : 1.0

True Positive=  72 False positive=  0 True Negative=  102 False Negative=  0

Sensitivity of KNN is 1.0

Specifity of KNN is 1.0

              precision    recall  f1-score   support

           0       1.00      1.00      1.00        72
           1       1.00      1.00      1.00       102

    accuracy                           1.00       174
   macro avg       1.00      1.00      1.00       174
weighted avg       1.00      1.00      1.00       174
```

Result of DT



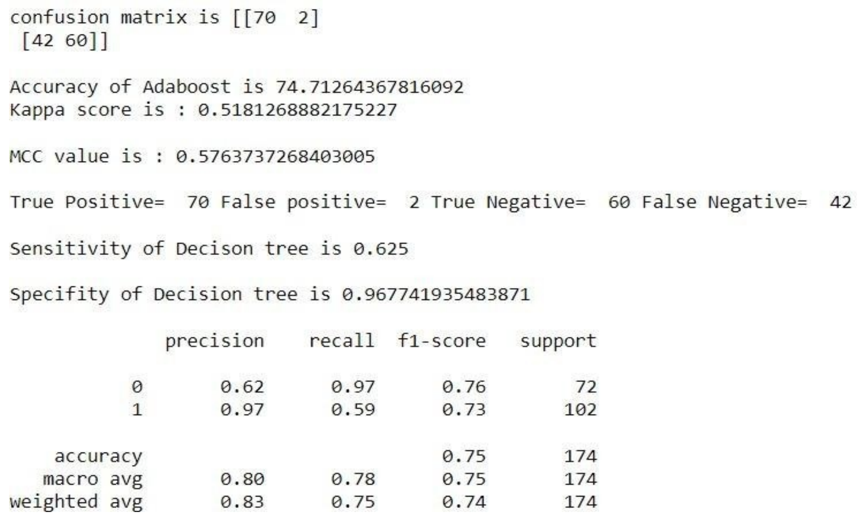
```
confusion matrix is [[72  0]
 [ 6 96]]

Accuracy of Decision tree is 96.55172413793103

Kappa score is : 0.9297820823244553

MCC score is : 0.9320827648567408

True Positive=  72 False positive=  0 True Negative=  96 False Negative=  6

Sensitivity of Decison tree is 0.9230769230769231

Specifity of Decision tree is 1.0

              precision    recall  f1-score   support

           0       0.92      1.00      0.96        72
           1       1.00      0.94      0.97       102

    accuracy                           0.97       174
   macro avg       0.96      0.97      0.96       174
weighted avg       0.97      0.97      0.97       174
```

Result of RF



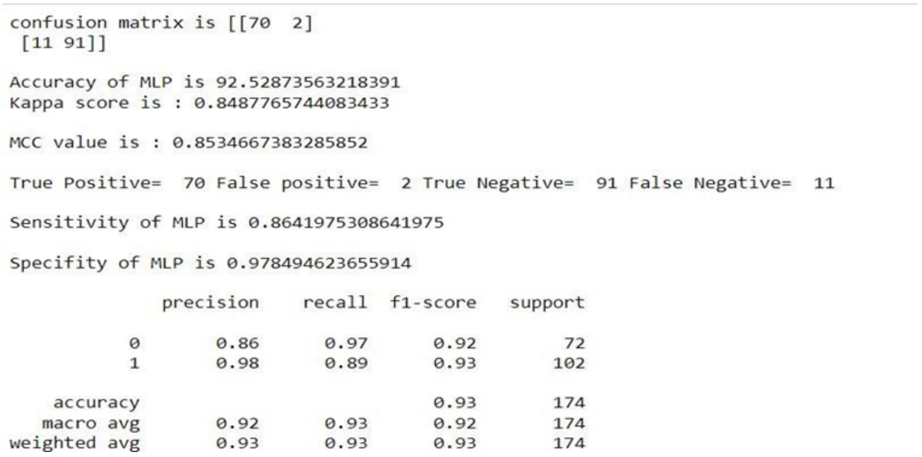
```
confusion matrix is [[ 72   0]
 [  2 100]]

Accuracy of random forest is: 98.85057471264368
Kappa score is : 0.9764035801464606

Mcc value is : 0.976675520089518

True Positive=  72 False positive=  0 True Negative=  100 False Negative=  2

Sensitivity of Decison tree is 0.972972972972973

Specifity of Decision tree is 1.0

              precision    recall  f1-score   support

           0       0.97      1.00      0.99        72
           1       1.00      0.98      0.99       102

    accuracy                           0.99       174
   macro avg       0.99      0.99      0.99       174
weighted avg       0.99      0.99      0.99       174
```

Result of AdaboostM1



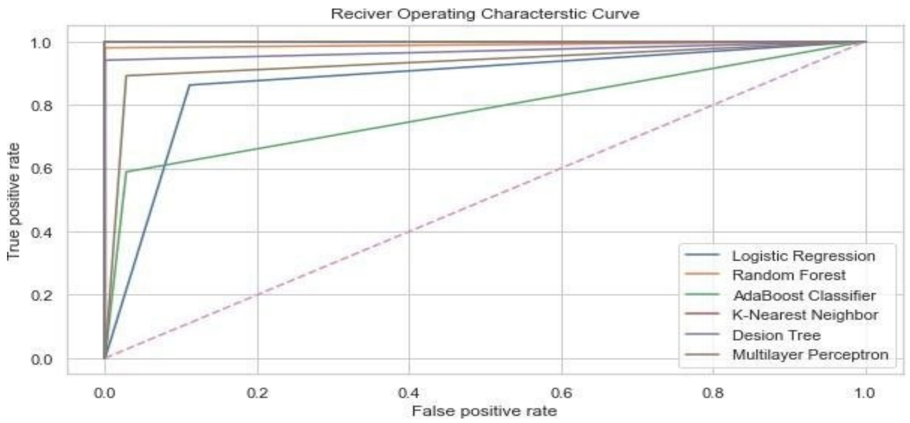
```
confusion matrix is [[70  2]
 [42 60]]

Accuracy of Adaboost is 74.71264367816092
Kappa score is : 0.5181268882175227

MCC value is : 0.5763737268403005

True Positive=  70 False positive=  2 True Negative=  60 False Negative=  42

Sensitivity of Decison tree is 0.625

Specifity of Decision tree is 0.967741935483871

              precision    recall  f1-score   support

           0       0.62      0.97      0.76        72
           1       0.97      0.59      0.73       102

    accuracy                           0.75       174
   macro avg       0.80      0.78      0.75       174
weighted avg       0.83      0.75      0.74       174
```

Result of MLP

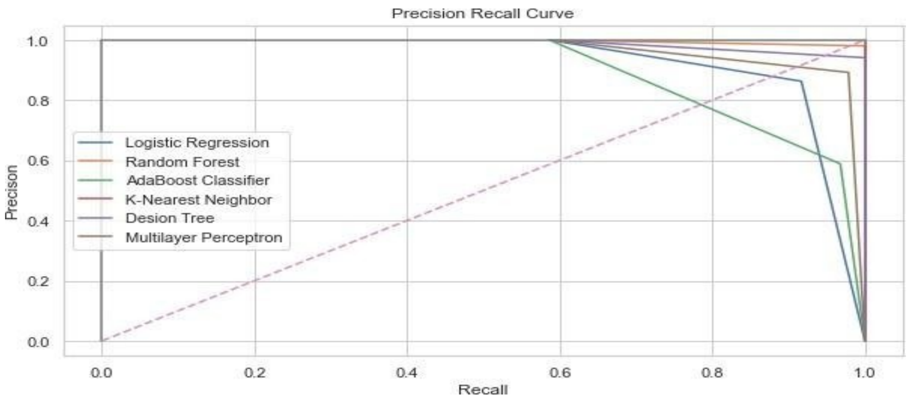


```
confusion matrix is [[70  2]
 [11 91]]

Accuracy of MLP is 92.52873563218391
Kappa score is : 0.8487765744083433

MCC value is : 0.8534667383285852

True Positive=  70 False positive=  2 True Negative=  91 False Negative=  11

Sensitivity of MLP is 0.8641975308641975

Specifity of MLP is 0.978494623655914

              precision    recall  f1-score   support

           0       0.86      0.97      0.92        72
           1       0.98      0.89      0.93       102

    accuracy                           0.93       174
   macro avg       0.92      0.93      0.92       174
weighted avg       0.93      0.93      0.93       174
```

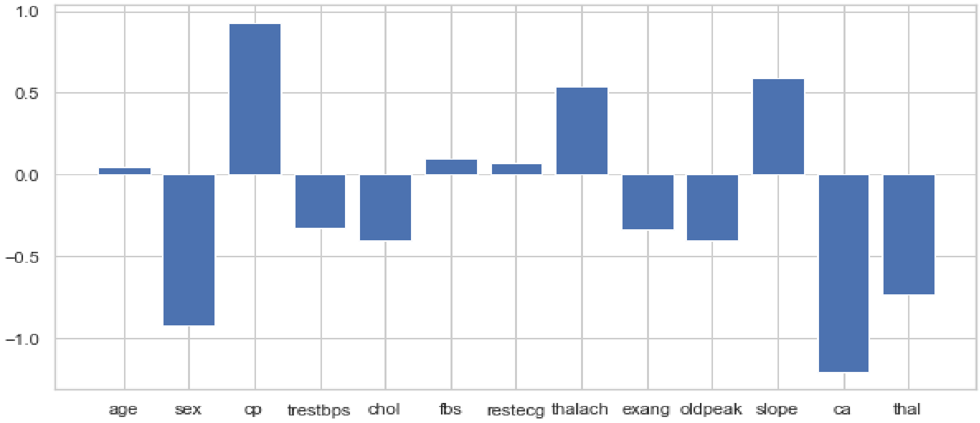
ROC curve for all Classifiers



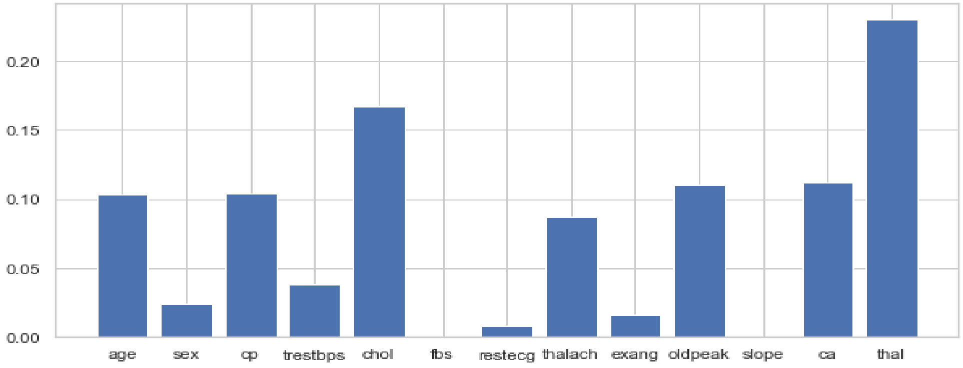
Precision vs Recall curve for all Classifiers



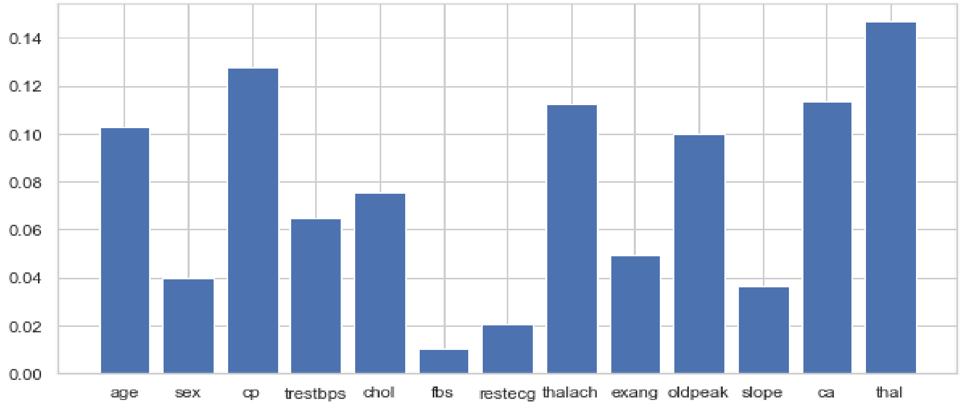
Feature Importance Score by Logistic Regression



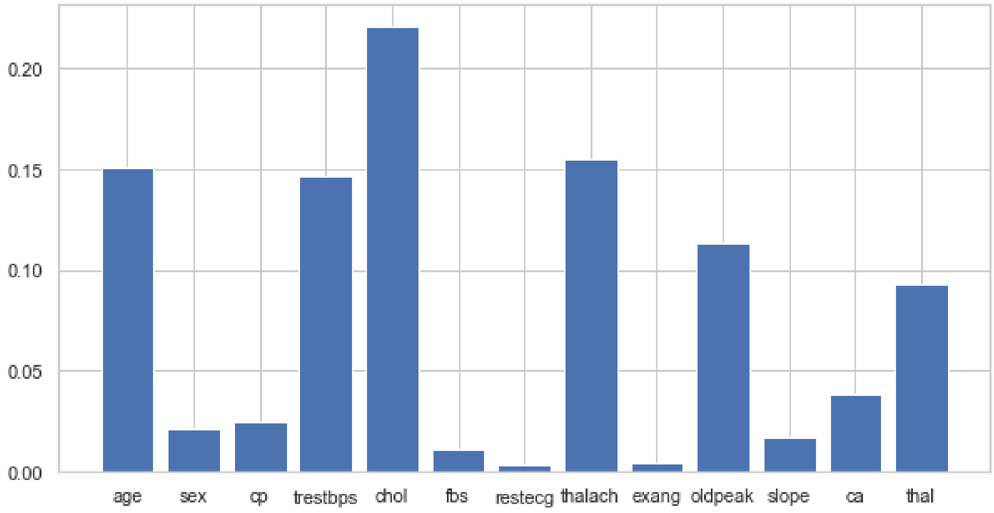
Feature Importance Score by Decision Tree



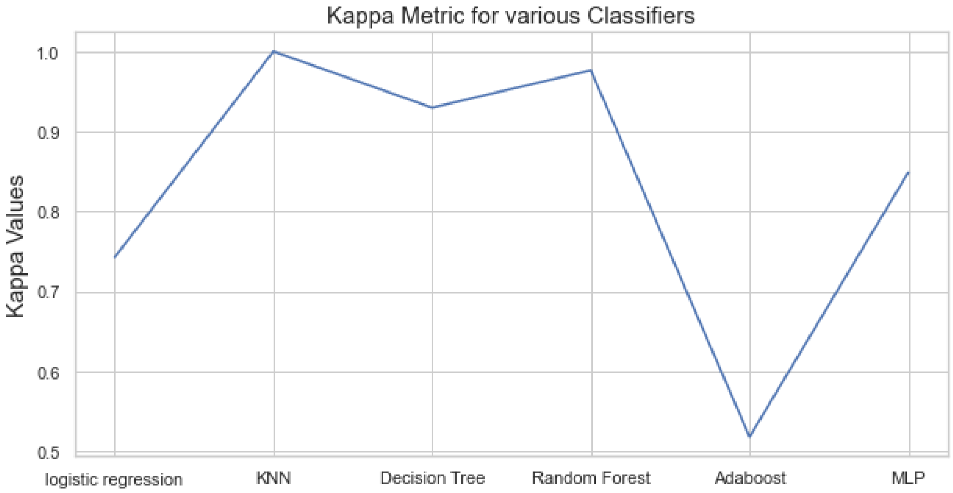
Feature Importance Score by Random Forest



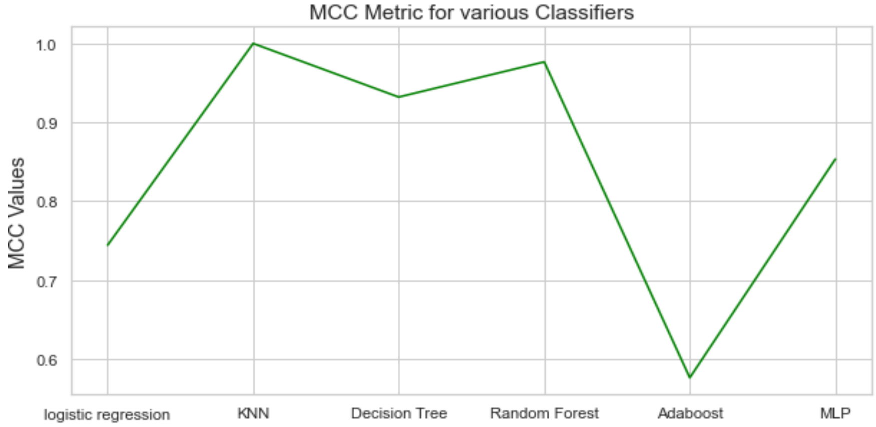
Feature Importance Score by AdaboostM1



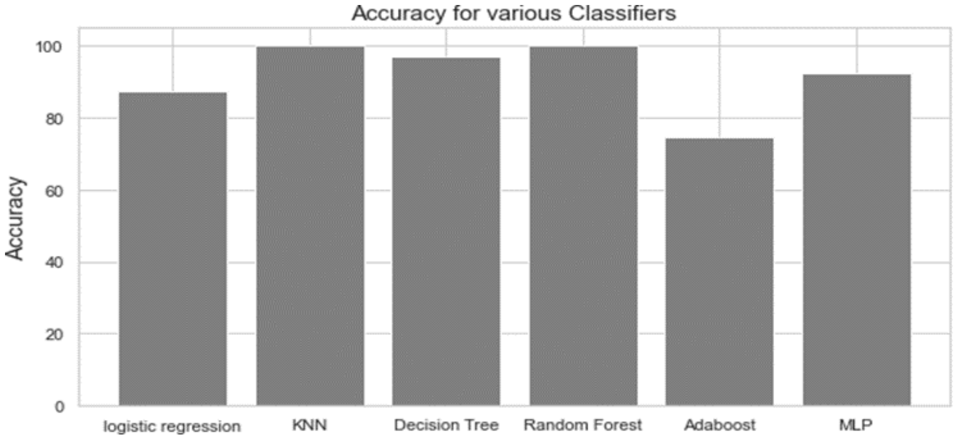
Kappa Metrics for All Classifiers



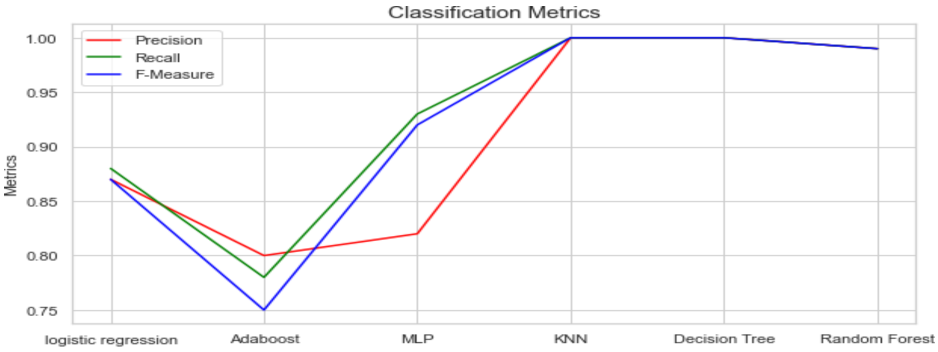
MCC Metrics for All Classifiers



Accuracy for All Classifiers



Classification Metrics for All Classifiers



Consolidated Feature Importance Scores:

Feature	LR	ABM1	DT	RF
Age	0.044	0.176	0.102	0.103
Sex	-0.921	0.027	0.024	0.045
Cp	0.929	0.034	0.104	0.14
Trestbps	-0.328	0.12	0.038	0.072
Chol	-0.408	0.171	0.167	0.076
Fb	0.099	0	0	0.012
restecg	0.075	0	0.007	0.016
Thalach	0.54	0.161	0.087	0.105
Exang	-0.336	0.005	0.016	0.037
Oldpeak	-0.401	0.155	0.109	0.099
slope	0.589	0.018	0	0.054
Ca	-1.208	0.054	0.112	0.103
thal	-0.733	0.074	0.229	0.132

10. CONCLUSION & FUTURE SCOPE

In this project, we introduce about the heart disease prediction system with different classifiertechniques for the prediction of heart disease. The techniques are Random Forest and LogisticRegression: we have analyzed that the Random Forest has better accuracy as compared toLogistic Regression. Our purpose is to improve the performance of the Random Forest byremoving unnecessary and irrelevant attributes from the dataset and only picking those thatare most informative for the classification task. Heart disease prediction which uses Machine learning algorithm provides users a prediction result if the user has heart disease. Recent advancements in technology made machine learning algorithms to evolve. In this proposed method Random Forest Algorithm was used because of its efficiency and accuracy. This algorithm is also used to find the heart disease prediction percentage by knowing the correlation details between diabetes and heart diseases. The similar prediction systems can be built by calculating correlation between heart diseases and other diseases. Also new algorithms can be used to achieve increased accuracy. Better performance is obtained with more parameter used in these algorithms

This study emphasizes how data mining techniques can be used to detect cardiac problems early. Promising paths for raising predicted accuracy have been found by analyzing a number of categorization methods, including Random Forest and Logistic Regression. The results point to the need for additional refining using feature selection techniques, as Random Forest performs better than Logistic Regression. We have the ability to revolutionize the diagnosis of cardiac disease and improve patient outcomes by adopting data mining techniques while maintaining patient safety as our first priority.

11. BIBILOGRAPHY

1. Patel, J., Upadhyay, P. and Patel, D. (2016) Heart Disease Prediction Using Machine learning and Data Mining Technique. Journals of Computer Science & Electronics, 7, 129-137.

2. Chavan Patil, A.B. and Sonawane, P. (2017) To Predict Heart Disease Risk and Medications Using Data Mining Techniques with an IoT Based Monitoring System for Post-Operative Heart Disease Patients. International Journal on Emerging Trends in Technology (IJETT), 4, 8274-8281.

3. Weng, S.F., Reps, J., Kai, J., Garibaldi, J.M. and Qureshi, N. (2017) Can Machine-Learning Improve Cardiovascular Risk Prediction Using Routine Clinical Data? PLoS ONE, 12, e0174944.
https://doi.org/10.1371/journal.pone.0174944

4. Zhao, W., Wang, C. and Nakahira, Y. (2011) Medical Application on Internet of Things. IET International Conference on Communication Technology and Application (ICCTA 2011), Beijing, 14-16 October 2011, 660-665.

5. Chiuchisan, I. and Geman, O. (2014) An Approach of a Decision Support and Home Monitoring System for Patients with Neurological Disorders Using Internet of Things Concepts. WSEAS Transactions on Systems, 13, 460-469.

6. Soni, J., Ansari, U. and Sharma, D. (2011) Intelligent and Effective Heart Disease Prediction System Using Weighted Associative Classifiers. International Journal on Computer Science and Engineering (IJCSE), 3, 2385-2392.

7. Yuce, M.R. (2010) Implementation of Wireless Body Area Networks for Healthcare Systems. Sensor and Actuators A: Physical, 162, 116-129.
https://doi.org/10.1016/j.sna.2010.06.004

8. Singh, M., Martins, L.M., Joanis, P. and Mago, V.K. (2016) Building a Cardiovascular Disease Predictive Model Using Structural Equation Model and Fuzzy Cognitive Map. IEEE International Conference on Fuzzy Systems (FUZZ), Vancouver, 24-29 July 2016, 1377-1382.
https://doi.org/10.1109/FUZZ-IEEE.2016.7737850

9. Ghadge, P., Girme, V., Kokane, K. and Deshmukh, P. (2016) Intelligent Heart Attack Prediction System Using Big Data. International Journal of Recent Research in Mathematics Computer Science and Information Technology, 2, 73-77.

10. Shouman, M., Turner, T. and Stocker, R. (2012) Using Data Mining Techniques in Heart Disease Diagnosis and Treatment. Electronics, Communications, and Computers, Alexandria, 173-177.