CHAPTER 1

INTRODUCTION

A heart attack which is analogous to acute myocardial infarction (AMI) is one of the most serious diseases in the segment of cardiovascular disease. It occurs due to the interruption of blood circulation to muscle of the heart which damages the heart the muscle. Diagnosing heart disease is also a crucial task. The symptoms, physical examination, and understanding of the different signs of this disease are required to diagnose heart disease. Different factors including cholesterol, genetic heart disease, high blood pressure, low physical activity, obesity, and smoking can be reasons for the occurrence of heart disease. The major reason for heart attacks is the stoppage of blood to the coronary arteries. The red blood cells (RBC) start getting low when blood flow is reduced; due to this the human body stops getting necessary oxygen and loses consciousness. The early diagnosis through symptoms and signs can help prevent patients of heart attacks if the prediction is accurate enough. Figure 1.1 shows different symptoms of a heart attack. activity, obesity, and smoking can be reasons for the occurrence of heart disease. The major reason for heart attacks is the stoppage of blood to the coronary arteries The work presented takes 13 features/attributes as input having number values. It has been stated that little modifications in lifestyle including quitting smoking/alcohol/tobacco, having healthy food habits, and routine exercises can help in the prevention of heart attacks. Any person living a healthy lifestyle with early treatment after diagnosis can greatly increase the positive results. However, it is difficult to identify the high risk of heart disease where different risks like diabetes, high blood pressure, and cholesterol problems are present. In these types of scenarios, ML can help in the early diagnosis of disease.

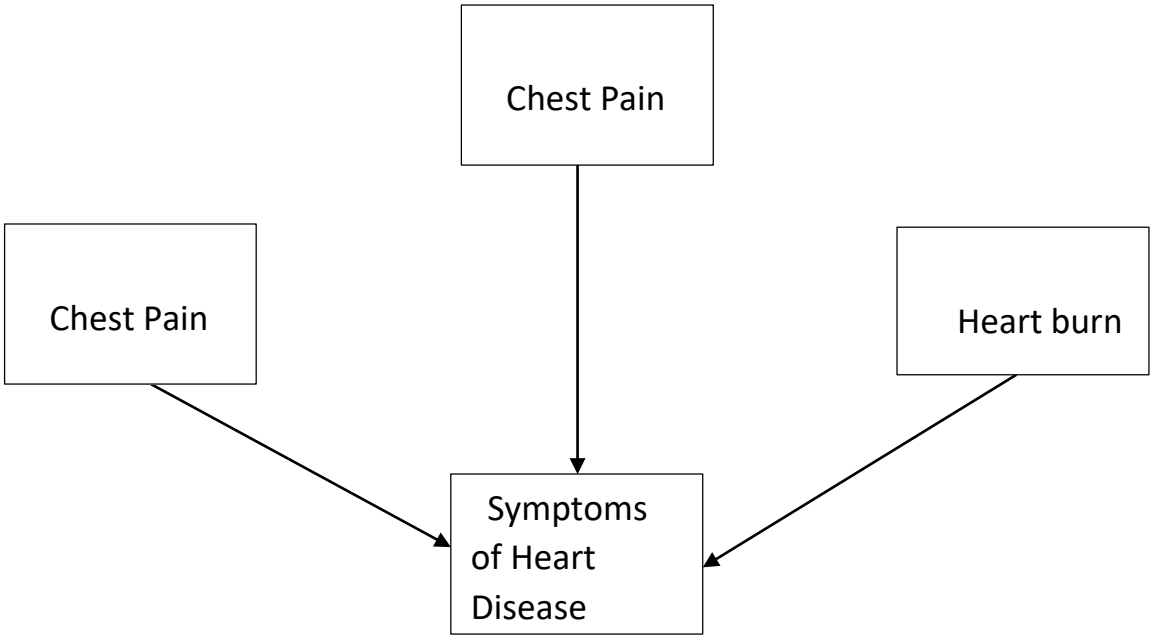

Figure 1.1 – Symptoms of Heart Disease

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

Machine learning proves to be effective in assisting in making decisions and predictions from the large quantity of data produced by the health care industry. This project aims to predict future Heart Disease by analysing data of patients which classifies whether they have heart disease or not using machine-learning algorithm. Machine Learning techniques can be a boon in this regard.

Even though heart disease can occur in different forms, there is a common set of core risk factors that influence whether someone will ultimately be at risk for heart disease or not. By collecting the data from various sources, classifying them under suitable headings & finally analysing to extract the desired data we can say that this technique can be very well adapted to do the prediction of heart disease.

1.1 GENERAL OVERVIEW OF DATA SCIENCE

Data science is an interdisciplinary field that uses scientific methods, processes, algorithms and systems to extract knowledge and insights from structured and unstructured data, and apply knowledge and actionable insights from data across a broad range of application domains. The term "data science" has been traced back to 1974, when Peter Naur proposed it as an alternative name for computer science. In 1996, the International Federation of Classification Societies became the first conference to specifically feature data science as a topic. However, the definition was still in flux. In less than a decade, it has become one of the hottest and most trending professions in the market. Data science is the field of study that combines domain expertise, programming skills, and knowledge of mathematics and statistics to extract meaningful insights from data. Data science can be defined as a blend of mathematics, business acumen, tools, algorithms and machine learning techniques, all of which help us in finding out the hidden insights or patterns from raw data which can be of major use in the formation of big business decisions.

1.2 DATA SCIENCE

Data science is the study of data to extract meaningful insights for business. It is a multidisciplinary approach that combines principles and practices from the fields of mathematics, statistics, artificial intelligence, and computer engineering to analyse large amounts of data.

Required skills for a Data science

Machine Learning – NLP, Classification, Clustering

Programming Language – Python, SQL , Scale, Java, Matlab

Data Visualization – Tableauu, SAS, Java ,R Libraries, D3.JS

Big Data Platform – Mango DB, Oracle, Microsoft Azure

1.3 ARTIFICIAL INTELLIGENCE

Artificial Intelligence (AI) refers to the simulation of human intelligence in machines that are programmed to think like humans and mimic their actions. The term may also be applied to any machine that exhibits traits associated with a human mind such as learning and problem-solving. Leading AI textbooks define the field as the study of "intelligent agents" any system that perceives its environment and takes actions that maximize its chance of achieving its goals. Some popular accounts use the term "artificial intelligence" to describe machines that mimic "cognitive" functions that humans associate with the human mind, such as "learning" and "problem solving", however this definition is rejected by major AI researchers. Artificial intelligence is the simulation of human intelligence processes by machines, especially computer systems. Specific applications of AI include expert systems, natural languageprocessing, speech recognition and machine vision. Artificial intelligence was founded as an academic discipline in 1956, and in the years since has experienced several waves of optimism, followed by disappointment and the loss of finding (known as an "AI winter"), followed by new approaches, success and renewed funding. AI research has tried and discarded many different approaches during its lifetime, including simulating the brain, modelling human problem solving, formal logic, large databases of knowledge and imitating animal behaviour.

In the first decades of the 21st century, highly mathematical statistical machine learning has dominated the field, and this technique has proved highly

successful, helping to solve many challenging problems throughout industry and academia. The various sub-fields of AI research are centered on particular goals and the use of particular tools. General intelligence (the ability to solve an arbitrary problem) is among the field's long-term goals. AI also draws upon computer science, psychology, linguistics, philosophy, and many other fields. The field was founded on the assumption that human intelligence "can be so precisely described that a machine can be made to simulate it". This raises philosophical arguments about the mind and the ethics of creating artificial beings endowed with human-like intelligence. These issues have been explored by myth, fiction and philosophy since antiquity. As the hyper around AI has accelerated, vendors have been scrambling to promote how their products and services use AI.

Often what they refer to as AI is simply one component of AI, such as machine learning. AI requires a foundation of specialized hardware and software for writing and training machine learning algorithms. No one programming language is synonymous with AI, but a few, including Python, R and Java, are popular. In general, AI systems work by ingesting large amounts of labelled training data, analysing the data for correlations and patterns, and using these patterns to make predictions about future states. In this way, a Chabot that is fed examples of text chats can learn to produce life like exchanges with people, or an image recognition tool can learn to identify and describe objects in images by reviewing millions of examples.AI programming focuses on three cognitive skills: learning, reasoning and self-correction.

1.3.1 LEARNING PROCESSES

This aspect of AI programming focuses on acquiring data and creating rules for how to turn the data into actionable information. The rules, which are called algorithms, provide computing devices with step-by-step instructions for how to complete a specific task.

1.3.2 REASONING PROCESSES

This aspect of AI programming focuses on choosing the right algorithm to reach a desired outcome.

1.3.3 SELF – CORRECTION PROCESSES

This aspect of AI programming is designed to continually fine-tune algorithms and ensure they provide the most accurate results possible. AI is important because it can give enterprises insights into their operations that they may not have been aware of previously and because, in some cases, AI can perform tasks better than humans. Particularly when it comes to repetitive, detail-oriented tasks like analysing large numbers of legal documents to ensure relevant fields are filled in properly, AI tools often complete jobs quickly and with relatively few errors. Artificial neural networks and deep learning artificial intelligence technologies are quickly evolving, primarily because AI processes large amounts of data much faster and makes predictions more accurately than humanly possible.

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

1.4 PROBLEM STATEMENT

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

CHAPTER 2

LITERATURE SURVEY

1. Akkaya, B., Sener, E. and Gursu, C. (2022) A Comparative Study of Heart Disease Prediction Using Machine Learning Techniques. 2022 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA), Ankara,9-11 June 2022.

In this study, data from the 2020 survey presented by the Centers for Disease Control and Prevention (CDC) on the Behavioral Risk Factor Surveillance System (BRFSS) were analyzed using 8 different machine learning classification methods. These methods are Logistic Regression (LR), Support Vector Machines (SVM), Naive Bayes (NB), k-Nearest Neighbor (k-NN), Decision Tree (DT), Adaboost, Multilayer Perceptron (MLP), and XGBoost (XGB). To overcome this problem, the dependent variable was stabilized using the Synthetic Minority Oversampling Technique Tomek Links (SMOTE -Tomek Link) method before applying the classification methods.

2. Xing, Y.W., Wang, J., Zhao, Z.H. and Gao, Y.H. (2007) Combination Data Mining Methods with New Medical Data to Predicting Outcome of Coronary Heart Disease.Convergence Information Technology, Gwangju, 21-23 November 2007, 868-872.

The goal of this paper is to develop data mining algorithms for predicting survival of CHD patients based on 1000 cases .We carry out a clinical observation and a 6-month follow up to include 1000 CHD cases. The survival information of each case is obtained via follow up. Based on the data, we employed three popular data mining algorithms to develop the prediction models using the 502 cases.

3. Nahar, J., Imam, T., Tickle, K.S. and Chen, Y.P.P. (2013) Computational Intelligence for Heart Disease Diagnosis: A Medical Knowledge Driven Approach. Expert Systems with Applications , 40, 96-104

This paper investigates a number of computational intelligence techniques in the detection of heart disease. Particularly, comparison of six well known classifiers for the well used Cleveland data is performed. Further, this paper highlights the potential of an expert judgment based (i.e., medical knowledge driven) feature selection process (termed as MFS), and compare against the generally employed computational intelligence based feature selection mechanism.

4. Desai, F., Chowdhury, D., Kaur, R., Peeters, M., Arya, R.C., Wander, G.S., Gill, S.S. and Buyya, R. (2022) HealthCloud: A System for Monitoring Health Status of Heart Patients Using Machine Learning and Cloud Computing. Internet of Things , 17, Article Id : 100485

The presence of heart disease is predicted using machine learning algorithms such as Support Vector Machine, K-Nearest Neighbours, Neural Networks,Logistic Regression and Gradient Boosting Trees. This paper evaluates these machine learning algorithms to obtain the most accurate model, in compliance with Quality of Service (QoS) parameters.

5. Nahar, J., Imam, T., Tickle, K.S. and Chen, Y.-P.P. (2013) Association Rule Mining to Detect Factors Which Contribute to Heart Disease in Males and Females. Expert Systems with Applications , 40, 1086-1093.

This paper investigates the sick and healthy factors which contribute to heart disease for males and females.Analyzing the information available on sick and healthy individuals and taking confidence as an indicator, females are seen to have less chance of coronary heart disease then males.

6. Wilson, P.W.F., D'Agostino, R.B., Levy, D., Belanger, A.M., Silbershatz, H. and Kannel, W.B. (1998) Prediction of Coronary Heart Disease Using Risk Factor Categories.Circulation, 97, 1837-1847

The objective of this study was to examine the association of Joint National Committee (JNC-V) blood pressure and National Cholesterol Education Program (NCEP) cholesterol categories with coronary heart disease (CHD) risk, to incorporate them into coronary prediction algorithms, and to compare the discrimination properties of this approach with other noncategorical prediction functions.

7. Liu, X., Wang, X.L., Su, Q., Zhang, M., Zhu, Y.H., Wang, Q.G. and Wang, Q. (2017) A Hybrid Classification System for Heart Disease Diagnosis Based on the RFRS Method. Computational and Mathematical Methods in Medicine , 2017, 1-11.

Heart disease is one of the most common diseases in the world. The objective of this study is to aid the diagnosis of heart disease using a hybrid classification system based on the Relief and Rough Set (RFRS) method. The proposed system contains two subsystems: the RFRS feature selection system and a classification system with an ensemble classifier.

8. Makino, K., Lee, S., Bae, S., Chiba, I., Harada, K., Katayama, O., Shinkai,Y. and Shimada, H. (2021) Absolute Cardiovascular Disease Risk Assessed in Old Age Predicts Disability and Mortality: A Retrospective Cohort Study of Community-Dwelling Older Adults. Journal of the American Heart Association, 10, e022004.

We aimed to examine the longitudinal associations of absolute CVD risk assessed using region-specific risk estimation charts with disability and mortality among community-dwelling people aged ≥65 years. Methods and Results This retrospective cohort study included 7456 community-dwelling

people aged ≥65 years (mean age, 73.7 years) without CVD and functional decline at baseline.

9. Nagaraj M Lutimath,Chethan C,Basavaraj S Pol.,Prediction Of Heart Disease using Machine Learning, International journal Of Recent Technology and Engineering,8,(2S10), pp 474-477, 2019.

Naive Bayes Classification is a vital approach of classification in machine learning. The heart disease consists of set of range disorders affecting the heart. It includes blood vessel problems such as irregular heart beat issues, weak heart muscles, congenital heart defects, cardio vascular disease and coronary artery disease.

10. Fahd Saleh Alotaibi, Implementation of Machine Learning Model to Predict Heart Failure Disease, (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 10, No. 6, 2019.

This paper aims to improve the HF prediction accuracy using UCI heart disease dataset. For this, multiple machine learning approaches used to understand the data and predict the HF chances in a medical database. Furthermore, the results and comparative study showed that, the current work improved the previous accuracy score in predicting heart disease.

CHAPTER 3

METHODOLOGY

3.1 EXISTING SYSTEM

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

3.2 PROPOSED SYSTEM

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

This system is implemented using the following modules.

1. Collection of Dataset
2. Selection of attributes
3. Data Pre-Processing
4. Balancing of Data
5. Disease Prediction

3.2.1 COLLECTION OF DATASET

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

Every day, the average human heart beats around 100,000 times, pumping 2,000 gallons of blood through the body. Inside your body there are 60,000 miles of blood vessels.

The signs of a woman having a heart attack are much less noticeable than the signs of a male. In women, heart attacks may feel uncomfortable squeezing, pressure, fullness, or pain in the center of the chest. It may also cause pain in one or both arms, the back, neck, jaw or stomach, shortness of breath, nausea and other symptoms. Men experience typical symptoms of heart attack, such as chest pain , discomfort, and stress. They may also experience pain in other areas, such as arms, neck, back, and jaw, and shortness of breath, sweating, and discomfort that mimics heartburn.

It’s a lot of work for an organ which is just like a large fist and weighs between 8 and 12 ounces.

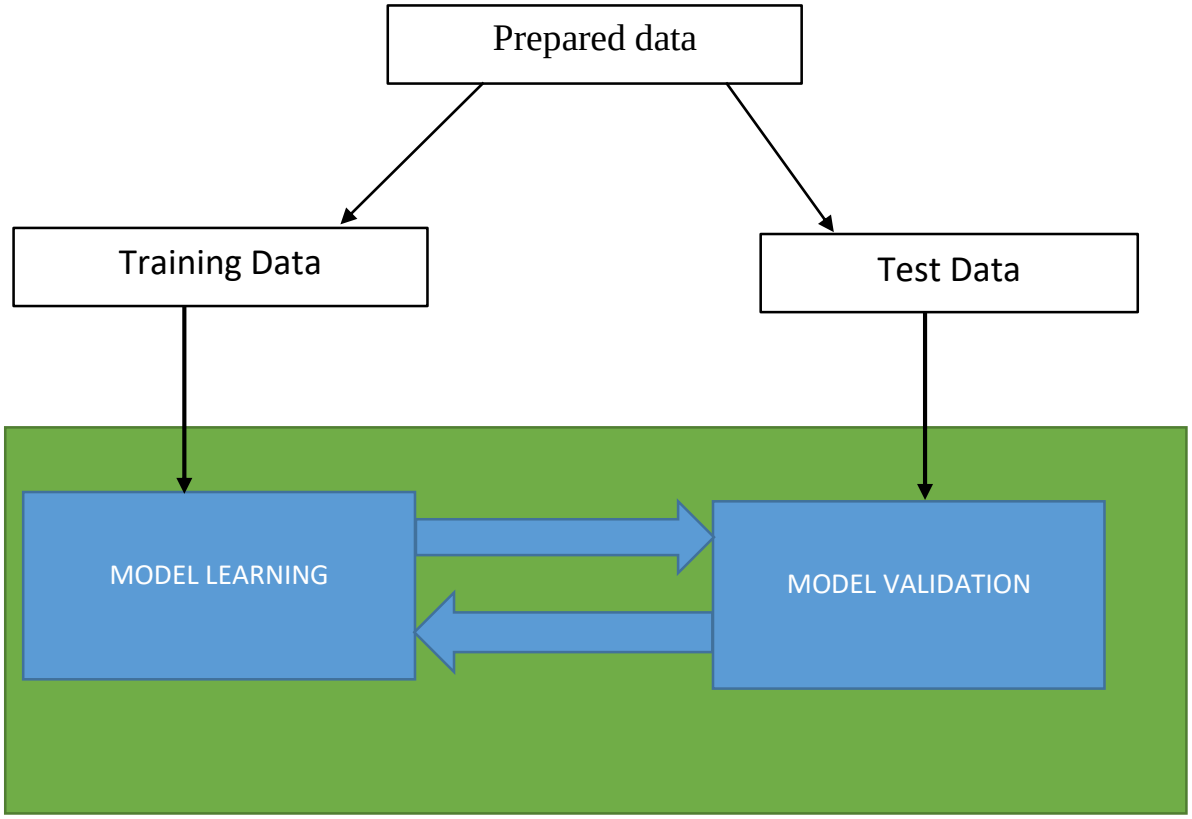

Figure 3.1 – Machine Learning

Machine learning is used to build algorithms that can receive the input data and use statistical analysis to predict the output, based upon the type of data available. These machine learning algorithms are classified as supervised, unsupervised and reinforcement learning where all these algorithm has various limitless applications such as Image Recognition, Voice Recognition, Predictions, Video Surveillance, Social Media Platform, Spam and Malware, Customer support, Search engine, Applications, Fraud and Preferences, etc. Machine learning (ML) also helps in developing the application for voice recognition. It also referred to as virtual personal assistants (VPA). It will help you to find the information when asked over the voice.

3.2.2 PRE-PROCESSING OF DATA

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

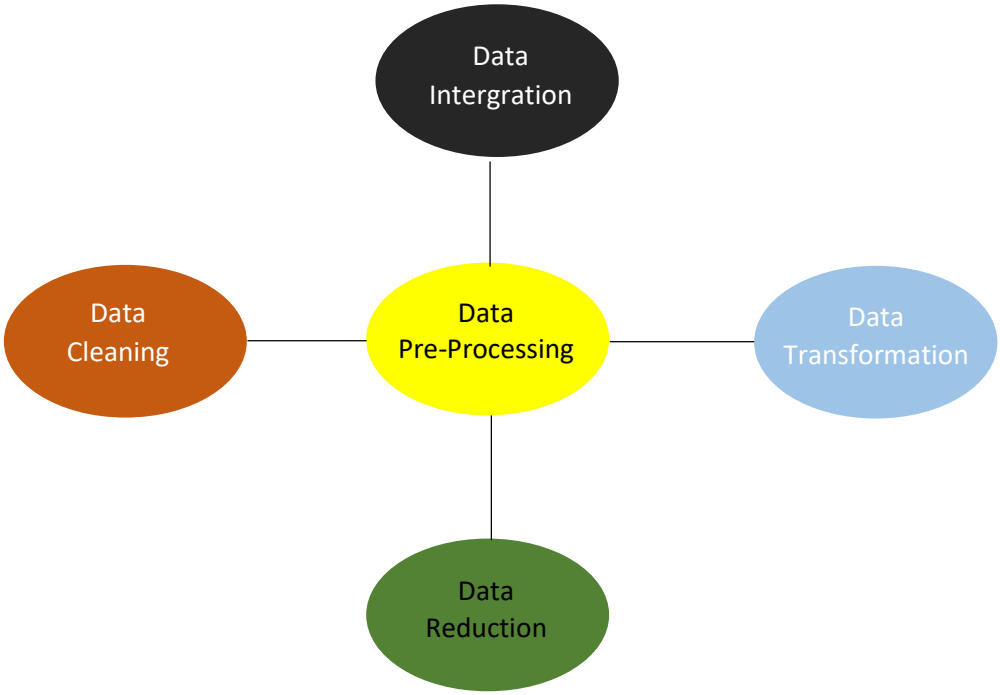

Figure – 3.2 Pre-processing of Data

3.2.3 SELECTION OF ATTRIBUTES

Attribute or Feature selection includes the selection of appropriate attributes for the prediction system. Initially, data may not be clean or in the required format for the model which can cause misleading outcomes Various attributes of the patient like gender, chest pain type, fasting blood pressure, serum cholesterol, exang, etc are selected for the prediction. The Correlation matrix is used for attribute selection for this model.

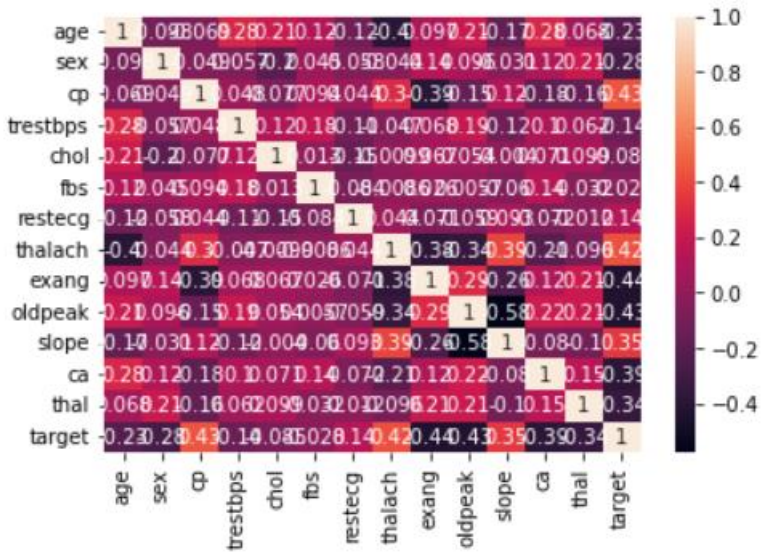

Figure 3.3 - Correlation matrix

3.2.4 BALANCING OF DATA

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

In recent times dealing with data has become a tedious job, considering the fact that most of the time is spent on cleaning and preprocessing the data. Often the data in the real-world is not available as per our expectations

A balanced dataset is a dataset where each output class (or target class) is represented by the same number of input samples.

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

(b) Over Sampling: in over Sampling, dataset balance is done by increasing the size of the scarce samples. This process is considered when the amount of data is inadequate.

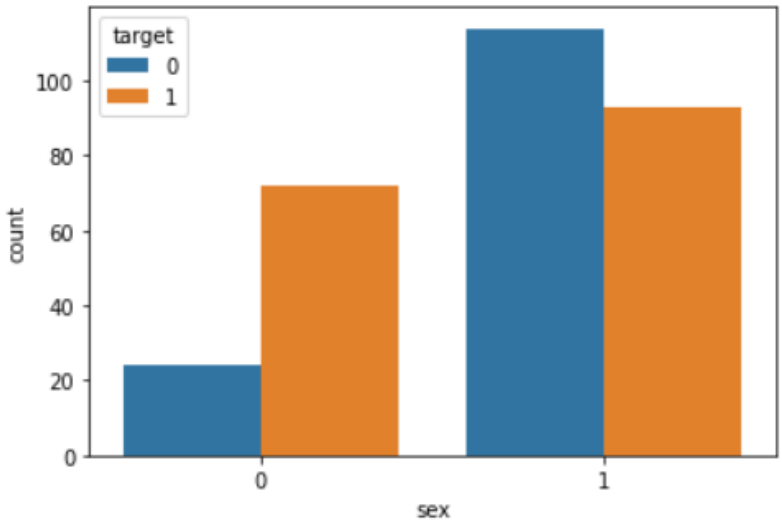

Figure 3.4 – Balancing of Data

3.2.5 PREDICTION OF DISEASE

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

Cardiovascular disease is the leading cause of death in many countries. Physicians often diagnose cardiovascular disease based on current clinical tests and previous experience of diagnosing patients with similar symptoms. Patients who suffer from heart disease require quick diagnosis, early treatment and constant observations. To address their needs, many data mining approaches have been used in the past in diagnosing and predicting heart diseases. Previous research was also focused on identifying the significant contributing features to heart disease prediction, however, less importance was given to identifying the strength of these features.

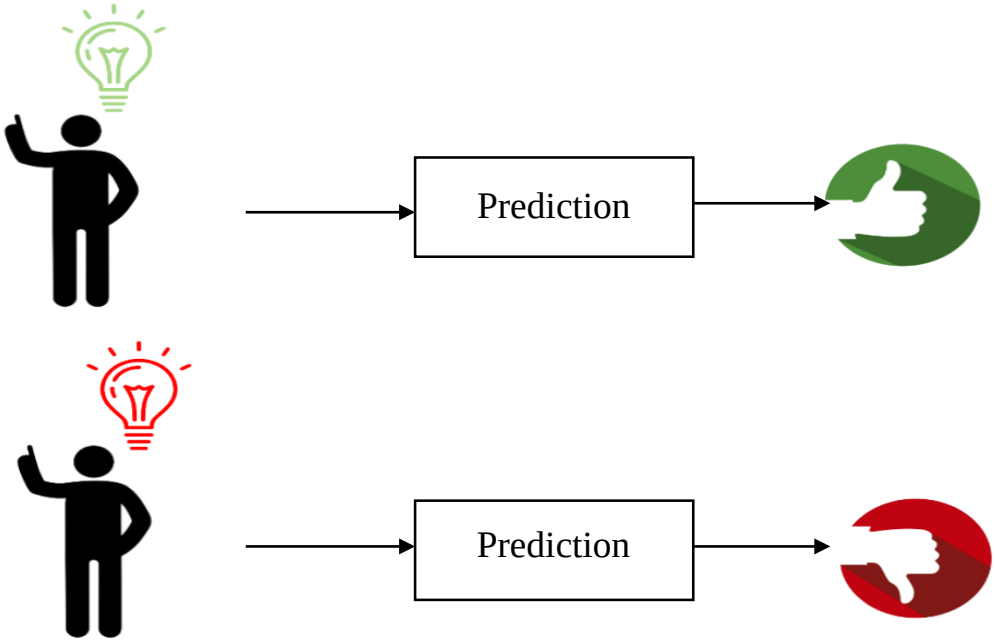

Figure 3.5 – Prediction Disease

CHAPTER 4

WORKING OF SYSTEM

4.1 SYSTEM ARCHITECTURE

The system architecture gives an overview of the working of the system.The working of this system is described as follows:

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

Figure 4.1 depicts how machine learning model was created with the help of collected dataset. Then Dataset should be pre-processed. Thereafter create various machine learning algorithm models were created for getting better performance .Machine learning model is trained with Xtrain and Ytrain . Then Model ready to predict outcome with the help of Xtest data .Thereafter model predicts output based on input data.

4.2 MACHINE LEARNING

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

4.2.1 SUPERVISED LEARNING

Supervised learning is the type of machine learning in which machines are trained using well "labelled" training data, and on the basis of that data, machines predict the output

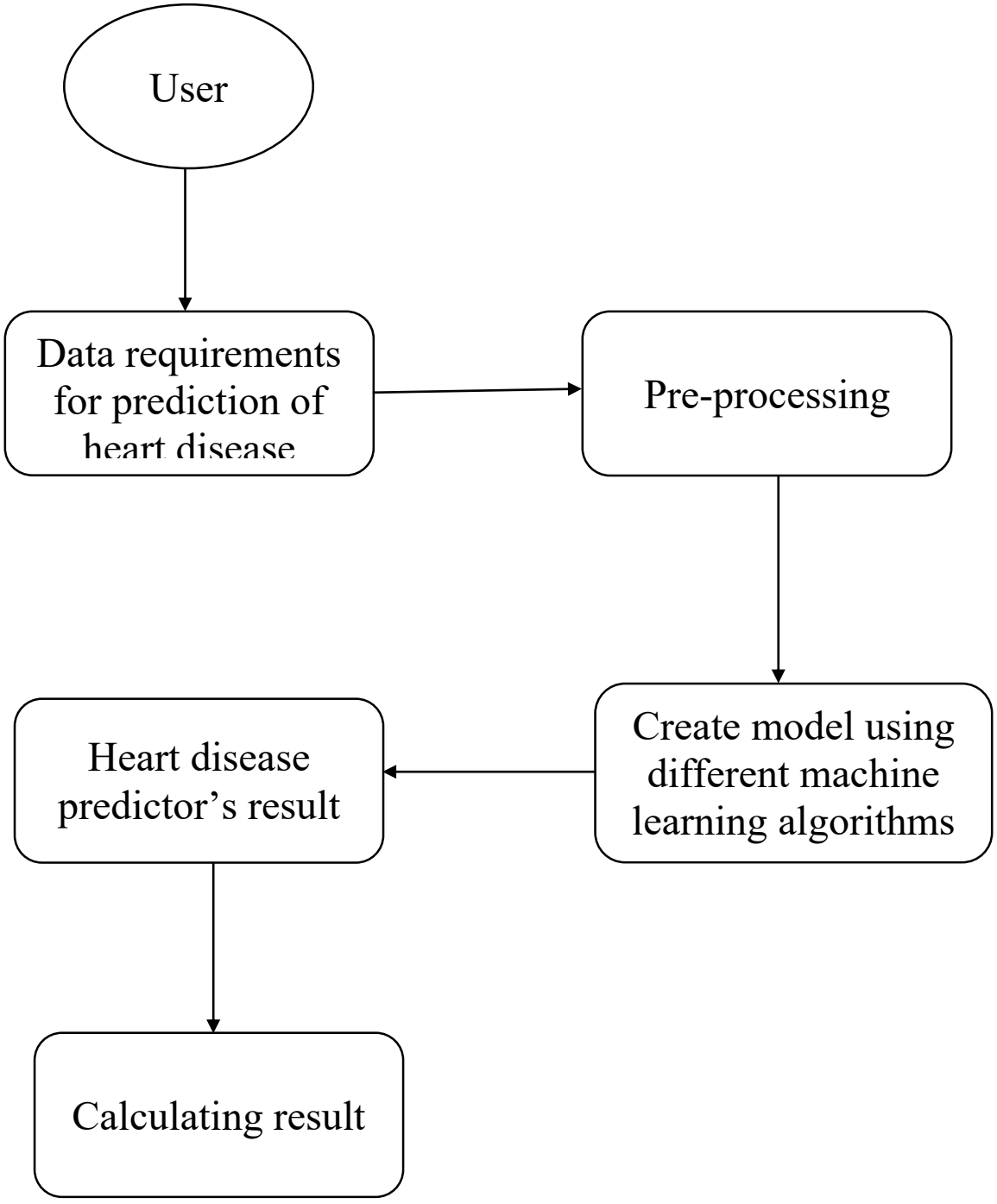

Figure 4.1 System Architecture

In supervised learning, the training data provided to the machines work as the supervisor that teaches the machines to predict the output correctly. It applies the same concept as a student learns in the supervision of the teacher. Supervised learning is a process of providing input data as well as correct output data to the machine learning model. The aim of a supervised learning algorithm is to find a mapping function to map the input variable(x) with the output variable(y).

4.2.2 UNSUPERVISED LEARNING

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

Unsupervised learning is helpful for finding useful insights from the data.

Unsupervised learning is much similar to how a human learns to think by their own experiences, which makes it closer to the real AI.

Unsupervised learning works on unlabelled and uncategorized data which make unsupervised learning more important.

In real-world, we do not always have input data with the corresponding output so to solve such cases, we need unsupervised learning.

4.2.3 REINFORCEMENT LEARNING

Reinforcement learning is an area of Machine Learning. It is about taking suitable action to maximize reward in a particular situation. It is employed by various software and machines to find the best possible behaviour or path it should take in a specific situation. Reinforcement learning differs from supervised learning in a way that in supervised learning the training data has the answer key

with it so the model is trained with the correct answer itself whereas in reinforcement learning, there is no answer but the reinforcement agent decides what to do to perform the given task. In the absence of a training dataset, it is bound to learn from its experience.

4.3 ALGORITHMS

4.3.1 SUPPORT VECTOR MACHINE (SVM)

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

The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane.SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called support vectors, and hence the algorithm is termed as Support Vector Machine. Support vector machines (SVMs) are powerful yet flexible supervised machine learning algorithms which are used both for classification and regression. But generally, they are used in classification problems. In the 1960s, SVMs were first introduced but later they got refined in 1990. SVMs have their unique way of implementation as compared to other machine learning algorithms. Lately, they are extremely popular because of their ability to handle multiple continuous and categorical variables.

4.3.1.1 THE FOLLOWINGS ARE IMPORTANT CONCEPTS IN SVM

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

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

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

4.3.1.2 TYPES OF SVM:

SVM can be of two types:

1.Linear SVM: Linear SVM is used for linearly separable data, which means if a dataset can be classified into two classes by using a single straight line, then such data is termed as linearly separable data, and classifier is used called as Linear SVM classifier.

2. Non-linear SVM: Non-Linear SVM is used for non-linearly separated data, which means if a dataset cannot be classified by using a straight line, then such data is termed as non-linear data and classifier used is called as Non-linear SVM classifier. The objective of the support vector machine algorithm is to find a hyperplane in an N-dimensional space (N - the number of features) that distinctly classifies the data points.

4.3.1.3 THE ADVANTAGES OF SUPPORT VECTOR MACHINES ARE

1. Effective in high dimensional spaces.
2. Still effective in cases where the number of dimensions is greater than the number of samples.
3. Uses a subset of training points in the decision function (called support vectors), so it is also memory efficient.

4. Versatile: different kernel functions can be specified for the decision function. Common kernels are provided, but it is also possible to specify custom kernels.

4.3.1.4 THE DISADVANTAGES OF SUPPORT VECTOR MACHINES INCLUDE

1. If the number of features is much greater than the number of samples, avoid over-fitting in choosing Kernel functions and regularization term is crucial.
2. SVMs do not directly provide probability estimates, these are calculated using an expensive five-fold cross-validation.

4.3.2 LOGISTIC REGRESSION ALGORITHM

Logistic regression is one of the most popular Machine Learning algorithms, which comes under the Supervised Learning technique. It is used for predicting the categorical dependent variable using a given set of independent variables.

Logistic regression predicts the output of a categorical dependent variable. Therefore the outcome must be a categorical or discrete value. It can be either Yes or No, 0 or 1, true or False, etc. but instead of giving the exact value as 0 and 1, it gives the probabilistic values which lie between 0 and 1.

Logistic Regression is much similar to the Linear Regression except that how they are used. Linear Regression is used for solving Regression problems, whereas logistic regression is used for solving the classification problems.

In Logistic regression, instead of fitting a regression line, we fit an "S"shaped logistic function, which predicts two maximum values (0 or 1).

The curve from the logistic function indicates the likelihood of something such as whether the cells are cancerous or not, a mouse is obese or not based on its weight, etc.

Logistic Regression is a significant machine learning algorithm because it has the ability to provide probabilities and classify new data using continuous and discrete datasets.

4.3.2.1 WORKING

Machine learning generally involves predicting a quantitative outcome or a qualitative class. The former is commonly referred to as a regression problem. In the scenario of linear regression, the input is a continuous variable, and the prediction is a numerical value. When predicting a qualitative outcome (class), the task is considered a classification problem. Examples of classification problems include predicting what products a user will buy or if a target user will click on an online advertisement.

Not all algorithms fit cleanly into this simple dichotomy, though, and logistic regression is a notable example. Logistic regression is part of the regression family as it involves predicting outcomes based on quantitative relationships between variables. When predicting a qualitative outcome (class), the task is considered a classification problem. Examples of classification problems include predicting what products a user will buy or if a target user will click on an online advertisement. However, unlike linear regression, it accepts both continuous and discrete variables as input and its output is qualitative. In addition, it predicts a discrete class such as “Yes/No” or “Customer/Non-customer”.

In practice, the logistic regression algorithm analyses relationships between variables. It assigns probabilities to discrete outcomes using the sigmoid function, which converts numerical results into an expression of probability between 0 and 1.0

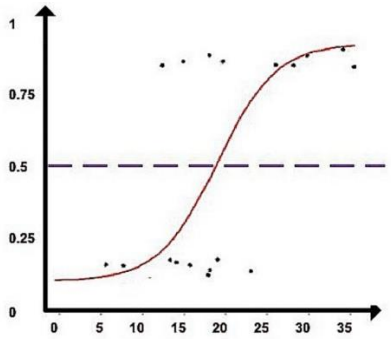

Figure 4.1 – Sigmoid Curve

Probability is either 0 or 1, depending on whether the event happens or not. For binary predictions, you can divide the population into two groups with a cut-off of 0.5. Everything above 0.5 is considered to belong to group A, and everything below is considered to belong to group B.

Figure 4.2 - Hyperplane

A hyperplane is used as a decision line to separate two categories (as far as possible) after data points have been assigned to a class using the sigmoid function. The class of future data points can then be predicted using the decision boundary.

4.3.2.2 ADVANTAGES

- Logistic Regression is one of the simplest machine learning algorithms and is easy to implement yet provides great training efficiency in some cases. Also due to these reasons, training a model with this algorithm doesn't require high computation power.
- The predicted parameters (trained weights) give inference about the importance of each feature. The direction of association i.e. positive or negative is also given. So we can use Logistic Regression to find out the relationship between the features.
- This algorithm allows models to be updated easily to reflect new data, unlike Decision Tree or Support Vector Machine. The update can be done using stochastic gradient descent.

4.3.2.3 DISADVANTAGES

- Logistic Regression is a statistical analysis model that attempts to predict precise probabilistic outcomes based on independent features.
- On high dimensional datasets, this may lead to the model being over-fit on the training set, which means overstating the accuracy of predictions on the training set and thus the model may not be able to predict accurate results on the test set.
- This usually happens in the case when the model is trained on little training data with lots of features.
- So on high dimensional datasets, Regularization techniques should be considered to avoid over-fitting (but this makes the model complex).
- Very high regularization factors may even lead to the model being under-fit on the training data.
- Nonlinear problems can't be solved with logistic regression since it has a linear decision surface. Linearly separable data is rarely found in real world

scenarios. So the transformation of nonlinear features is required which can be done by increasing the number of features such that the data becomes linearly separable in higher dimensions.

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

4.3.3 ADABOOST ALGORITHM

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

4.3.3.1 ALGORITHM

1. Initially, Adaboost selects a training subset randomly.

2. It iteratively trains the Adaboost machine learning model by selecting the training set based on the accurate prediction of the last training.

3. It assigns the higher weight to wrong classified observations so that in the next iteration these observations will get the high probability for classification.

4. Also, it assigns the weight to the trained classifier in each iteration according to the accuracy of the classifier. The more accurate classifier will get high weight.

5. This process iterates until the complete training data fits without any error or until reached to the specified maximum number of estimators.

6. To classify, perform a "vote" across all of the learning algorithms you built

4.3.3.2 ADVANTAGES

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

4.3.3.3 DISADVANTAGES

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

4.3.4 XGBOOST ALGORITHM

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

Regularization: XG-boost has in-built L1 (Lasso Regression) and L2 (Ridge Regression) regularization which prevents the model from over fitting. That is why, XG-boost is also called regularized form of GBM (Gradient Boosting Machine). While using Scikit Learn library, we pass two hyper-parameters (alpha and lambda) to XG-boost related to regularization. Alpha is used for L1 regularization and lambda is used for L2 regularization.

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

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

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

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

4.3.4.1 WORKING

XGBoost is short for extreme gradient boosting. This method is based on decision trees and improves on other methods such as random forest and gradient boost. It works well with large, complicated datasets by using various optimization methods.

To fit a training dataset using XGBoost, an initial prediction is made.

XGBoost the Algorithm operates on decision trees, models that construct a graph that examines the input under various “if” statements (vertices in the graph). Whether the “if” condition is satisfied influences the next “if” condition and eventual prediction.

XGBoost the Algorithm progressively adds more and more “if” conditions to the decision tree to build a stronger model.

4.3.4.2 ADVANTAGES

1. High accuracy: XGBoost is known for its high accuracy, making it a popular choice for machine learning tasks that require high precision. It works by combining multiple decision trees to make more accurate predictions, making it effective for tasks such as image and speech recognition, natural language processing, and recommendation systems.

2. Speed: XGBoost is designed to be fast and efficient, even for large datasets. It is optimized for both single- and multi-core processing, making it an excellent choice for tasks that require fast predictions.

3. Regularization: XGBoost includes regularization techniques that help to prevent over fitting, which is a common problem in machine learning. It uses a combination of L1 and L2 regularization to reduce the complexity of the model, resulting in more robust and accurate predictions.

4. Flexibility: XGBoost is a flexible algorithm that can be used for a variety of machine-learning tasks, including classification, regression, and ranking. It is also compatible with a wide range of programming languages, including Python, R, and Java.

4.3.4.3 DISADVANTAGES

1. Complexity: XGBoost is a complex algorithm that requires some degree of technical expertise to implement and optimize effectively. It can be challenging to configure and tune the many hyper parameters that are involved, which can make it time-consuming to work with.

2. Overfitting: While XGBoost includes regularization techniques to prevent overfitting, it is still possible for the algorithm to overfit the training data. This can lead to inaccurate predictions of new data.

3. Memory usage: XGBoost can be memory-intensive, especially for large datasets. This can make it challenging to run on computers with limited memory, leading to slower performance.

4. Lack of transparency: XGBoost has often been considered a "black box" algorithm, which means that it can be difficult to interpret and understand how it arrives at its predictions. This can make it challenging to troubleshoot and fine-tune.

4.3.5 DECISION TREE

Decision Tree is a supervised learning technique that can be used for both classification and regression problems, but mostly it is preferred for solving classification problems. It is a tree-structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules and each

leaf node represents the outcome. In a Decision Tree, there are two nodes, which are the Decision Node and Leaf Node.

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

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

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

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

Decision Trees usually mimic human thinking ability while making a decision, so it is easy to understand.

The logic behind the decision tree can be easily understood because it shows a tree-like structure.

In Decision Tree the major challenge is to identify the attribute for the root node in each level. This process is known as attribute selection.

4.3.5.1 WE HAVE TWO POPULAR ATTRIBUTE SELECTION MEASURES:

1. Information Gain:

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

Entropy is the measure of uncertainty of a random variable, it characterizes the impurity of an arbitrary collection of examples.

The higher the entropy the more the information content.

2. Gini Index:

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

4.3.5.2 THE NOTABLE TYPES OF DECISION TREE ALGORITHMS ARE:

1. IDichotomiser 3 (ID3): This algorithm uses Information Gain to decide which attribute is to be used to classify the current subset of the data. For each level of the tree, information gain is calculated for the remaining data recursively.

2. C4.5: This algorithm is the successor of the ID3 algorithm. This algorithm uses either Information gain or Gain ratio to decide upon the classifying attribute. It is a direct improvement from the ID3 algorithm as it can handle both continuous and missing attribute values.

3. Classification and Regression Tree (CART): It is a dynamic learning algorithm which can produce a regression tree as well as a classification tree depending upon the dependent variable.

4.3.5.3 WORKING:

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

For the next node, the algorithm again compares the attribute value with the other sub-nodes and moves further. It continues the process until it reaches the leaf node of the tree.

4.3.5.4 THE COMPLETE PROCESS CAN UNDERSTOOD USING THE BELOW ALGORITHM:

- Step-1: Begin the tree with the root node, says S, which contains the complete dataset.
- Step-2: Find the best attribute in the dataset using Attribute Selection Measure (ASM).
- Step-3: Divide the S into subsets that contains possible values for the best attributes.
- Step-4: Generate the Decision Tree node, which contains the best attribute.

● Step-5: Recursively make new decision trees using the subsets of the dataset created in step -3. Continue this process until a stage is reached where you cannot further classify the nodes and call the final node as a leaf node.

4.3.5.5 ADVANTAGES

1. Compared to other algorithms decision trees requires less effort for data preparation during pre-processing.
2. A decision tree does not require normalization of data.
3. A decision tree does not require scaling of data as well.
4. Missing values in the data also do NOT affect the process of building a decision tree to any considerable extent.
5. A Decision tree model is very intuitive and easy to explain to technical teams as well as stakeholders.

4.3.5.6 DISADVANTAGES

1. A small change in the data can cause a large change in the structure of the decision tree causing instability.
2. For a Decision tree sometimes calculation can go far more complex compared to other algorithms.
3. Decision tree often involves higher time to train the model.
4. Decision tree training is relatively expensive as the complexity and time has taken are more.
5. The Decision Tree algorithm is inadequate for applying regression and predicting continuous values.

4.3.6 RANDOM FOREST ALGORITHM

Random Forest is a supervised learning algorithm. It is an extension of machine learning classifiers which include the bagging to improve the performance of Decision Tree. It combines tree predictors, and trees are

dependent on a random vector which is independently sampled. The distribution of all trees are the same. Random Forests splits nodes using the best among of a predictor subset that are randomly chosen from the node itself, instead of splitting nodes based on the variables. The time complexity of the worst case of learning with Random Forests is O (M (dnlogn)), where M is the number of growing trees, n is the number of instances, and d is the data dimension.

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

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

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

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

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

4.3.6.1 ASSUMPTIONS:

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

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

4.3.6.2 ALGORITHM STEPS

It works in four steps:

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

4.3.6.3 ADVANTAGES

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

4.3.6.4 DISADVANTAGES

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

4.3.7 STOCHASTIC GRADIENT DESCENT ALGORITHM

Batch methods, such as limited memory BFGS, which use the full training set to compute the next update to parameters at each iteration tend to converge very well to local optima. They are also straight forward to get working provided a good off the shelf implementation (e.g. minFunc) because they have very few hyper-parameters to tune. However, often in practice computing the cost and gradient for the entire training set can be very slow and sometimes intractable on a single machine if the dataset is too big to fit in main memory. Another issue with batch optimization methods is that they don't give an easy way to incorporate new data in an 'online' setting. Stochastic Gradient Descent (SGD) addresses both of these issues by following the negative gradient of the objective after seeing only a single or a few training examples. The use of SGD In the neural network setting is motivated by the high cost of running back propagation over the full training set. SGD can overcome this cost and still lead to fast convergence.

Generally each parameter update in SGD is computed w.r.t a few training examples or a minibatch as opposed to a single example. The reason for this is twofold: first this reduces the variance in the parameter update and can lead to more stable convergence, second this allows the computation to take advantage of highly optimized matrix operations that should be used in a well vectorized computation of the cost and gradient. A typical minibatch size is 256, although the optimal size of the minibatch can vary for different applications and architectures.

One final but important point regarding SGD is the order in which we present the data to the algorithm. If the data is given in some meaningful order, this can bias the gradient and lead to poor convergence. Generally a good method to avoid this is to randomly shuffle the data prior to each epoch of training.

The word ‘stochastic‘means a system or process linked with a random probability. Hence, in Stochastic Gradient Descent, a few samples are selected randomly instead of the whole data set for each iteration. In Gradient Descent, there is a term called “batch” which denotes the total number of samples from a dataset that is used for calculating the gradient for each iteration. In typical Gradient Descent optimization, like Batch Gradient Descent, the batch is taken to be the whole dataset. Although using the whole dataset is really useful for getting to the minima in a less noisy and less random manner, the problem arises when our dataset gets big.

Suppose, you have a million samples in your dataset, so if you use a typical Gradient Descent optimization technique, you will have to use all of the one million samples for completing one iteration while performing the Gradient Descent, and it has to be done for every iteration until the minima are reached. Hence, it becomes computationally very expensive to perform.

This problem is solved by Stochastic Gradient Descent. In SGD, it uses only a single sample, i.e., a batch size of one, to perform each iteration. The sample is randomly shuffled and selected for performing the iteration.

Stochastic Gradient Descent (SGD) is a variant of the Gradient Descent algorithm used for optimizing machine learning models. In this variant, only one random training example is used to calculate the gradient and update the parameters at each iteration. Here are some of the advantages and disadvantages of using SGD:

4.3.7.1 MOMENTUM

If the objective has the form of a long shallow ravine leading to the optimum and steep walls on the sides, standard SGD will tend to oscillate across the narrow ravine since the negative gradient will point down one of the steep sides rather than along the ravine towards the optimum. The objectives of deep architectures have this form near local optima and thus standard SGD can lead to very slow convergence particularly after the initial steep gains.

4.3.7.2 ADVANTAGES

- **Speed:** SGD is faster than other variants of Gradient Descent such as Batch Gradient Descent and Mini-Batch Gradient Descent since it uses only one example to update the parameters.
- **Memory Efficiency:** Since SGD updates the parameters for each training example one at a time, it is memory-efficient and can handle large datasets that cannot fit into memory.
- **Avoidance of Local Minima:** Due to the noisy updates in SGD, it has the ability to escape from local minima and converge to a global minimum.

4.3.7.3 DISADVANTAGES

- SGD is much faster but the convergence path of SGD is noisier than that of original gradient descent.
- This is because in each step it is not calculating the actual gradient but an approximation

4.3.8 KNN ALGORITHM

K Nearest Neighbor algorithm falls under the Supervised Learning category and is used for classification (most commonly) and regression. It is a versatile algorithm also used for imputing missing values and resampling datasets.

As the name (K Nearest Neighbor) suggests it considers K Nearest Neighbors (Data points) to predict the class or continuous value for the new Data point.

4.3.8.1 THE ALGORITHM'S LEARNING IS

1. Instance-based learning: Here we do not learn weights from training data to predict output (as in model-based algorithms) but use entire training instances to predict output for unseen data.
2. Lazy Learning: Model is not learned using training data prior and the learning process is postponed to a time when prediction is requested on the new instance.
3. Non -Parametric: In KNN, there is no predefined form of the mapping function.

4.3.8.2 HOW TO CHOOSE THE VALUE FOR K?

K is a crucial parameter in the KNN algorithm. Some suggestions for choosing K Value are:

1. Using error curves: The figure below shows error curves for different values of K for training and test data.

At low K values, there is overfitting of data/high variance. Therefore test error is high and train error is low. At K=1 in train data, the error is always zero, because the nearest neighbor to that point is that point itself. Therefore though training error is low test error is high at lower K values. This is called overfitting. As we increase the value for K, the test error is reduced.

2. Also, domain knowledge is very useful in choosing the K value.

3. K value should be odd while considering binary (two-class) classification.

4.3.8.3 HOW DOES K-NN WORK?

The K-NN working can be explained on the basis of the below algorithm:

Step-1: Select the number K of the neighbors

Step-2: Calculate the Euclidean distance of K number of neighbors

Step-3: Take the K nearest neighbors as per the calculated Euclidean distance.

Step-4: Among these k neighbors, count the number of the data points in each category.

Step-5: Assign the new data points to that category for which the number of the neighbour is maximum.

Step-6: Our model is ready.

4.3.8.4 ADVANTAGES

- It is simple to implement.
- It is robust to the noisy training data
- It can be more effective if the training data is large.

4.3.8.5 DISADVANTAGES

- Always needs to determine the value of K which may be complex some time.
- The computation cost is high because of calculating the distance between the data points for all the training samples.

CHAPTER 5

EXPERIMENTAL ANALYSIS

5.1 SYSTEM CONFIGURATION

5.1.1 HARDWARE REQUIREMENTS:

Processer: Any Update Processer

Ram: Min 4GB

Hard Disk: Min 100GB

5.1.2 SOFTWARE REQUIREMENTS:

Operating System: Windows

Technology: Python3.7

IDE: Jupiter notebook

5.2 DATASET DETAILS

• 76 attributes available in the dataset, 14 attributes are considered for the prediction of the output.

• Heart Disease UCI: https://archive.ics.uci.edu/ml/datasets/Heart+Disease.

• The dataset consists of 303 individual's data. There are 14 columns in the dataset, which are described below.

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

Figure 5.2.1 – Dataset Attributes

5.2.1 Input dataset attributes

- Gender (value 1: Male; value 0: Female)
- Chest Pain Type (value 1: typical type 1 angina, value 2: typical type angina, value 3: non-angina pain; value 4: asymptomatic)
- Fasting Blood Sugar (value 1: > 120 mg/dl; value 0 :< 120 mg/dl)
- Exang – exercise induced angina (value 1: yes; value 0: no)
- CA – number of major vessels collared by fluoroscopy (value 0 – 3)
- Thal (value 3: normal; value 6: fixed defect; value 7: reversible defect)
- Trest Blood Pressure (mm Hg on admission to the hospital)
- Thalach – maximum heart rate achieved
- Age in Year
- Cholesterol
- Restecg

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

Table 1 - Attributes of the dataset

5.3 PERFORMANCE ANALYSIS

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

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

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

5.3.1 CONFUSION MATRIX- It gives us a matrix as output and gives the total performance of the system

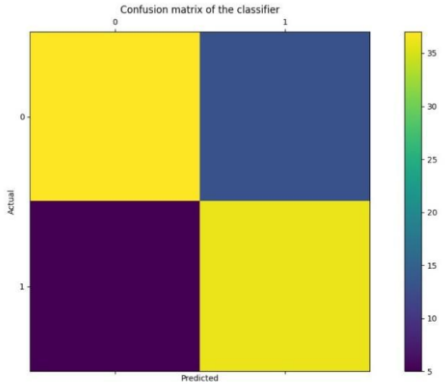

Figure 5.2 - Confusion Matrix

Where

TP: True positive

FP: False Positive

FN: False Negative

TN: True Negative

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

Figure 5.3 - Correlation Matrix

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

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

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

5.4 PERFORMANCE EVALUATION

5.4.1 EXISTING SYSTEM

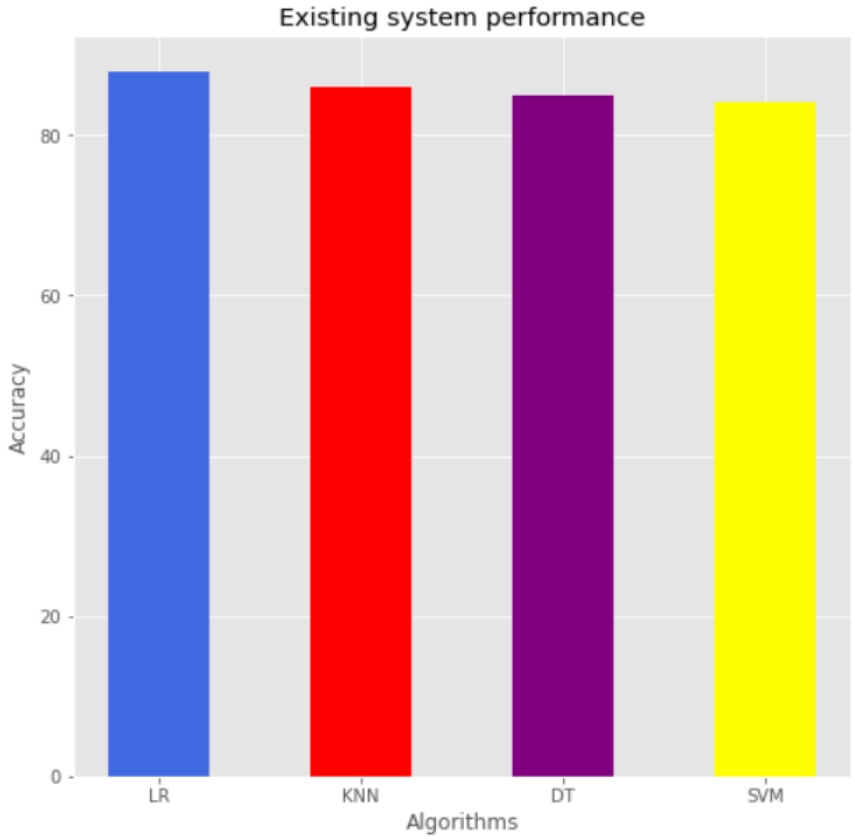

Figure 5.4 - Existing system performance

Logistic regression have the highest accuracy at 88 within all the machine learning algorithms . In comparison the KNN provides 86 accuracy . Decision tree provides 85 accuracy . SVM yields lowest accuracy at 84. After comparing the accuracy between machine learning algorithms with all physical indicators .

5.4.2 NEW SYSTEM

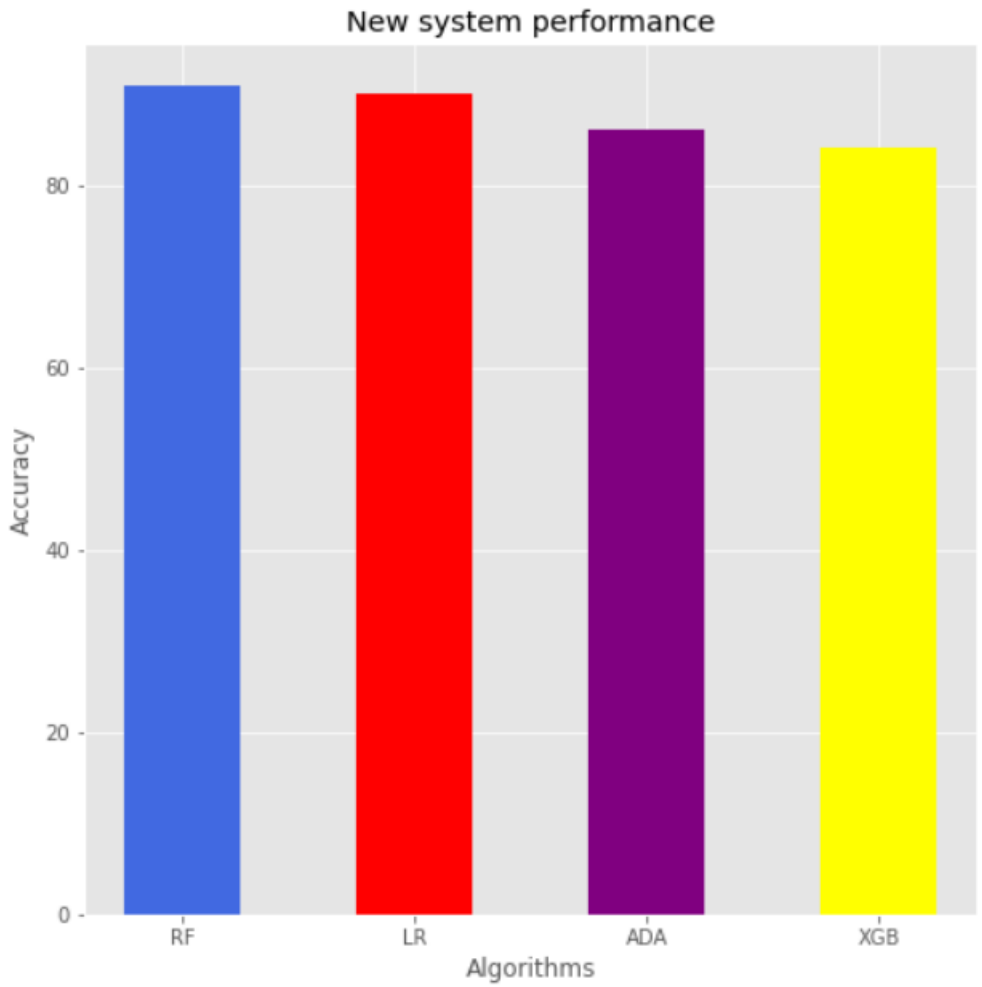

Figure 5.5 - New System Performance

We create a Machine Learning Model with different test size and random state to get better performance. Create a Random Forest model with test size = 0.2 provides highest accuracy at 91 along with other algorithms . Logistic Regression model with test size = 0.1 provides accuracy at 90 . It works quickly when compare to existing system .

CHAPTER 6

SYSTEM DESIGN

6.1 GENERAL

Design Engineering deals with the various UML [Unified Modeling language] diagrams for the implementation of project. Design is a meaningful engineering representation of a thing that is to be built. Software design is a process through which the requirements are translated into representation of the software. Design is the place where quality is rendered in software engineering. Design is the means to accurately translate customer requirements into finished product.

6.2 DATA 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 modelling methods use four kinds of symbols: Processes, Data stores, Dataflow and external entities. These symbols are used to represent four kinds of system components. Circles in DFD represent processes. Data Flow represented by a thin line in the DFD and each data store has a unique name and square or rectangle represents external entities

Level 0

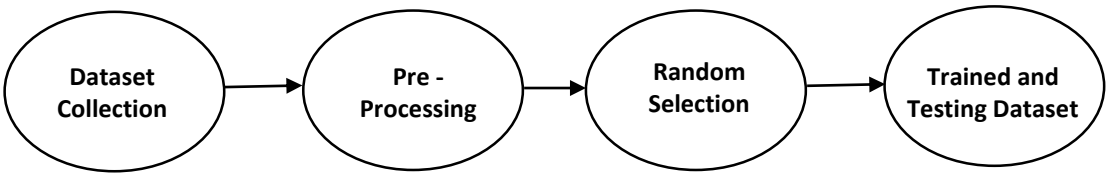

Figure 6.1 – Data Flow Diagram Level 0

Level 1

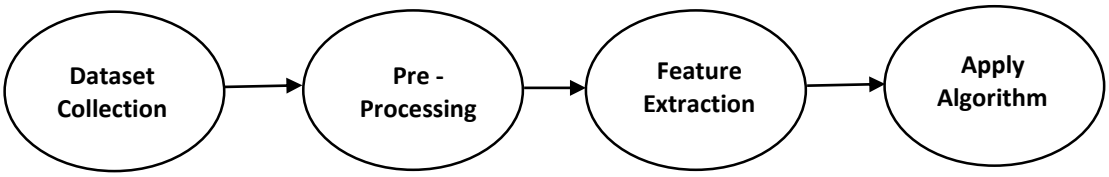

Figure 6.2 – Data Flow Diagram Level 1

6.3 USE-CASE DIAGRAM

A use case diagram is a diagram that shows a set of use cases and actors and their relationships. A use case diagram is just a special kind of diagram and shares the same common 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 of diagrams is its particular content.

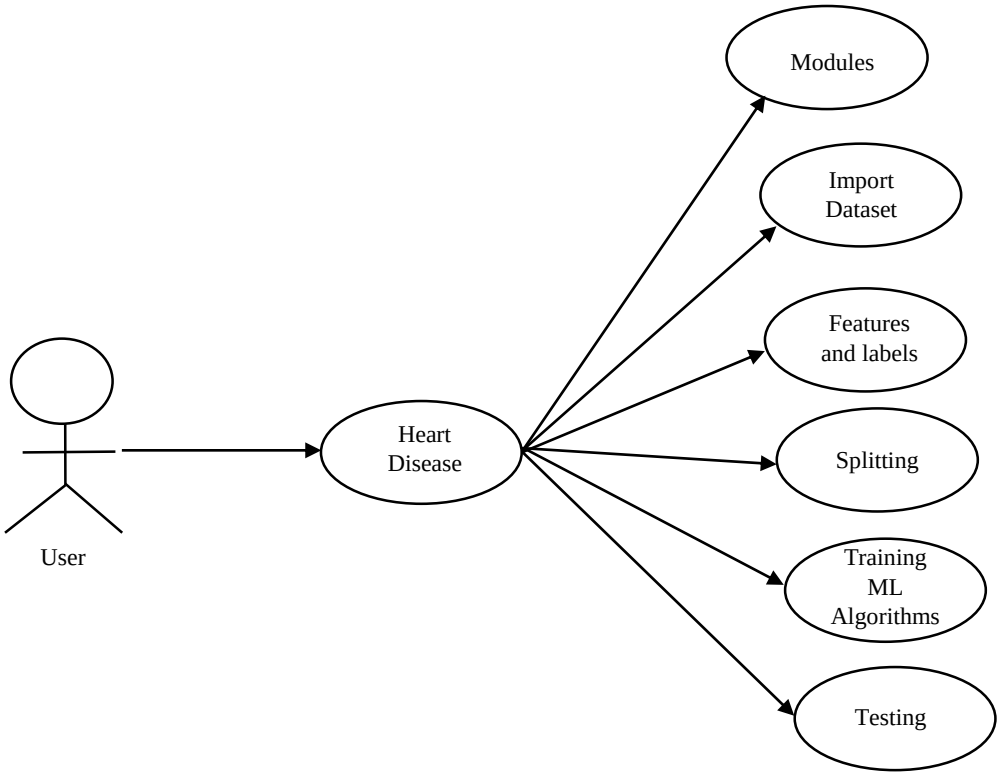

Figure 6.3 – Use Case Diagram

6.4 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 the elements found in an activity graph, a special case of a state machine in which all or most states are activity states and in which all or most transitions are triggered by completion of activities in the source.

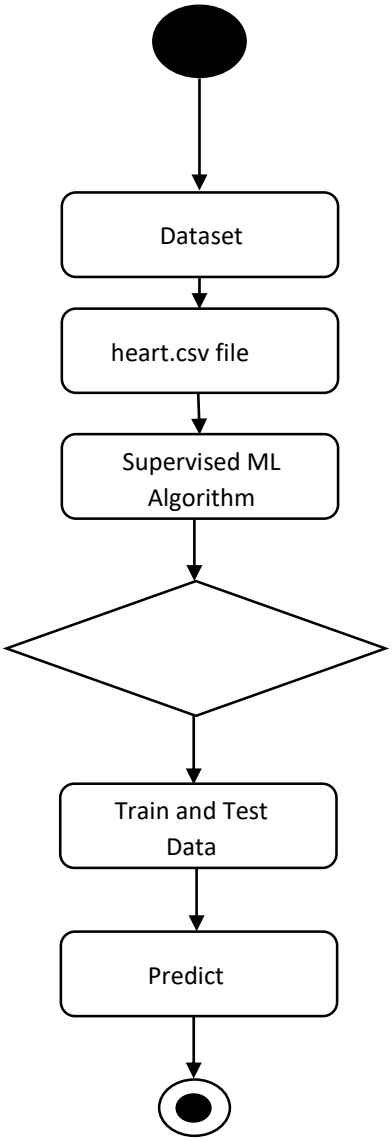

Figure 6.4 – Activity Diagram

6.5 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 those objects. The objects are typically named or anonymous instances of classes, but may also represent instances of other things, such as collaborations, components, and nodes. We use sequence diagrams to illustrate the dynamic view of a system.

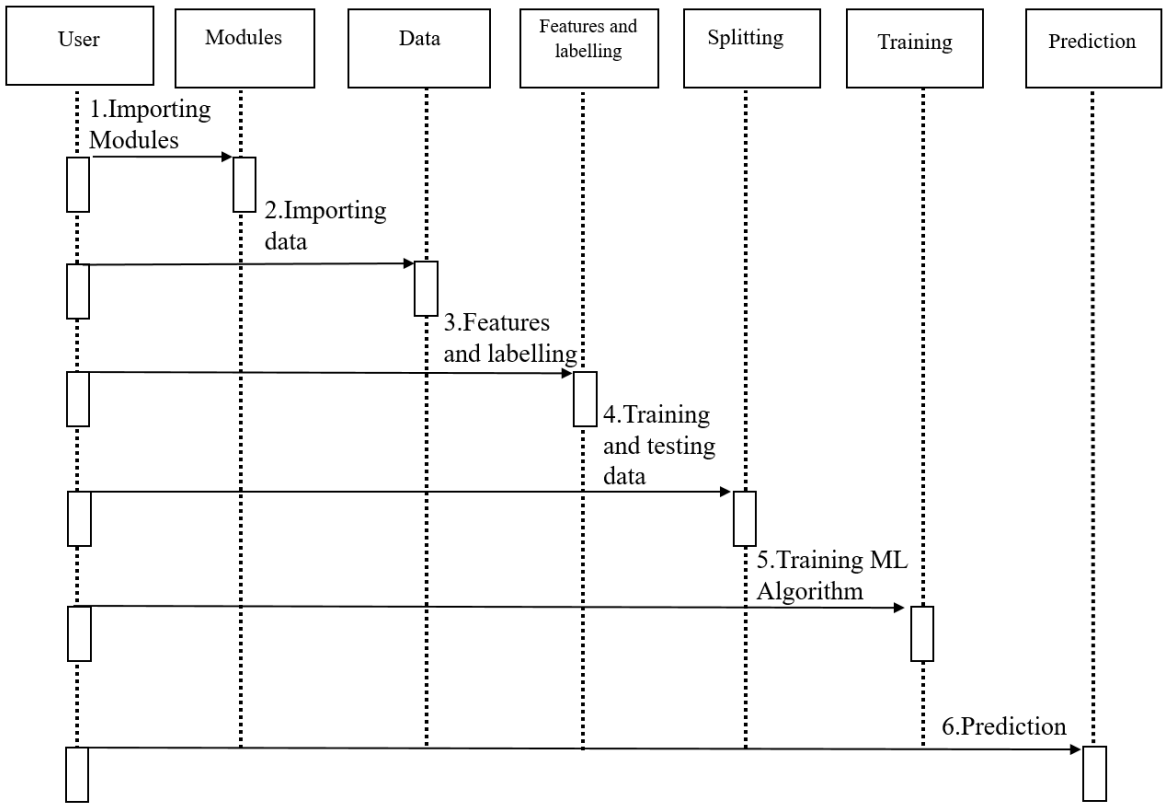

Figure 6.5 Sequence Diagram

6.6 CLASS DIAGRAM

A Class diagram in the Unified Modeling Language (UML)is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects. It provides a basic notation for other structure diagrams prescribed by UML. It is helpful for developers another team members too.

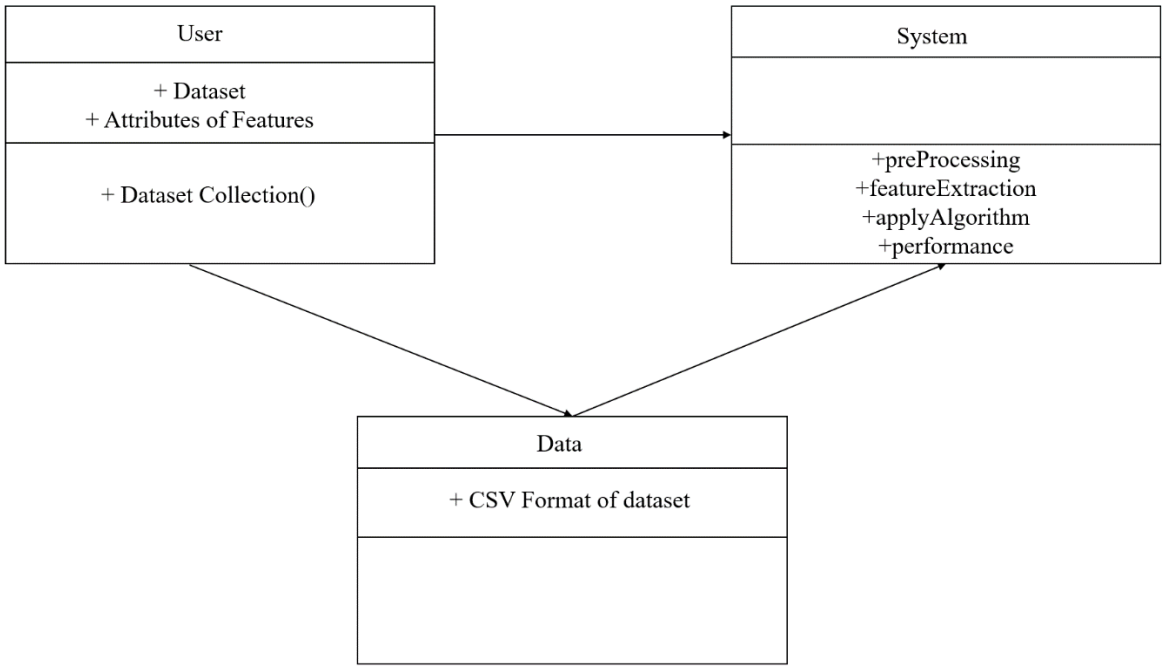

Figure 6.6 Class Diagram

CHAPTER 7

CONCLUSION

7.1 CONCLUSION

Heart diseases are a major killer in India and throughout the world, application of promising technology like machine learning to the initial prediction of heart diseases will have a profound impact on society. The early prognosis of heart disease can aid in making decisions on lifestyle changes in high-risk patients and in turn reduce the complications, which can be a great milestone in the field of medicine. This prompts for its early diagnosis and treatment. The utilization of suitable technology support in this regard can prove to be highly beneficial to the medical fraternity and patients. In this paper, the eight different machine learning algorithms used to measure the performance are SVM, Decision Tree, Random Forest, KNN, Logistic Regression, Adaptive Boosting, Stochastic Gradient Descent and Extreme Gradient Boosting applied on the dataset.

The expected attributes leading to heart disease in patients are available in the dataset which contains14 important features that are useful to evaluate the system are selected among them. If all the features taken into the consideration then the efficiency of the system the author gets is less. To increase efficiency, attribute selection is done.

All the eight machine learning methods accuracies are compared based on which one prediction model is generated. Hence, the aim is to use various evaluation metrics like confusion matrix, accuracy, precision, recall, and f1-score which predicts the disease efficiently. Comparing all eight the Random Forest gives the highest accuracy of 91%.

7.2 APPENDIX

heart_dis.py

```
import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

from sklearn.model_selection import train_test_split

from sklearn.neighbors import KNeighborsClassifier

from sklearn.svm import SVC

from sklearn.tree import DecisionTreeClassifier

from sklearn.ensemble import RandomForestClassifier

from sklearn.linear_model import LogisticRegression

from sklearn.ensemble import  AdaBoostClassifier

from xgboost import XGBClassifier

from sklearn.linear_model import  SGDClassifier

from sklearn.metrics import confusion_matrix

from sklearn.metrics import accuracy_score

from sklearn.metrics import classification_report

df = pd.read_csv('heart.csv')

print("First 5 rows \n")

print(df.head())

print("\n")
```

```
print("Last 5 rows \n")
print(df.tail())
print("\n")
print("Shape of the dataframe: \n")
print(df.shape)
print("\n")
print("Information of the dataframe: \n")
print(df.info())
print("\n")
print("Number of null count \n")
print(df.isnull().sum())
print("\n")
print(df["ca"].value_counts())
print("Splitting Independent and Dependent variable \n")
X = df.drop(["target"],axis = 1)
Y  = df["target"]
print(f"X shape is {X.shape}\nY shape is {Y.shape}")
X_train , X_test , Y_train , Y_test = train_test_split(X , Y , test_size = 0.2)
print("Without random state")
print(X_train)
print("With random state ")
```

```
X = df.drop(["target"],axis = 1)

Y  = df["target"]

X_train , X_test , Y_train , Y_test = train_test_split(X , Y , test_size = 0.2, random_state = 2)

print(X_train)

print("Logistic Regression Algorithm")

for k in range(1,40):

    lr = LogisticRegression(random_state = k)

    lr.fit(X_train , Y_train)

    lr_scores.append(lr.score(X_test , Y_test))

print(f"Best choice of k:{np.argmax(lr_scores)+1}")

lr2 = LogisticRegression(random_state = 1)

print(lr2.fit(X_train , Y_train))#fit is used to train the model

from sklearn.metrics import confusion_matrix

te_score = lr2.score(X_test , Y_test)

print(f"\n Testing accuracy : {te_score}")

ypred= lr2.predict(X_test)#model is ready to predict

cm = confusion_matrix(Y_test,ypred)

print("Confusion matrix:\n" , cm )

from sklearn.metrics import classification_report

print(classification_report(Y_test,ypred))
```

```
print("Accuracy :" , accuracy_score(Y_test , ypred))
print("\n")
print("Decision Tree Algorithm")
dt_scores  = []
for k in range(1,40):
    dt = DecisionTreeClassifier(random_state = k)
    dt.fit(X_train , Y_train)
    dt_scores.append(dt.score(X_test , Y_test))
print(f"Best choice of k:{np.argmax(dt_scores)+1}")
dt2 = DecisionTreeClassifier(random_state = 2)
print(dt2.fit(X_train , Y_train))#fit is used to train the model
te_score1iii = dt2.score(X_test , Y_test)
print(f"\n Testing accuracy : {te_score1iii}")
ypred1iii = dt2.predict(X_test)
cm1iii = confusion_matrix(Y_test,ypred1iii)
print("Confusion matrix:\n" , cm1iii )
print(classification_report(Y_test,ypred1iii))
print("Accuracy :" , accuracy_score(Y_test , ypred1iii))
print("Random Forest Algorithm")
rf_scores  = []
for k in range(1,40):
```

```
    rf = RandomForestClassifier(random_state = k)
    rf.fit(X_train , Y_train)
    rf_scores.append(rf.score(X_test , Y_test))
print(f"Best choice of k:{np.argmax(rf_scores)+1}")
rf2 =  RandomForestClassifier(random_state = 4 )
print(rf2.fit(X_train , Y_train))#fit is used to train the model
te_score1iv = rf2.score(X_test , Y_test)
print(f"\n Testing accuracy : {te_score1iv}")
ypred1iv = rf2.predict(X_test)
cm1iv = confusion_matrix(Y_test,ypred1iv)
print("Confusion matrix:\n" , cm1iv )
print(classification_report(Y_test,ypred1iv))
print("Accuracy :" , accuracy_score(Y_test , ypred1iv))
from sklearn.ensemble import  AdaBoostClassifier
print("Ada Algorithm")
ad_scores  = []
for k in range(1,40):
    ad = AdaBoostClassifier(random_state =k)
    ad.fit(X_train , Y_train)
    ad_scores.append(ad.score(X_test , Y_test))
print(f"Best choice of k:{np.argmax(ad_scores)+1}")
```

7.3 SNAPSHOT

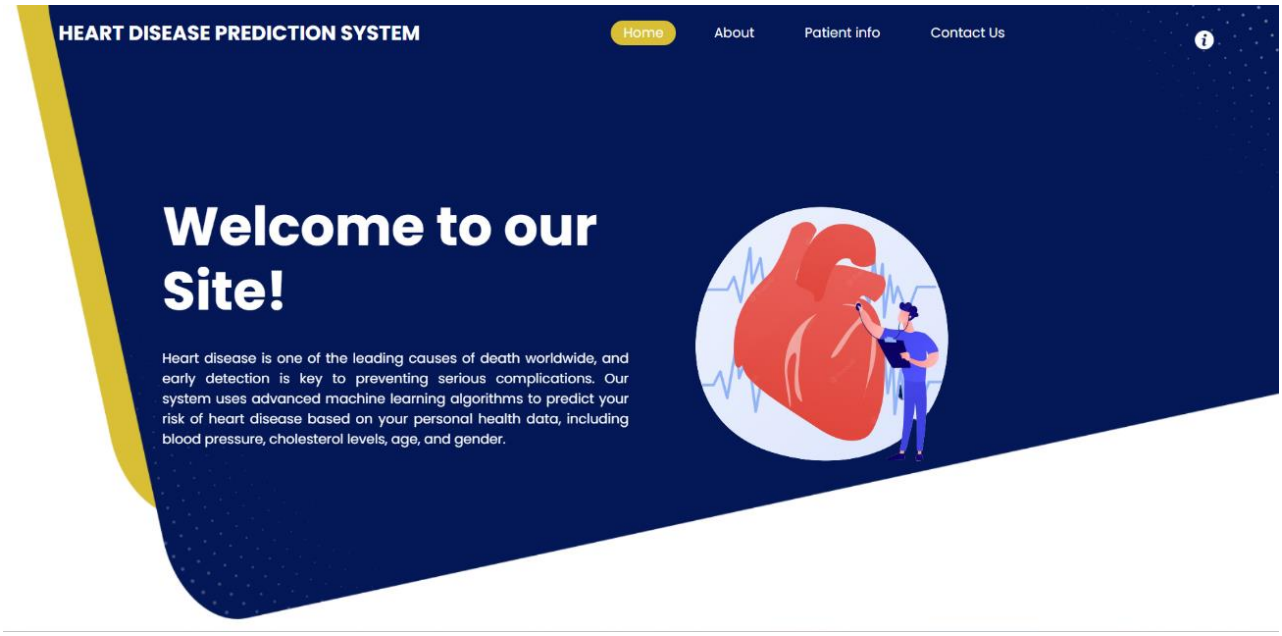

Figure 7.1 – Home Page

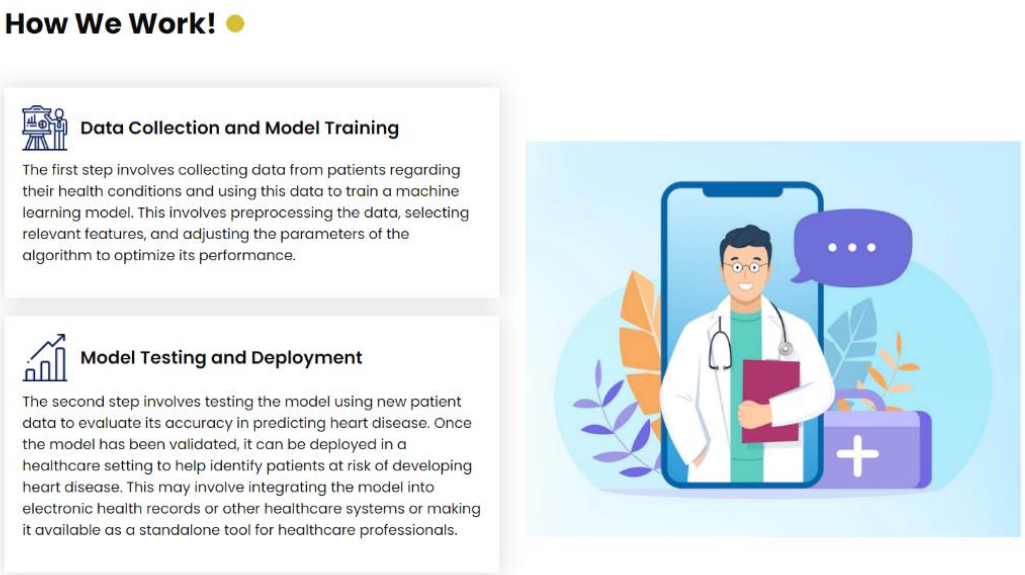

Figure 7.2 – Working Page

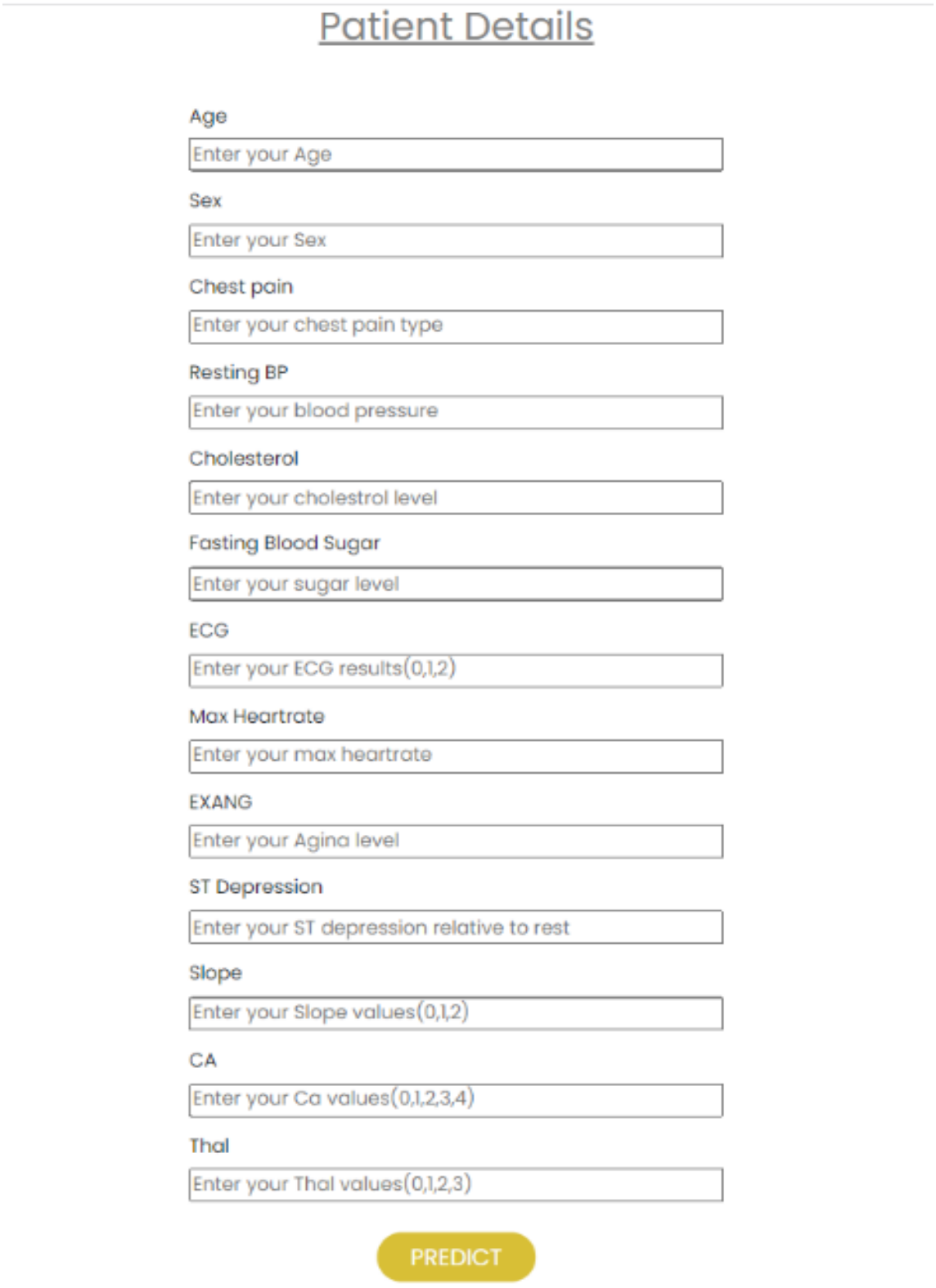

Figure 7.3– Predict Page

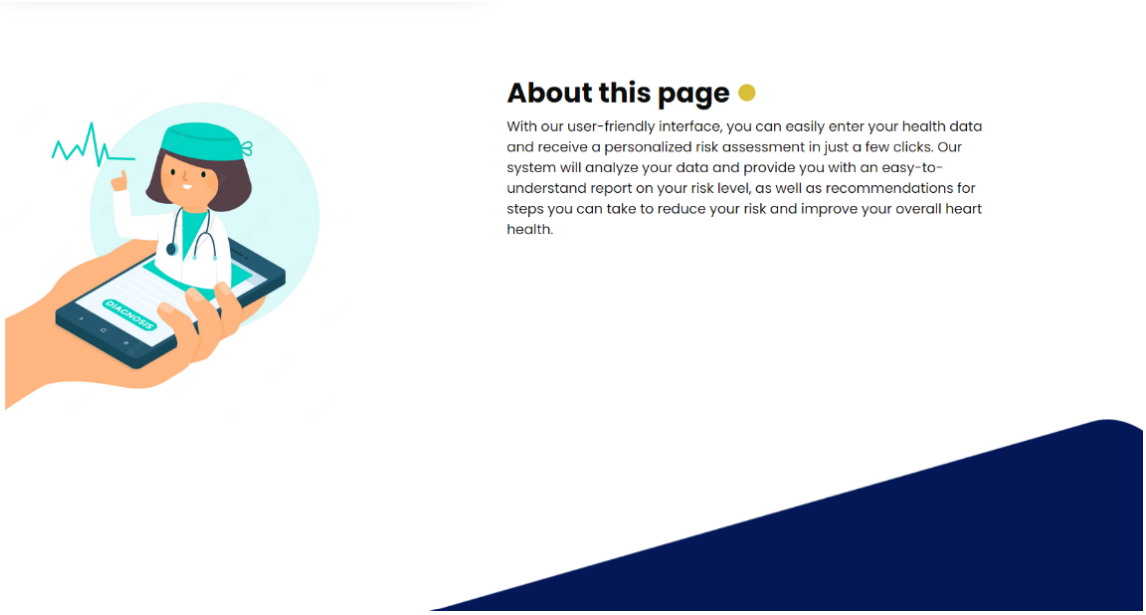

Figure 7.4 – About Page

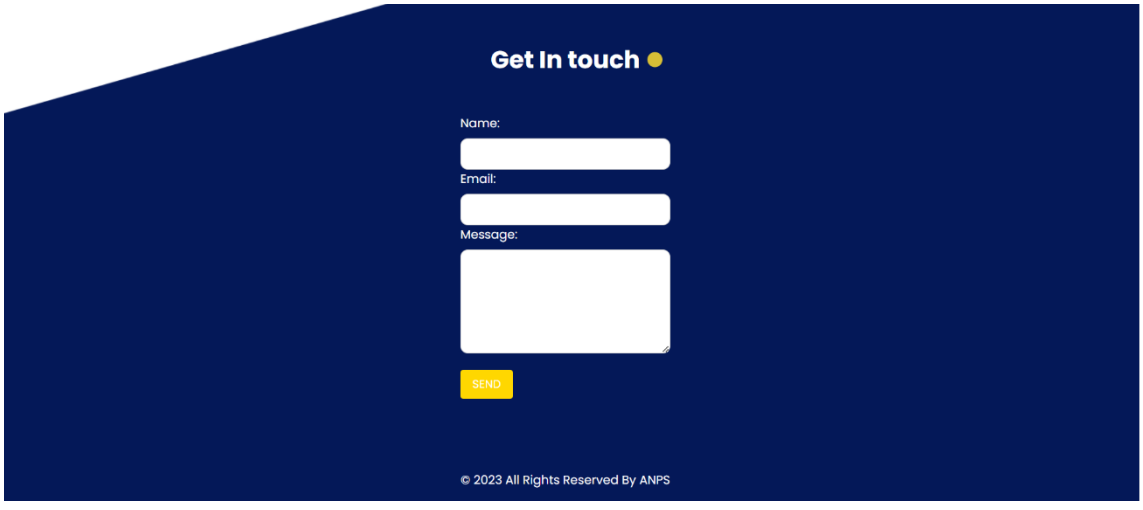

Figure 7.5 – Contact Page

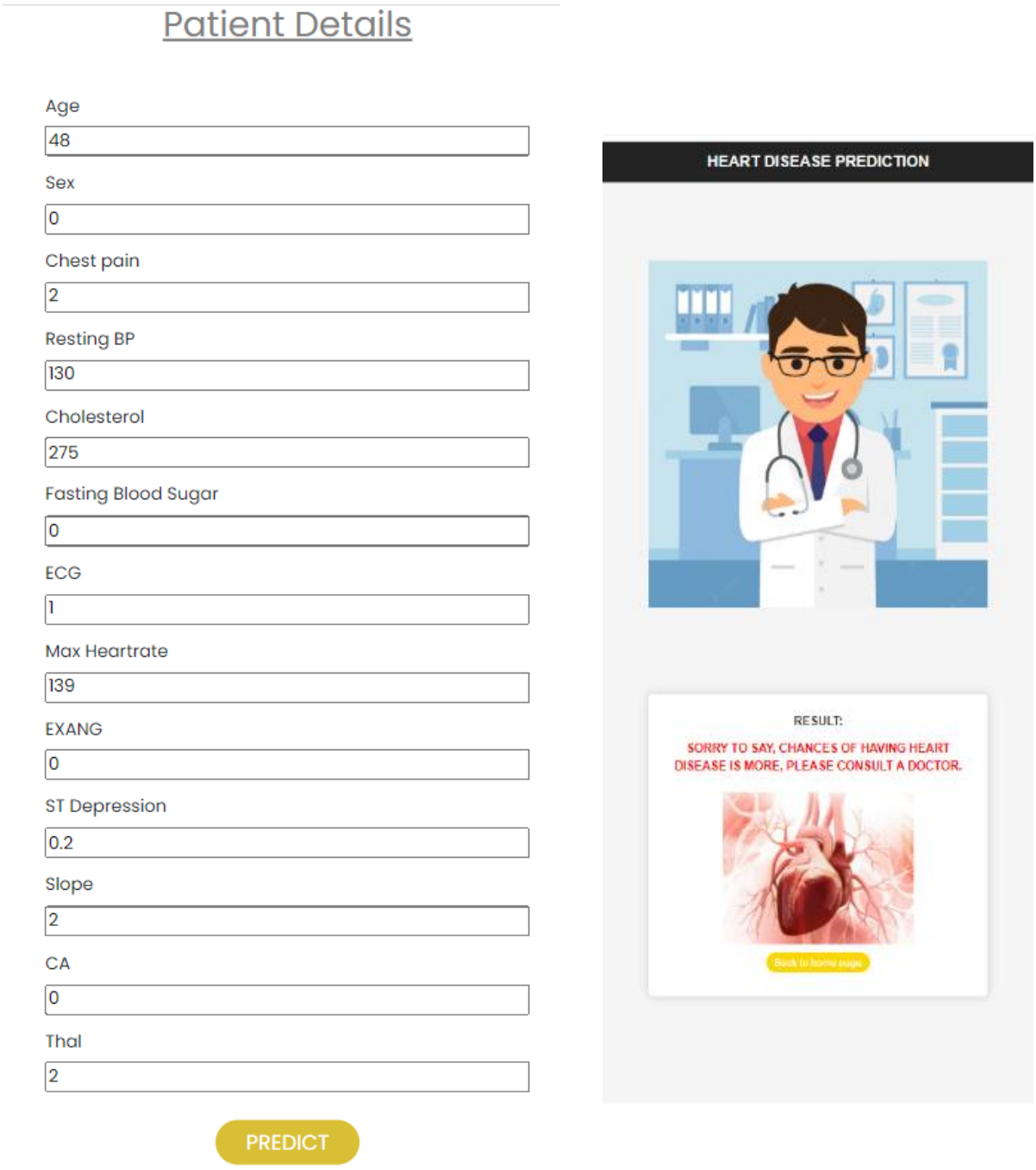

Figure 7.6.1 –Result Page

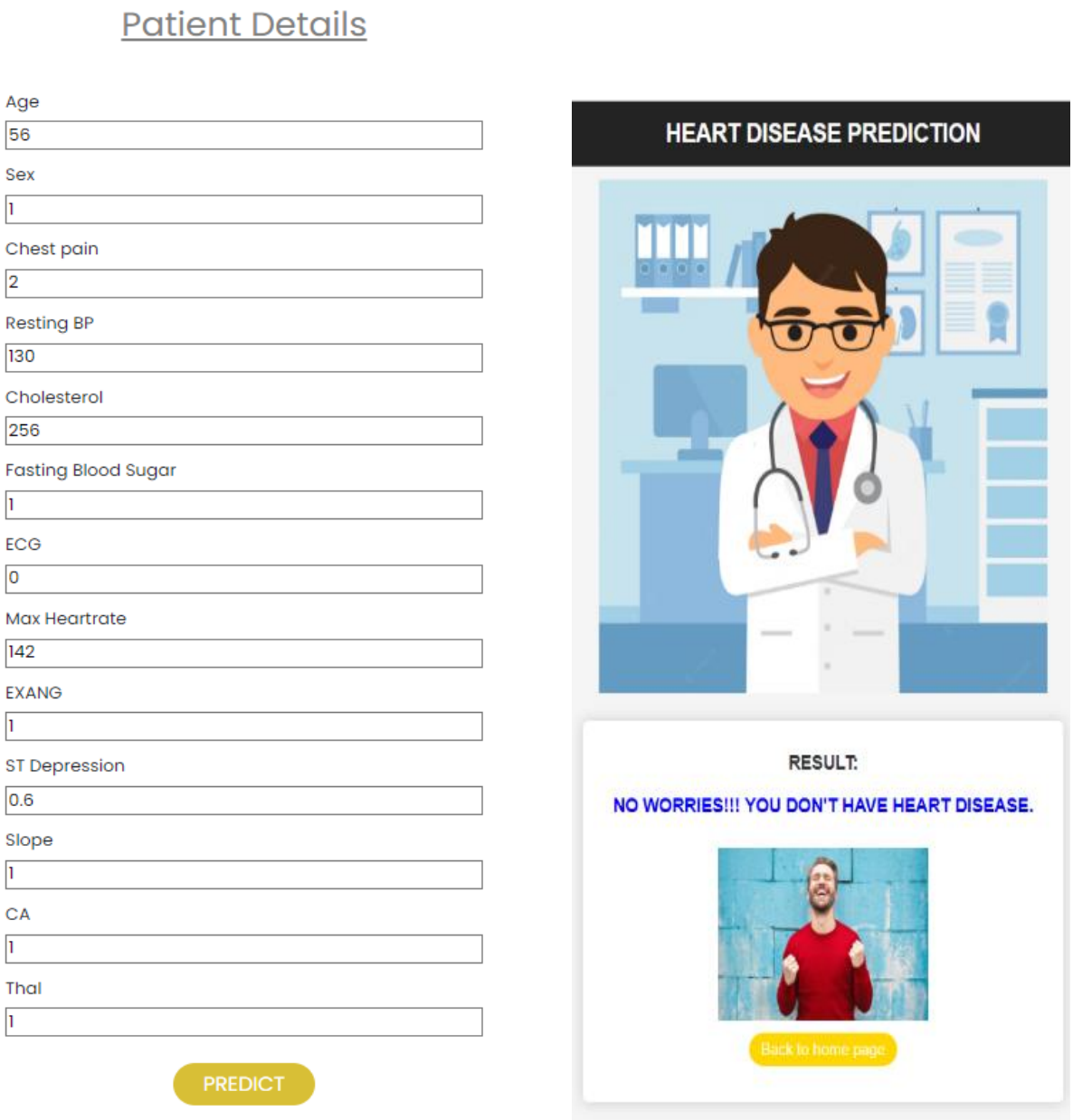

Figure 7.6.2 – Result Page

CHAPTER 8

REFERENCE

8.1 REFERENCE

[1] Akkaya, B., Sener, E. and Gursu, C. (2022) A Comparative Study of Heart Disease Prediction Using Machine Learning Techniques. 2022 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA), Ankara,9-11 June 2022.https://doi.org/10.1109/HORA55278.2022.9799978.

[2] Xing, Y.W., Wang, J., Zhao, Z.H. and Gao, Y.H. (2007) Combination Data Mining Methods with New Medical Data to Predicting Outcome of Coronary Heart Disease.Convergence Information Technology, Gwangju, 21-23 November 2007, 868-872.https://doi.org/10.1109/ICCIT.2007.204.

[3] Nahar, J., Imam, T., Tickle, K.S. and Chen, Y.P.P. (2013) Computational Intelligence for Heart Disease Diagnosis: A Medical Knowledge Driven Approach. Expert Systems with Applications , 40, 96-104.https://doi.org/10.1016/j.eswa.2012.07.032.

[4] Desai, F., Chowdhury, D., Kaur, R., Peeters, M., Arya, R.C., Wander, G.S., Gill, S.S. and Buyya, R. (2022) HealthCloud: A System for Monitoring Health Status of Heart Patients Using Machine Learning and Cloud Computing. Internet of Things , 17, Article Id : 100485.https://doi.org/10.1016/j.iot.2021.100485.

[5] Nahar, J., Imam, T., Tickle, K.S. and Chen, Y.-P.P. (2013) Association Rule Mining to Detect Factors Which Contribute to Heart Disease in Males and Females. Expert Systems with Applications , 40, 1086-1093.https://doi.org/10.1016/j.eswa.2012.08.028.

[6] Wilson, P.W.F., D'Agostino, R.B., Levy, D., Belanger, A.M., Silbershatz, H. and Kannel, W.B. (1998) Prediction of Coronary Heart Disease Using Risk Factor Categories.Circulation, 97, 1837-1847.https://doi.org/10.1161/01.CIR.97.18.1837.

[7] Liu, X., Wang, X.L., Su, Q., Zhang, M., Zhu, Y.H., Wang, Q.G. and Wang, Q. (2017) A Hybrid Classification System for Heart Disease Diagnosis Based on the RFRS Method. Computational and Mathematical Methods in Medicine , 2017, 1-11.https://doi.org/10.1155/2017/8272091.

[8] Makino, K., Lee, S., Bae, S., Chiba, I., Harada, K., Katayama, O., Shinkai,Y. and Shimada, H. (2021) Absolute Cardiovascular Disease Risk Assessed in Old Age Predicts Disability and Mortality: A Retrospective Cohort Study of Community-Dwelling Older Adults. Journal of the American Heart Association, 10, e022004.https://doi.org/10.1161/JAHA.121.022004.

[9] Nagaraj M Lutimath,Chethan C,Basavaraj S Pol.,Prediction Of Heart Disease using Machine Learning, International journal Of Recent Technology and Engineering,8,(2S10), pp 474-477, 2019.DOI:10.35940/ijrte.B1081.0982S1019.

[10] Fahd Saleh Alotaibi, Implementation of Machine Learning Model to Predict Heart Failure Disease, (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 10, No. 6, 2019.Digital Object Identifier:10.14569/IJACSA.2019.0100637.

[11] UCI, Heart Disease Data Set.[Online]. Available (Accessed on May 1 2020): https://www.kaggle.com/ronitf/heart-disease-uci.

[12] Jafar Alzubi, Anand Nayyar, Akshi Kumar. "Machine Learning from Theory to Algorithms: An Overview", Journal of Physics: Conference Series, 2018.