HEART DISEASE PREDICTION USING MACHINE LEARNING

19CSPN6601 INNOVATIVE AND CREATIVE PROJECT

Submitted by

SHANGAVIE G	**(19BCS089)**
NIKESH R	**(19BCS093)**
VEERARAGHAVAN V	**(19BCS107)**

in partial fulfillment for the award of the degree

of

Bachelor of Engineering

in

Computer Science and Engineering

Dr. Mahalingam College of Engineering and Technology

Pollachi – 642 003

An Autonomous Institution

Affiliated to Anna University, Chennai - 600 025

June 2022

Dr. Mahalingam College of Engineering and Technology
Pollachi - 642003
An Autonomous Institution
Affiliated to Anna University, Chennai - 600 025

BONAFIDE CERTIFICATE

Certified that this mini project report, “HEART DISEASE PREDICTION USING MACHINE LEARNING” is the bonafide work of

SHANGAVIE G	**(19BCS089)**
NIKESH R	**(19BCS093)**
VEERARAGHAVAN V	**(19BCS107)**

who carried out the project work under my supervision.

Mr. K. Prabhu
SUPERVISOR
Assistant Professor
Computer Science and Engineering
Dr. Mahalingam College of Engineering
and Technology, NPT-MCET Campus
Pollachi – 642 003 India

Dr.G.Anupriya
HEAD OF THE DEPARTMENT
Professor
Computer Science and Engineering
Dr. Mahalingam College of Engineering
and Technology, NPT-MCET Campus
Pollachi – 642 003 India

Submitted for the Autonomous End Semester Examination Mini Project viva-voce held on ____________

INTERNAL EXAMINER **EXTERNAL EXAMINER**

HEART DISEASE PREDICTION
USING MACHINE LEARNING

ABSTRACT

The Heart Disease Prediction Using Machine Learning is completely done with the help of Machine Learning algorithms and Python Programming language with and also using the dataset that's available previously by the hospitals using that the system will predict the disease. This model is developed using classification algorithms, as they play important role in prediction. The proposed work predicts the chances of Heart Disease and classifies patient's risk levels by implementing different machine learning techniques such as Logistic Regression, Random Forest, Support vector machine, Gaussian Naive Bayes, Gradient boosting, K-Nearest Neighbor, Naive bayes and Decision trees.

ACKNOWLEDGEMENT

First and foremost, we wish to express our deep unfathomable feeling, gratitude to our institution and our department for providing us a chance to fulfill our long cherished dreams of becoming Computer Science Engineers.

We express our sincere thanks to our honorable Secretary **Dr. C. Ramaswamy** for providing us with required amenities.

We wish to express our hearty thanks to **Dr. A. Rathinavelu,** Principal of our college, for his constant motivation and continual encouragement regarding our project work.

We are grateful to **Dr. G. Anupriya,** Head of the Department, Computer Science and Engineering, for her direction delivered at all times required. We also thank her for her tireless and meticulous efforts in bringing out this project to its logical conclusion.

Our hearty thanks to our guide **Mr. K. Prabhu,** Assistant Professor for his constant support and guidance offered to us during the course of ourproject by being one among us and all the noble heart that gave as immense encouragement towards the completion of our project.

We also thank our Project Coordinators **Dr. N. Suba Rani**, Associate Professor and **Ms. B. Suganya**, Assistant Professor for their continuous support and guidance.

LIST OF ABBREVIATIONS

CP	Chest Pain
ECG	Electro Cardio Graphy
FBS	Fasting Blood Sugar
KNN	K – Nearest Neighbor
LIBSVM	Library for Support Vector Machine
LMT	Logistic Model Tree
LR	Logistic Regression
MATLAB	Matrix Laboratory
RF	Random Forest
SVM	Support Vector Machine

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

Heart disease is considered as one of the fatal diseases across the globe. From the past few years, millions of cases are increasing and many people died due to heart related issues. According to a report conducted by the World Health Organization, heart disease is responsible for 17 million deaths worldwide. As heart is important organ in the human body, any issue related to it highly affects human health. The main symptoms of heart disease are chest pain, bloating, swollen legs, breathing issues, fatigue and irregular heart beat rhythm. The factors that cause heart disease are age, overweight, stress, unhealthy diet and smoking. The main goal of the System is to create a model to predict heart disease using machine learning algorithms, which will aid doctors in detecting the disease early on with less medical tests and providing appropriate care, potentially saving many lives. There is a traditional approach to identify heart disease in hospitals then why machine learning? In hospitals, large amount of data related to patients suffering from heart diseases and other diseases is generated each day, it is difficult for doctors to use or handle the patient’s data efficiently to make decisions without datamining techniques. Data mining is highly recommended for the prediction of heart diseases as it extracts more accurate and useful data from large amount of data which makes prediction easy. It is primary foundation of machine learning that helps to handles large amount of data, the processing speed of machine learning is high and it makes predictions in early stages. There are different datamining techniques can be used such as classification, prediction and recognizing patterns for diagnosing heart disease. In this system, classification models which are part of machine learning are used for identifying cardio vascular diseases. Classification algorithms uses input data to predict and classify them to which class or category that the data belongs. Some of the classification techniques are Logistic regression, Decision trees, Random Forest, Support vector machine, Naive Bayes and K-Nearest Neighbour. In the system, all classification models are trained to predict the heart disease and compare the performance of them using evaluation metrics such as sensitivity, accuracy and so on, which gives the best classification model for prediction of occurrence of heart disease.

1.1 Problem Statement

Heart disease is one of the critical health issues and many people across the world are suffering with this disease. It is important to identify this disease in early stages to save many lives. The purpose of this system is to design a model to predict the heart diseases using machine learning techniques. This model is developed using classification algorithms, as they play important role in prediction. The model is developed using different classification algorithms which include Logistic Regression, Random Forest, Support vector machine, Gaussian Naive Bayes, Gradient boosting, K-Nearest Neighbours, Multinomial Naïve bayes and Decision trees. Cleveland data repository is used to train and test the classifiers. In addition to this, feature selection algorithm named chi square is used to select key features from the input data set, which will decrease the execution time and increases the performance of the classifiers. Out of all the classifiers evaluated using performance metrics, Random forest is giving good accuracy. So, the model built using Random forest is efficient and feasible solution in identifying heart diseases and it can be implemented in healthcare which plays key role in the stream of cardiology.

1.2 Objective

The main objective of this model is to get a better accuracy to detect the heart-disease using algorithms in which the target output counts that a person having heart disease or not. This in turn will help to provide effective treatment to patients and avoid severe consequences.

1.3 Machine Learning Model

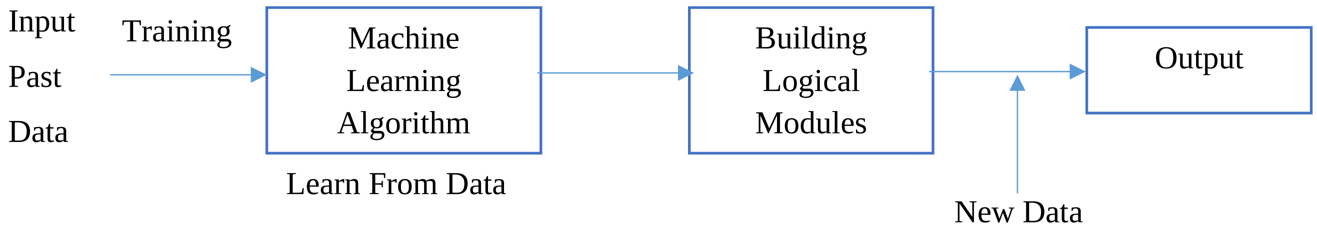

Fig.1. Phases of Machine Learning Model

Machine learning is a part of artificial intelligence which can learn by itself and improve from past experiences, makes decisions and predictions. According to Fig 1In this model, training and testing is conducted by using Cleveland data set. Initially, the system trains the classification algorithm using the input dataset features, then the model learns from the input data and able to find or recognize the patterns from dataset, then by testing with new data it makes the prediction that represents to which class the data belongs to. The model built using this technique learn from the data and classify the patients into normal and heart disease category and predicts the heart disease.

1.4 Significance

The modern lifestyle or fast forward life has significant impact on lives of people. Many people across the globe are suffering from heart related diseases due to stress, lifestyle habits and some from genes irrespective of their age. The objective of this article is to predict the occurrence of heart disease in early stages, so that it will help doctors to take proactive measures to control many lives of the people. Although there is traditional approach in hospitals for dealing with heart issues but most of them are re-active, they can only know after its occurrence and hospitals cannot handle huge amount of data that generates each day related to patient's data, so it becomes difficult for doctors to make accurate predictions. In the system, machine learning techniques are used for prediction as they can handle huge data and their performance is high. Classification models that are part of machine learning techniques are used, as they play key role in prediction. In this project, Random Forest algorithm along with some other classification algorithms are used to predict the heart disease. The purpose of this project is to reduce the deaths caused due to heart diseases and to predict its occurrence efficiently.

1.5 Comparision of Existing System

Ref No	Author	Technique	Dataset	Accuracy	Limitations
[1]	Jayshril S. Sonawane	Multilayer Perception Neural Networks	Cleveland heart disease database	97.5%	Independently trained subnetworks scale well because the learning time of multilayer perceptron networks with backpropagation

					scales exponentially for complex boolean functions by using minimal training sets.
[2]	Ketut Agung Enrico	K-Nearest Neighbours	Hungarian dataset	81.85%	Using KNN , with increase of number of parameters the performance decreases and it considers 90% of data for training which is computationally expensive and does nothing during training phase.
[3]	M.Akhil jabbar	Lazy association classification	UCI repository	80%	The vast amount of space used to store the entire data collection. Since no abstraction is made during the training phases, especially noisy training data increases the case base unnecessarily.
[4]	Jaymin Patel	Data mining technique Decision Tree model (J48, Logistic model tree LMT, Random forest)	Cleveland dataset	56.76%	The drawback of J48 is that the tree increases linearly with large data. LMT is slower and takes long time for implementation and accuracy is low
[5]	Rifki wijaya	Artificial Neural Network	Diff tools or database	100%	To function, the neural network needs to be trained. Big neural networks necessitate a long processing time. Microprocessor design

					and history necessitate their emulation.
[6]	Carlos Ordonez	Associatio n rules	Medical dataset from hospital	70%	It uses too many parameters from patients record and produce irrelevant rules. So, this technique is computationally expensive and performance is low.
[7]	Jyoti soni	Weighted Associative classifier	UCI machine learning dataset	81.51%	All attributes are not equally important in predicting the class mark in the prediction model. As a result, different weights can be assigned to different attributes depending on their predictive performance.
[8]	Idticeme sedjelmaci	Fractal dimension and chaos theory	Hospital database	80%	The drawbacks of applying Chaos Theory are primarily due to the input parameters chosen. The underlying dynamics of the data, as well as the type of analysis being done, which is usually complex and not always accurate, determine the methods used to measure these parameters.
[9]	D.R.Patil	Learning Vector Quantization Algorithm	Cleveland heart disease database	85.55%	There is no limitation on how many prototype can be used per class, the only requirement

					being that there is at least 1 for each class.
[10]	AH CHEN	Artificial neural network algorithm	ML UCI repository	80%	It exhibits black box nature, which doesn’t give information about how much time required for prediction, amount of data required. It is computationally expensive.
[11]	M. Anbarasi	Feature subset selection using genetic algorithm	Hospital database	70%	The language used to specify candidate solution must be robust. Population size, mutation rate, and crossover rate are all parameters of a Genetic Algorithm that must be carefully selected. A bad fitness function option can cause serious issues, such as being unable to solve a problem or, even worse, returning an incorrect answer to a problem the solution to the issue
[12]	Manpreet Singh	Structural equation modelling and Fuzzy cognitive mapping	Canadian community health survey (CCHS)dataset	74%	It doesn’t work well with large data and accuracy is low.
[13]	Kathleen j. Miao	Deep Neural Network	Cleveland heart disease database	83.67%	It is difficult to be adopted by people who are less experienced. It is difficult to comprehend performance based solely on

					understanding, and this necessitates the use of classifiers.
[14]	Jae Kwon Kim	Feature correlation Analysis	KNHANE S-VI dataset	81.163%	A correlational analysis can only be used when the variables are two measurable on scale. Cannot conclude cause and effect, strong association between variables can be misleading.
[15]	Sairabi H. Mujawa	Modified K-means and Naïve Bayes	Cleveland dataset	93% for presence of HD. 89% for absence of HD.	Naive Bayes assumes that all predictors are independent and It also have zero frequency problem.

1.6 Proposed System

The main idea behind the proposed system after reviewing the existing system was to create a heart disease prediction system based on the inputs as shown in Table.2. This system will analyze the classification algorithms namely Decision Tree, Random Forest, Logistic Regression, Support Vector Machine, K – Nearest Neighbor and Naive Bayes based on their Accuracy, Precision, Recall and f-measure scores and identify the best classification algorithm which can be used in the heart disease prediction.

S.No	Attribute Selection	Distinct Values of Attribute
1.	Age- represent the age of a person	Multiple values between 29 & 71
2.	Sex- describe the gender of person (0-Feamle, 1-Male)	0,1
3.	CP- represents the severity of chest pain patient is suffering.	0,1,2,3

4.	RestBP-It represents the patient’s BP.	Multiple values between 94& 200
5.	Chol-It shows the cholesterol level of the patient.	Multiple values between 126 & 564
6.	FBS-It represent the fasting blood sugar in the patient.	0,1
7.	Resting ECG-It shows the result of ECG	0,1,2
8.	Heartbeat- shows the max heart beat of patient	Multiple values from 71 to 202
9.	Exang- used to identify if there is an exercise induced angina. If yes=1 or else no=0	0,1
10.	OldPeak- describes patient’s depression level.	Multiple values between 0 to 6.2.
11.	Slope- describes patient condition during peak exercise. It is divided into three segments(Unsloping, Flat, Down sloping)	1,2,3.
12.	CA- Result of fluoroscopy.	0,1,2,3
13.	Thal- test required for patient suffering from pain in chest or difficulty in breathing. There are 4 kinds of values which represent Thallium test.	0,1,2,3
14.	Target-It is the final column of the dataset. It is class or label Colum. It represents the number of classes in dataset. This dataset has binary classification i.e. two classes (0,1).In class “0” represent there is less possibility of heart disease whereas “1” represent high chances of heart disease. The value “0” Or “1” depends on other 13 attribute.	0,1

Table.2. Features Extracted from the Dataset

2. LITERATURE SURVEY

In [1] Jayshril S. Sonawane et.al, proposed prediction of heart disease using multilayer perception neural network(2014). The accuracy by using this technique is 80%. Since the time complexity will be more due to usage of complex Boolean functions while using limited data sets for training, independently trained subnetworks scale quite well.

In [2] Ketut Agung Enrico et.al, a system was proposed by him for heart disease prediction using KNN algorithm with simplified parameters(2016). The accuracy of this algorithm is 81.85%. Using KNN , with increase of number of parameters the performance decreases and it considers 90% of data for training which is computationally expensive and does nothing during.

In [3] M.Akhil jabbar et.al, studies about heart disease prediction using Lazy Associative classification. The vast amount of space used to store the entire data collection. Since no abstraction is made during the training phases, especially noisy training data increases the case base unnecessarily.

In [4] Jaymin Patel et.al, proposed prediction of heart disease using data mining techniques(2015). The accuracy is 56.76%. The drawback of J48 is that the tree increases linearly with large data. LMT is slower and takes long time for implementation and accuracy is low.

In [5] Rifki wijaya et.al, studies about preliminary design of estimating heart disease by using machine learning ANN within one year(2013). The accuracy is 81.85%. To function, the neural network needs to be trained. Big neural networks necessitate a long processing time. Microprocessor design and history necessitate their emulation.

In [6] Carlos Ordonez et.al, proposed association rules to implement this prediction system. The accuracy is 70%. It uses too many parameters from patients record and produce irrelevant rules. So, this technique is computationally expensive and performance is low.

In [7] Jyoti soni et.al, did evaluation of Weighted Associative Classifier (WAC).The accuracy is 81.51%. All attributes are not equally important in predicting the class mark in the prediction model. As a result, various weights may be allocated to different attributes based on their ability to predict.

In [8] Idticeme sedjelmaci et.al, proposed detection of some heart diseases using fractal dimension and chaos theory(2013). The accuacy of this technique is 80%. Fractal analysis was created to analyse complex irregular objects. The drawbacks of applying Chaos Theory are primarily due to the input parameters chosen. The methods used to compute these parameters are determined by the underlying dynamics of the data as well as the type of analysis being performed, which is usually complex and not always precise.

In [9] D.R.Patil et.al, proposed prediction of heart disease using learning vector quantization algorithm. The accuracy of this algorithm is 85.55%. There is no limit to the number of prototypes that can be used per class; the only requirement is that each class have at least one prototype.

In [10] AH Chen et.al, presented a heart disease prediction system that can help doctors predict heart disease status using patient clinical data. The C language is used an artificial neural networks for classification and prediction of heart disease.The C and C# programming languages are used to develop system.The proposed method accuracy is 80%. m.

In [11] Anbarasi et.al, proposed prediction of heart disease with feature subset selection using genetic algorithm(2010). The accuracy is 70%. The language used to specify candidate solution must be robust.

In [12] Manpreet Singh et.al, proposed cardiovascular disease prediction system based on structural equation modelling (SEM) and Fuzzy cognitive map (FCM) (2016). The accuracy of SEM and FCM 74%. It doesn’t work well with large data and accuracy is low.

In [13] Kathleen j. Miao et.al, studies about coronary heart disease diagnosis using Deep Neutral Network(2015). The accuracy is 83.67%. It is difficult to be adopted by people who are less experienced. It is difficult to comprehend performance based solely on understanding, and this necessitates the use of classifiers.

In [14] Jae Kwon Kim et.al, proposed neural network based coronary heart disease risk prediction using feature correlation analysis(2017). The accuracy of this is 81.163%. A correlational analysis can only be used when the variables are two measurable on scale. It's difficult to tell which variables cause which effects, and a high correlation between variables can be misleading.

In [15] Sairabi H. Mujawar et.al, proposed a model for prediction of heart disease using modified K-means and Naive Bayes(2015). Naive Bayes assumes that all predictors are independent and it also have zero frequency problem.

3. METHODOLOGY

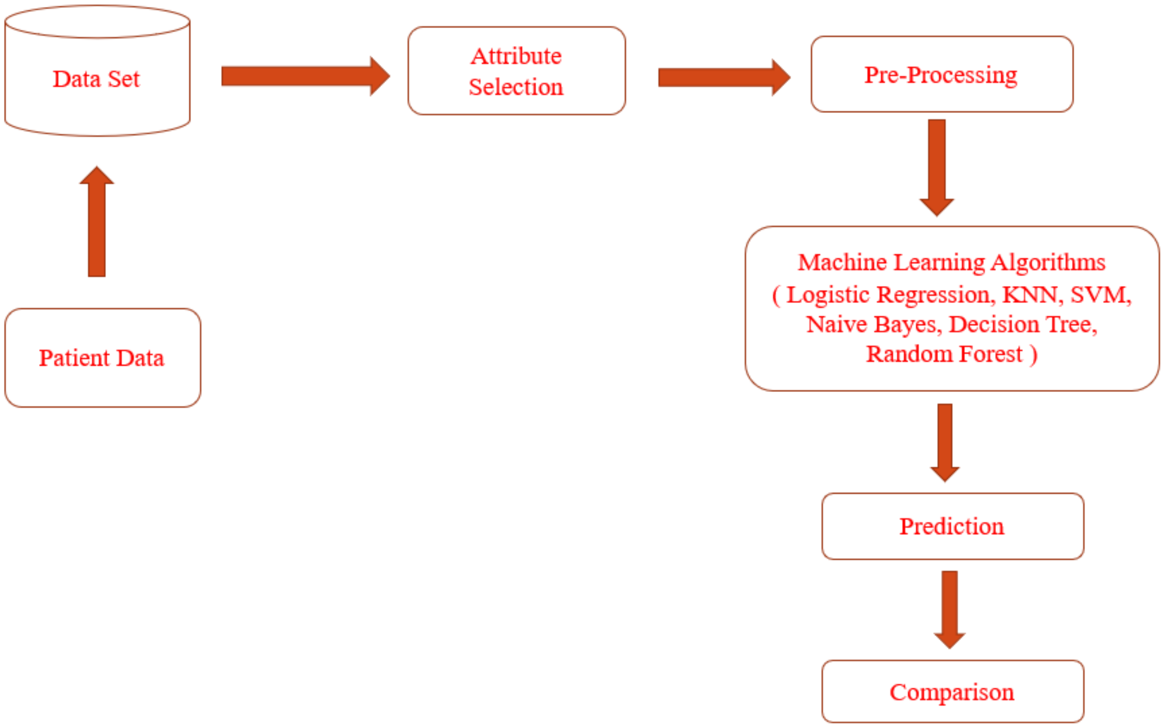
Data Set
Attribute Selection
Pre-Processing
Patient Data
Machine Learning Algorithms (Logistic Regression, KNN, SVM, Naive Bayes, Decision Tree, Random Forest)
Prediction
Comparison

Fig.2.Block Diagram for the Proposed System

3.1 Dataset for Heart Disease Prediction

Heart disease predicting model is trained by using Cleveland dataset which take age, sex, chest pain, resting blood pressure, rest ECG, maximum heart rate, exercise include angina, ST depression, ST slope, number of major blood vessels, types of thalassemia as inputs and produce output stating whether patient is suffering from heart disease or not. Random forest classifier is used to classify the data and for prediction. Feature selection algorithms are also used which helps to improve accuracy of the model. Basically random forest classifier is used for both classification and regression but we use this for classification purpose and can overcome missing values. Accuracy obtained for random forest classifier is 93.44%. Univariate data analysis is done in that process categorical variables are dropped.

3.2 Attribute Selection

An attribute selection measure is a heuristic for choosing the splitting test that “best” separates a given data partition, D, of class-labeled training tuples into single classes. If it can split D into smaller partitions as per the results of the splitting criterion, ideally every partition can be pure (i.e., some tuples that fall into a given partition can belong to the same class). Conceptually, the “best” splitting criterion is the most approximately results in such a method. Attribute selection measures are called a splitting rules because they decides how the tuples at a given node are to be divided. The attribute selection measure supports a ranking for every attribute defining the given training tuples. The attribute having the best method for the measure is selected as the splitting attribute for the given tuples.

3.3 Pre-Processing

Data preprocessing is a process of preparing the raw data and making it suitable for a machine learning model. It is the first and crucial step while creating a machine learning model. When creating a machine learning project, it is not always a case that we come across the clean and formatted data. And while doing any operation with data, it is mandatory to clean it and put in a formatted way. So for this, we use data preprocessing task.

3.4 Machine Learning Algorithms

Heart disease predicting model is trained by using the Cleveland dataset which takes age, gender, chest pain, resting blood pressure, rest ECG, maximum heart rate, exercise including angina, ST depression, ST slope, number of major blood vessels, types of thalassemia as inputs and produce output stating whether the patient is suffering from heart disease or not. If only a single algorithm is used it cannot Pre-Process data and even it can’t get good accuracy. So it’s better to have a combination of algorithms like “Logistic Regression”, “KNN”, “SVM”, “Naive Bayes”, “Decision Tree”, and "Random Forest“. A lot of work has been carried out to predict heart disease using the Machine Learning dataset. Different levels of accuracy have been attained using various data mining techniques. This Project “Heart Disease Prediction using Machine Learning algorithms” is implemented using python completely.

3.4.1 Random Forest classifier

The random forest is a classification algorithm that uses several decision trees to classify data. When constructing each individual tree, it employs bagging and feature randomness in order to establish an uncorrelated forest of trees whose committee prediction is more reliable than that of any single tree. It also aims to reduce the difficulties associated with high blood pressure. By averaging, you can establish a natural equilibrium between high variation and high bias between two extremities. This approach can be implemented in R and Python using robust libraries.

3.4.2 Logistic Regression Classifier

Logistic regression is supervised classification technique. It's used to forecast a categorical dependent variable using a variety of independent variables. The output of a categorical dependent variable is predicted using logistic regression. As a result, the outcome must be a discrete or categorical attribute. It’s easier to put into practice, interpret, and train with. Logistic regression should not be employed if the number of observations is less than the number of features in input data set; otherwise, overfitting may occur.

3.4.3 KNN Classifier

KNN is a simple classification model, which act as non-parametric algorithm which doesn’t make any pre assumptions about data distribution during analysis. KNN considers 90% of data for training which is computationally expensive and does nothing during training phase. K-Nearest Neighbour is a classification method that uses an imaginary border to classify data. When fresh data points are received, the algorithm will attempt to anticipate them as closely as possible to the boundary line. It is imported from scikit-learn package.

3.4.4 SVM Classifier

A support vector machine (SVM) is a supervised machine learning model that solves two-group classification problems with classification algorithms. SVM models will categorize new text after being fed sets of named training data for each group. In this, maximizing the hyperplane margin will help to overcome the problems of misclassification. Some of well known support vector machine implementations are Scikit-learn, MATLAB, and LIBSVM.

3.4.5 Decision Tree Classifier

Decision trees are similar to tree structures which are mainly used for decision making in machine learning. For classification and regression, Decision Trees (DTs) are a non-parametric supervised learning process. The aim is to learn basic decision rules from data features to build a model that predicts the value of a target variable. In decision tree, each internal node represents an attribute query, each branch a test result, and each a label class.

3.4.6 Naive Bayes Classifier

Naive Bayes methods are a class of supervised learning algorithms based on Bayes' theorem. Given the class variable, a naive Bayes classifier assumes that the existence (or absence) of one aspect of a class is unrelated to the existence (or absence) of any other feature. It's "naive" in the sense that it makes assumptions that may or may not be right. It assumes that each feature being classified is independent on other features. high blood pressure. By averaging, you can establish a natural equilibrium between high variation and high bias between two extremities. This approach can be implemented in R and Python using robust libraries.

3.5 Prediction

Prediction refers to the output of an algorithm after it has been trained on a historical dataset and applied to new data when predicting the likelihood of a particular outcome.

3.6 Comparision

The key to a fair comparison of machine learning algorithms is ensuring that each algorithm is evaluated in the same way on the same data you can achieve this by forcing each algorithm to be evaluated on a consistent test harness. In this system, these 6 different algorithms are compared: Logistic Regression, Decision Tree, K-Nearest Neighbors, Random Forest, Naive Bayes, Support Vector Machine.

3.7 Implementation

The dataset which is used for training data and classified by using datamining classifier algorithms. From the values of the dataset, it is reduced input values to 14 main data inputs taken from user and univariate, bivariate analysis is done on the data. The System split the dataset randomly into training and testing datasets then we preprocess them through this we get analytical results and based on that heart disease heart disease prediction will be done by the machine learning model.

4. RESULTS & DISCUSSION

Different techniques are used for classification purpose and minimum number of required attributes that is 14 required attributes are taken and Random forest, KNN, Support Vector Machine, Regression classifier, Naïve Bayes, and Decision Tree classification algorithms are used which gave very good results and future scope is extend the application like by using same attributes or by adding some more attributes we can predict different other diseases also like kidney related lungs related diseases. This application can be made as common platform for predicting all kind of diseases.

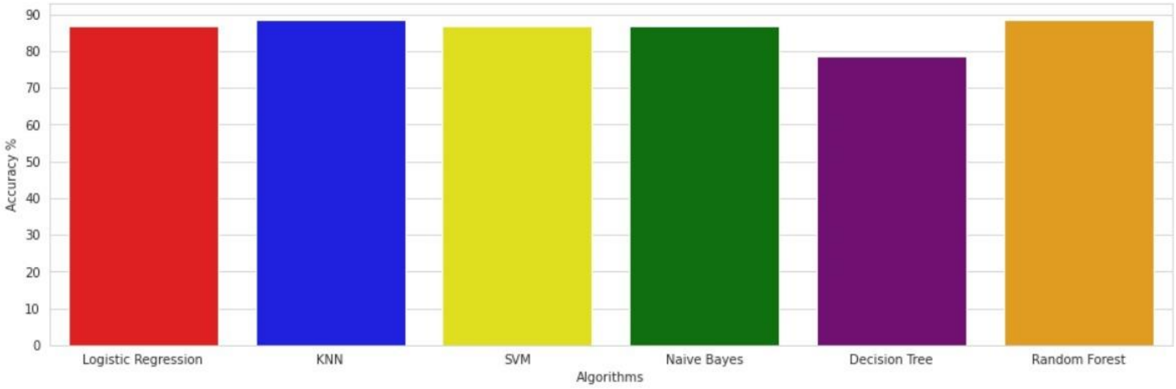

Fig.3. Bar Graph for Heart Disease Accuracy by various Algorithms

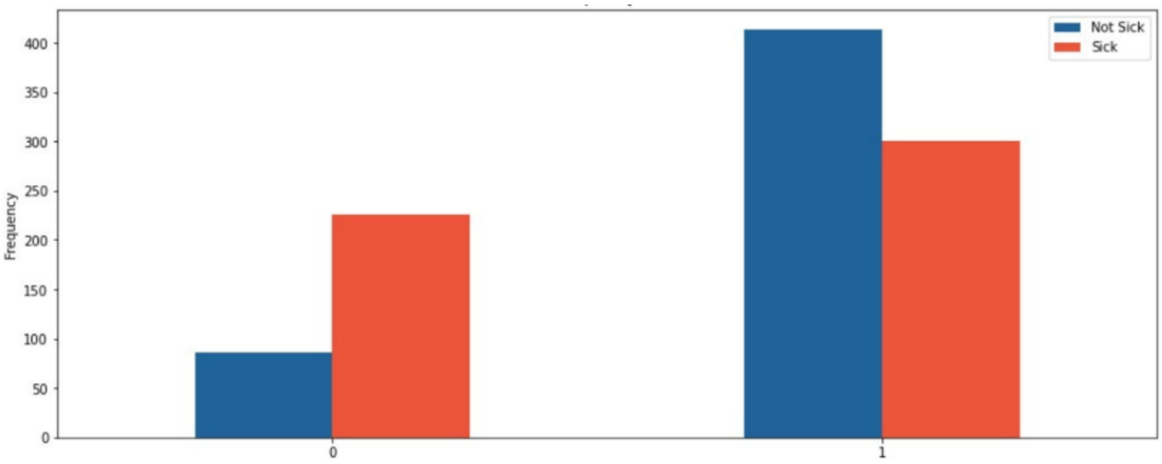

Fig.4. Heart Disease Prediction based on Gender

5. CONCLUSION

So, Finally the system concludes that, Heart Disease prediction using machine learning is extremely much useful in everyone’s day to day life and it's mainly more important for the healthcare sector, because they're the one that daily uses these systems to predict the diseases of the patients supported their general information and there symptoms that they're been through. Now a day’s health industry plays important role in curing the diseases of the patients so this is often also some quite help for the health industry to inform the user and also it's useful for the user just in case he/she doesn’t want to travel to the hospital or the other clinics, so just by entering the symptoms and every one other useful information within the form user can get to know the disease he/she is affected by and therefore the health industry also can get enjoy this technique by just asking the symptoms from the user and entering within the system and in only few seconds they will tell the precise and up to some extent the accurate diseases. If health industry adopts this project then the work of the doctors are often reduced and that they can easily predict the disease of the patient. The Disease prediction is to supply prediction for the varied and usually occurring diseases that when unchecked and sometimes ignored can turns into fatal disease and cause lot of problem to the patient. The System can be updated in future by adding more attributes to the dataset and more interactive to the users and can also be done as a mobile application and can also modify the system by connecting it to the hospital’s database.

6. REFERENCES

[1] Jayshril S. Sonawane, D.R Patil, 2014, " Prediction Of Heart Disease Using Multilayer Perceptron Neural Network ", *IEEE International Conference on Information Communication and Embedded Systems (ICICES2014).*

[2] I Ketut Agung Enriko, Muhammad Suryanegara,Dinda Agnes Gunawan, " Heart disease prediction system using k-Nearest neighbor algorithm with simplified patient's health parameters", 2016.

[3] M. Akhil Jabbar; B. L Deekshatulu; Priti Chandra, " Heart disease prediction using lazy associative classification", : 2013, *IEEE International Mutli-Conference on Automation, Computing, Communication, Control and Compressed Sensing (iMac4s).*

[4] Jaymin Patel, Prof. Tejal Upadhyay, and Dr. Samir Patel, Sep 2015-Mar 2016, "Heart Disease Prediction using Machine Learning and Data Mining Technique", *Vol. 7, No.1, pp. 129-137.*

[5] Rifki Wijaya, ArySetijadiPrihatmanto, Kuspriyanto, " Preliminary design of estimation heart disease by using machine learning ANN within one year", 2013, *IEEE Joint International Conference on Rural Information & Communication Technology and Electric-Vehicle Technology (rICT&ICeV-T).*

[6] Carlos Ordonez,2006, "Association Rule Discovery with the Train and Test Approach for Heart Disease Prediction", *IEEE Transactions on Information Technology in Biomedicine (TITB), pp. 334-343, vol. 10, no. 2.*

[7] Jyoti Soni, Uzma Ansari, Dipesh Sharma, and SunitaSoni,June 2011, "Intelligent and Effective Heart Disease Prediction System using Weighted Associative Classifiers", *International Journal on Computer Science and Engineering (IJCSE), Vol. 3, No. 6, pp. 2385-2392.*

[8] IbticemeSedielmaci; F. BereksiReguig, " Detection of some heart diseases using fractal dimension and chaos theory", 2013, *IEEE 8th International Workshop on Systems, Signal Processing and their Applications (WoSSPA).*

[9] Jayshril S. Sonawane; D. R. Patil, " Prediction of Heart Disease Using Learning Vector Quantization Algorithm ", 2014, *IEEE Conference on IT in Business, Industry and Government (CSIBIG).*

[10] AH Chen, SY Huang, PS Hong, CH Cheng, and EJ Lin,2011, "HDPS: Heart Disease Prediction System", Computing in Cardiology*, ISSN: 0276-6574, pp.557-560.*

[11] Anbarasi Masilamani, ANUPRIYA, N Ch Sriman Narayana Iyenger, "Enhanced Prediction of Heart Disease with Feature Subset Selection using Genetic Algorithm", October 2010, *International Journal of Engineering Science and Technology 2(10).*

[12] Manpreet Singh, Levi Monteiro Martins, Patrick Joanis, and Vijay K. Mago,2016, "Building a Cardiovascular Disease Predictive Model using Structural Equation Model & Fuzzy Cognitive Map", *IEEE International Conference on Fuzzy Systems (FUZZ), pp. 1377-1382.*

[13] Kathleen H. Miao, Julia H. Miao, "Coronary Heart Disease Diagnosis using Deep Neural Networks ",*(IJACSA)International Journal of Advanced Computer Science and Applications, Vol. 9, No. 10, 2018* .

[14] Jae Kwon Kim and Sanggil, "Neural Network-Based Coronary Heart Disease Risk Prediction Using Feature Correlation Analysis", 2017.

[15] Sairabi H. Mujawar, and P. R. Devale, October 2015,"Prediction of Heart Disease using Modified k-means and by using Naive Bayes", *International Journal of Innovative Research in Computer and Communication Engineering(An ISO 3297: 2007 Certified Organization) Vol. 3, Issue 10, pp. 10265-10273.*

APPENDIX A : SOURCE CODE

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
import os

df = pd.read_csv("heart.csv")
df.head()

df.target.value_counts()

sns.countplot(x="target", data=df, palette="bwr")
plt.show()

countTdkSakit = len(df[df.target == 0])
countSakit = len(df[df.target == 1])
print("Percentage of patients who are not sick: {:.2f}%".format((co
untTdkSakit / (len(df.target))*100)))
print("Percentage of patients who are sick: {:.2f}%".format((countS
akit / (len(df.target))*100)))

sns.countplot(x='sex', data=df, palette="mako_r")
plt.xlabel("Gender (0 = Female, 1= Male)")
plt.show()

countWanita = len(df[df.sex == 0])
countPria = len(df[df.sex == 1])
print("Presentage of Female Patients: {:.2f}%".format((countWanita
/ (len(df.sex))*100)))
print("Presentage of Male Patients: {:.2f}%".format((countPria / (l
en(df.sex))*100)))

df.groupby('target').mean()

pd.crosstab(df.age,df.target).plot(kind="bar",figsize=(20,6))
plt.title('Heart Disease Frequency based on Age')
plt.xlabel('Age')
plt.ylabel('Frequency')
plt.savefig('heartDiseaseAndAges.png')
plt.show()
```

```
pd.crosstab(df.sex,df.target).plot(kind="bar",figsize=(15,6),color=
['#20639B','#ED553B' ])
plt.title('Heart Disease Frequency based on Gender')
plt.xlabel('Sex (0 = Female, 1 = Male)')
plt.xticks(rotation=0)
plt.legend(["Not Sick", "Sick"])
plt.ylabel('Frequency')
plt.show()

plt.scatter(x=df.age[df.target==1], y=df.thalach[(df.target==1)], c
="red")
plt.scatter(x=df.age[df.target==0], y=df.thalach[(df.target==0)], c
="green")
plt.legend(["Sick", "Not Sick"])
plt.xlabel("Age")
plt.ylabel("Heart Rate Max")
plt.show()

pd.crosstab(df.slope,df.target).plot(kind="bar",figsize=(15,6),colo
r=['#6C5B7B','#F8B195' ])
plt.title('Heart Disease Frequency based on Slope')
plt.xlabel('The Slope of The Peak Exercise ST Segment ')
plt.xticks(rotation = 0)
plt.ylabel('Frequency')
plt.show()

pd.crosstab(df.fbs,df.target).plot(kind="bar",figsize=(15,6),color=
['#009999','#00FF00' ])
plt.title('Heart Disease Frequency According To FBS')
plt.xlabel('FBS > 120 mg/dl (1 = true; 0 = false)')
plt.xticks(rotation = 0)
plt.legend(["Not Sick", "Sick"])
plt.ylabel('Frequency Sick/Not Sick')
plt.show()

pd.crosstab(df.cp,df.target).plot(kind="bar",figsize=(15,6),color=[
'#0000CC','#FFFF99' ])
plt.title('Heart Disease Frequency According To Chest Pain Type')
plt.xlabel('Chest Pain Type')
plt.xticks(rotation = 0)
plt.ylabel('Frequency Sick/Not Sick')
plt.show()

a = pd.get_dummies(df['cp'], prefix = "cp")
b = pd.get_dummies(df['thal'], prefix = "thal")
c = pd.get_dummies(df['slope'], prefix = "slope")
```

```
frames = [df, a, b, c]
df = pd.concat(frames, axis = 1)
df.head()

df = df.drop(columns = ['cp', 'thal', 'slope'])
df.head()

y = df.target.values
x_data = df.drop(['target'], axis = 1)

#Normalization
x = (x_data - np.min(x_data)) / (np.max(x_data) - np.min(x_data)).v
alues
x_train, x_test, y_train, y_test = train_test_split(x,y,test_size =
 0.2,random_state=0)

x_train = x_train.T
y_train = y_train.T
x_test = x_test.T
y_test = y_test.T

def initialize(dimension):
    weight = np.full((dimension,1),0.01)
    bias = 0.0
    return weight,bias

#Sigmoid Function
def sigmoid(z):

    y_head = 1/(1+ np.exp(-z))
    return y_head

def forwardBackward(weight,bias,x_train,y_train):
    # Forward
    y_head = sigmoid(np.dot(weight.T,x_train) + bias)
    loss = -(y_train*np.log(y_head) + (1-y_train)*np.log(1-y_head))
    cost = np.sum(loss) / x_train.shape[1]
    # Backward
    derivative_weight = np.dot(x_train,((y_head-
y_train).T))/x_train.shape[1]
    derivative_bias = np.sum(y_head-y_train)/x_train.shape[1]
    gradients = {"Derivative Weight" : derivative_weight, "Derivati
ve Bias" : derivative_bias}
    return cost,gradients


def update(weight,bias,x_train,y_train,learningRate,iteration) :
    costList = []
    index = []
    for i in range(iteration):
        cost,gradients = forwardBackward(weight,bias,x_train,y_train)
        weight = weight - learningRate * gradients["Derivative Weight"]
        bias = bias - learningRate * gradients["Derivative Bias"]
        costList.append(cost)
        index.append(i)
    parameters = {"weight": weight,"bias": bias}
    print("iteration:",iteration)
    print("cost:",cost)
    plt.plot(index,costList)
    plt.xlabel("Number of Iteration")
    plt.ylabel("Cost")
    plt.show()
    return parameters, gradients

def predict(weight,bias,x_test):
    z = np.dot(weight.T,x_test) + bias
    y_head = sigmoid(z)
    y_prediction = np.zeros((1,x_test.shape[1]))
    for i in range(y_head.shape[1]):
        if y_head[0,i] <= 0.5:
            y_prediction[0,i] = 0
        else:
            y_prediction[0,i] = 1
    return y_prediction


def logistic_regression(x_train,y_train,x_test,y_test,learningRate,iteration):
    dimension = x_train.shape[0]
    weight,bias = initialize(dimension)
    parameters, gradients = update(weight,bias,x_train,y_train,learningRate,iteration)
    y_prediction = predict(parameters["weight"],parameters["bias"],x_test)
    print("Manuel Test Accuracy: {:.2f}%".format((100 - np.mean(np.abs(y_prediction - y_test))*100)/100*100))

logistic_regression(x_train,y_train,x_test,y_test,1,100)

lr = LogisticRegression()
lr.fit(x_train.T,y_train.T)
```

```
print("Test Accuracy {:.2f}%".format(lr.score(x_test.T,y_test.T)*10
0))

from sklearn.neighbors import KNeighborsClassifier
knn = KNeighborsClassifier(n_neighbors = 2)
knn.fit(x_train.T, y_train.T)
prediction = knn.predict(x_test.T)

print("{} NN Score: {:.2f}%".format(2, knn.score(x_test.T, y_test.T
)*100))

scoreList = []
for i in range(1,20):
    knn2 = KNeighborsClassifier(n_neighbors = i)  # n_neighbors mea
ns k
    knn2.fit(x_train.T, y_train.T)
    scoreList.append(knn2.score(x_test.T, y_test.T))

plt.plot(range(1,20), scoreList)
plt.xticks(np.arange(1,20,1))
plt.xlabel("K value")
plt.ylabel("Score")
plt.show()
print("KNN Score Max {:.2f}%".format((max(scoreList))*100))

from sklearn.svm import SVC

svm = SVC(random_state = 1)
svm.fit(x_train.T, y_train.T)

print("SVM ALgorithm Test Accuracy: {:.2f}%".format(svm.score(x_tes
t.T,y_test.T)*100))

from sklearn.naive_bayes import GaussianNB
nb = GaussianNB()
nb.fit(x_train.T, y_train.T)
print("Accuracy of Naive Bayes: {:.2f}%".format(nb.score(x_test.T,y
_test.T)*100))

from sklearn.tree import DecisionTreeClassifier
dtc = DecisionTreeClassifier()
dtc.fit(x_train.T, y_train.T)
print("Decision Tree Test Accuracy {:.2f}%".format(dtc.score(x_test
.T, y_test.T)*100))

methods = ["Logistic Regression", "KNN", "SVM", "Naive Bayes", "Dec
ision Tree", "Random Forest"]
```

```
accuracy = [86.89, 88.52, 86.89, 86.89, 78.69, 88.52]
colors = ["red", "blue", "yellow", "green","purple","orange"]

sns.set_style("whitegrid")
plt.figure(figsize=(16,5))
plt.yticks(np.arange(0,100,10))
plt.ylabel("Accuracy %")
plt.xlabel("Algorithms")
sns.barplot(x=methods, y=accuracy, palette=colors)
plt.show()

y_head_lr = lr.predict(x_test.T)
knn3 = KNeighborsClassifier(n_neighbors = 3)
knn3.fit(x_train.T, y_train.T)
y_head_knn = knn3.predict(x_test.T)
y_head_svm = svm.predict(x_test.T)
y_head_nb = nb.predict(x_test.T)
y_head_dtc = dtc.predict(x_test.T)

from sklearn.metrics import confusion_matrix

cm_lr = confusion_matrix(y_test,y_head_lr)
cm_knn = confusion_matrix(y_test,y_head_knn)
cm_svm = confusion_matrix(y_test,y_head_svm)
cm_nb = confusion_matrix(y_test,y_head_nb)
cm_dtc = confusion_matrix(y_test,y_head_dtc)
```

APPENDIX B : SCREENSHOTS

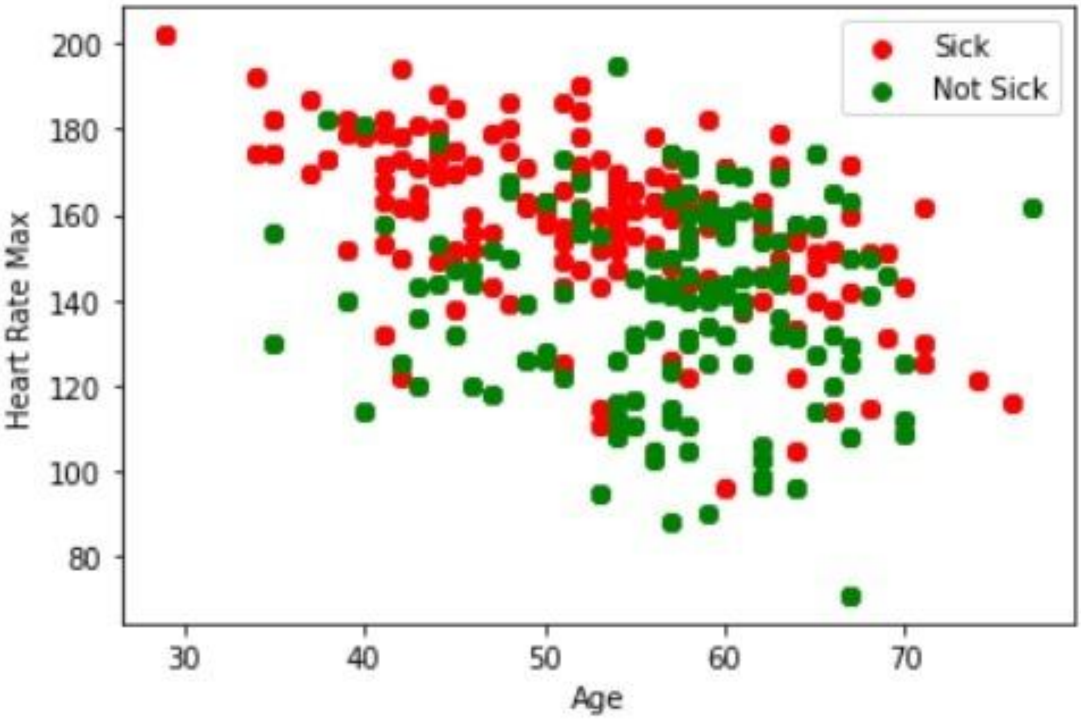

Fig.5. Plot Graph of the Output

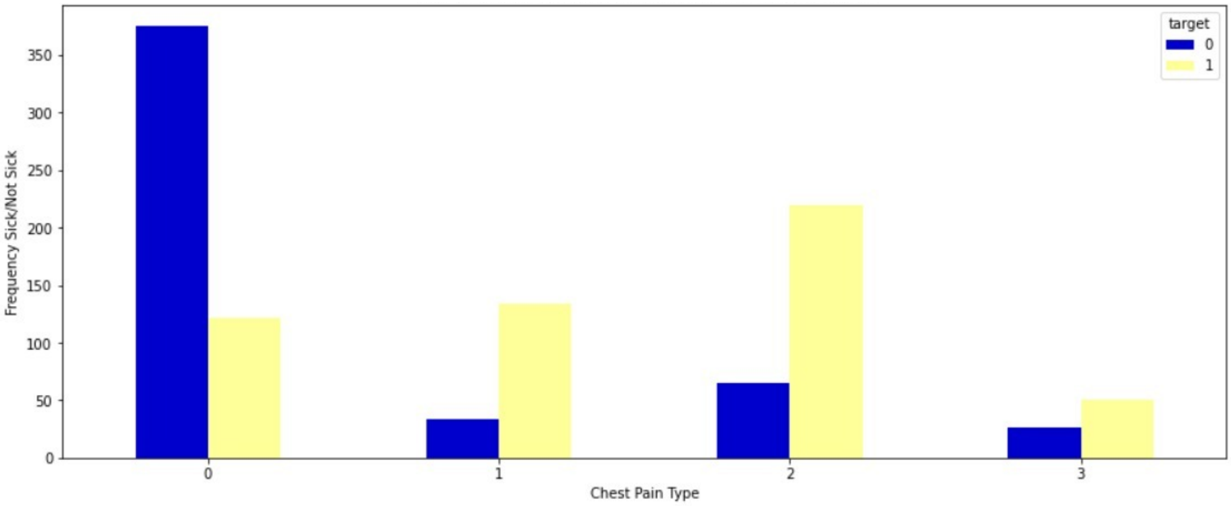

Fig.6. Heart Disease Frequency According to chest pain

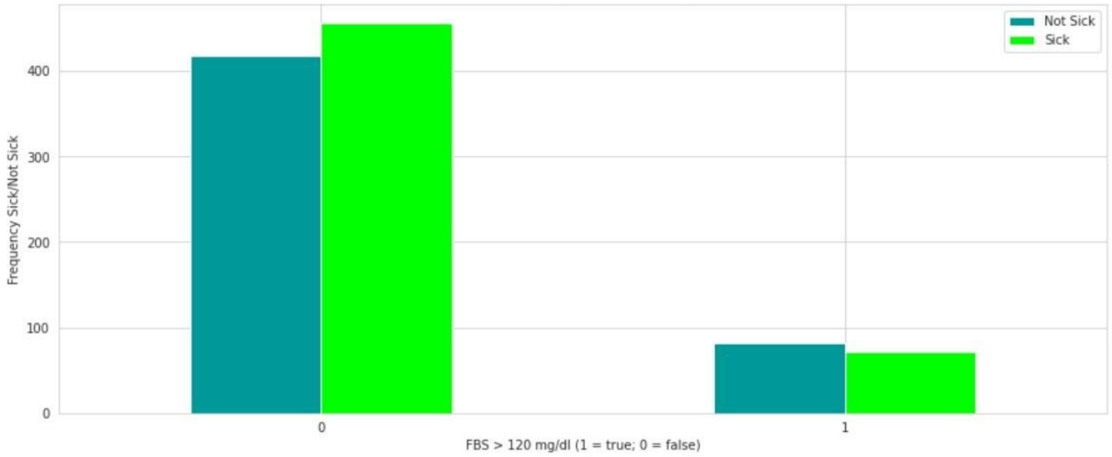

Fig.7. Heart Disease Frequency According to FBS

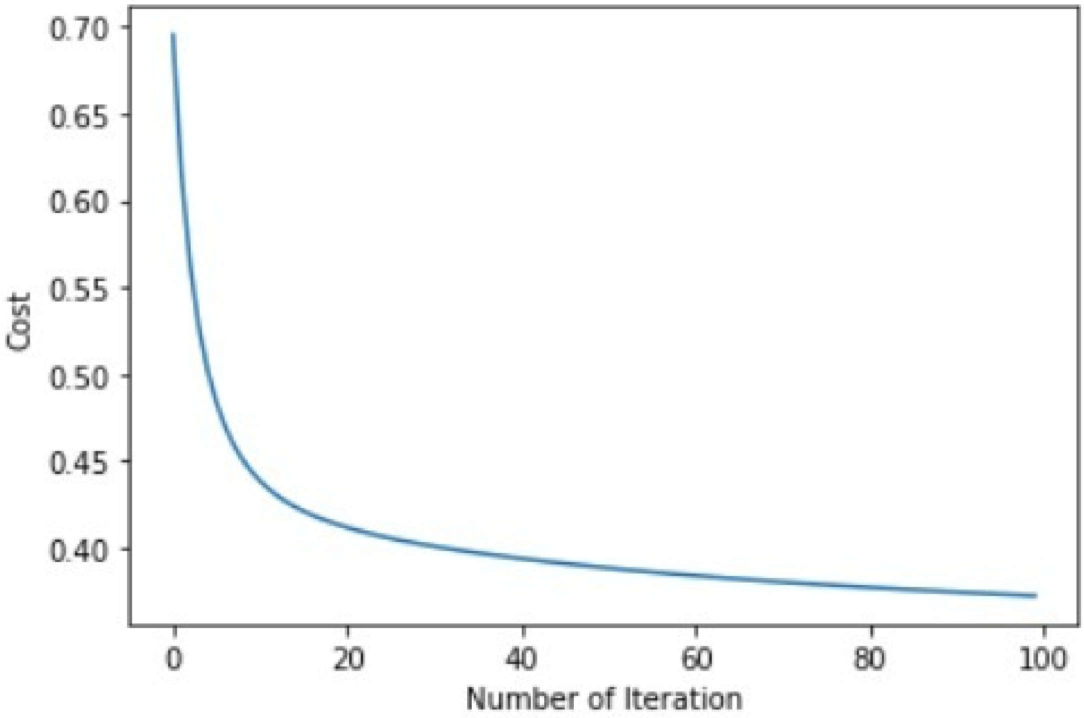

Fig.8. Accuracy Rate Testing

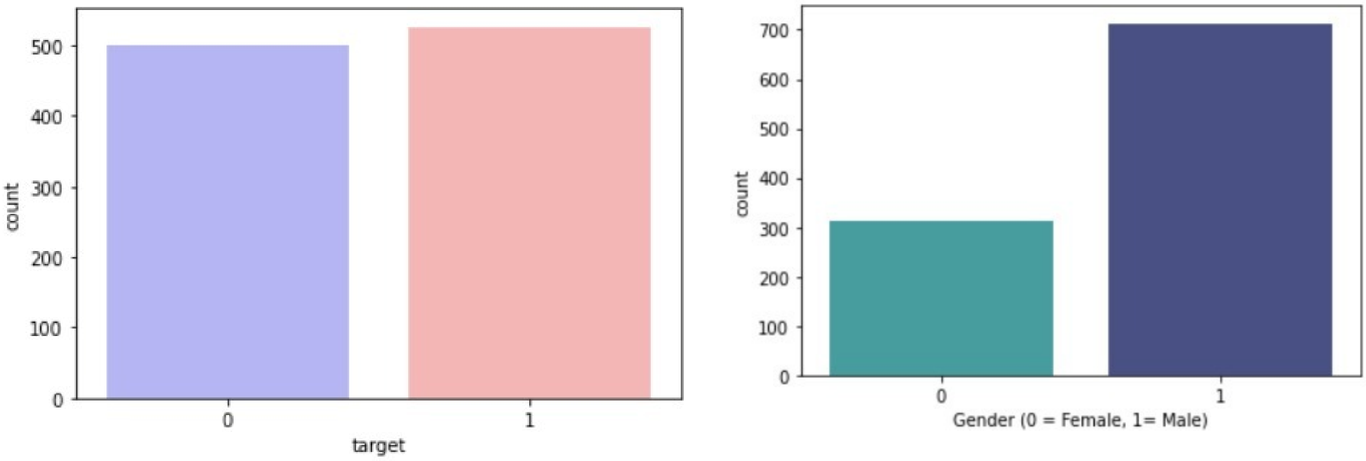

Fig.9 No.of Patients Affected/ Not Affected

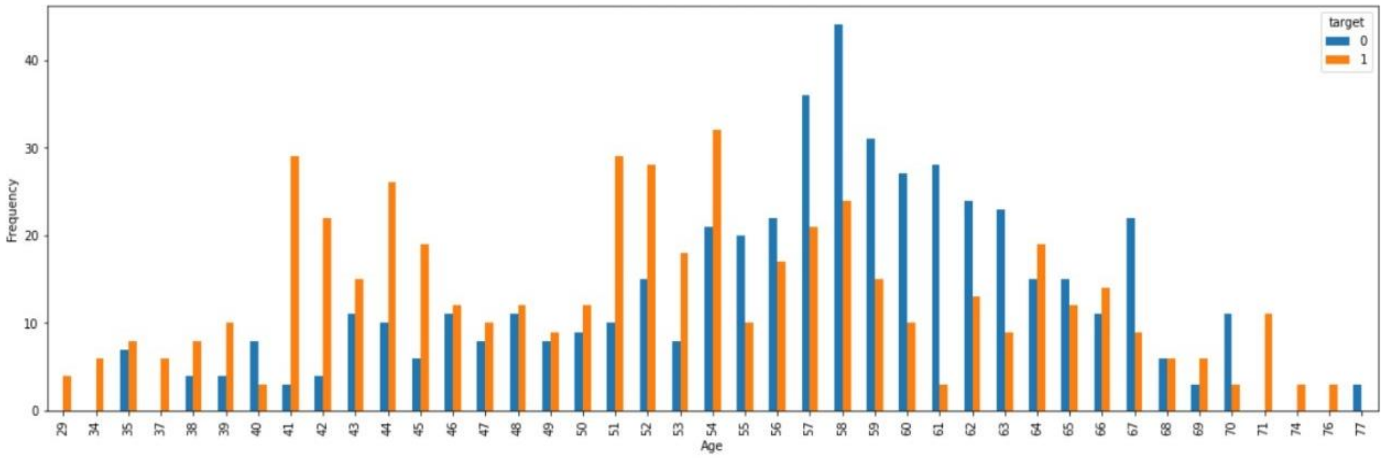

Fig.10 Heart Disease Frequency based on Age

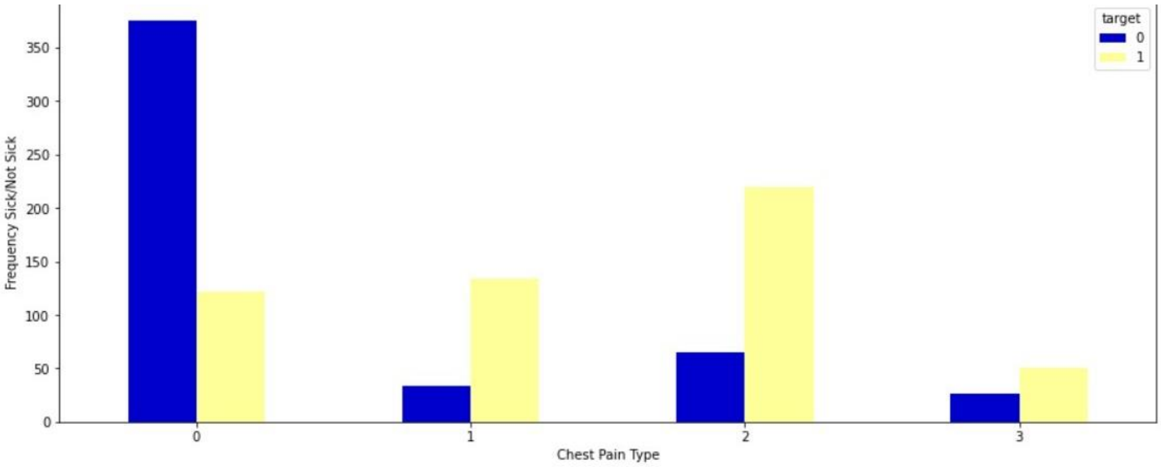

Fig.11 Heart Disease Frequency based on Chest Pain Type

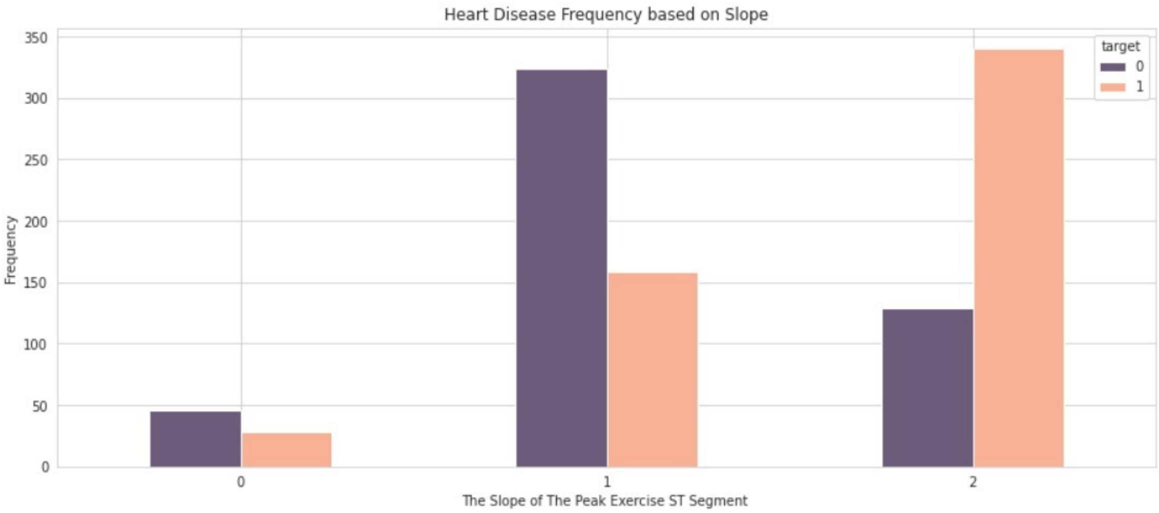

Fig.12 Heart Disease Frequency based on ST slope