Human Heart Disease Exploratory Data Analysis and Prediction Using Machine Learning Algorithms

Mamshad Pathan

Table of Contents

CHAPTER 1

INTRODUCTION

The most vital and essential part (organ) of human body is Heart. There are many diseases that are linked to heart so The analysis of prediction of heart must be accurate. To resolve this, virtual study about this field obligatory. Normally these diseases predicted at end stage and this is the main reason of death of heart patient due to deficiency of correctness because of this there is requisite to identify about proficient algorithms for diseases prediction [1]. Cardiovascular diseases are a group of disorders involving the heart and blood vessels and one of the leading causes of death globally, according to the American Heart Association. In 2019, cardiovascular diseases took the lives of nearly 18 million people, accounting for 32% of deaths worldwide (World Health Organization, 2021). 85% of these deaths were due to heart attacks and strokes, with 38% among people under the age of 70. Early detection is critical in the treatment and management of cardiovascular diseases; wherein machine learning can be a powerful tool in detecting a potential heart disease diagnosis.

In this project, we have developed a Machine Learning model to predict the presence of Heart disease and identify some of the most important predictors based on Machine Learning Algorithms. As ML description, it absorbs from the ordinary phenomenon. By using python libraries with the algorithms of machine learning, this prediction has to be done [2]. In this detection, elements of biological are used such as chest pain (cp), age, sex, trestbps, fbs, restecg, thalach, oldpeak, blood pressure(bp), cholesterol(chol). By using of these elements nine algorithms of ML such as Logistic Regression, K-Nearest Neighbor (KNN), Support Vector Machine (SVM), Gaussian Naive Bayes, Decision Tree, Random Forest, Extra Tree Classifier, Gradient Boosting, and AdaBoost is applied for prediction of analysis and conclude that which method to combine the results of those learning algorithms and compare the result of the combination solution [3].

1.1 Problem Statement

Healing illness is the responsibility of a doctor. However, it will be very helpful if patients can know a little bit about their health beforehand. Therefore, in this project, we plan to study and apply several existing machine learning algorithms on the heart disease prediction puzzle to achieve the goal that by entering the body information, people can get a straightforward description of their current health situation. This contributes to the prevention of a severe disease which may come in silence since people can use it to do daily check.

1.2 Motivation

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 is quite exciting to know, which parameters affect our heart and how! We know some of these like, cholesterol, age, blood pressure, etc. but we want to use data science skills to know more about this, and so we decided to work on this project.

1.3 Contribution

We have split our project work into Exploratory Data Analysis and Prediction using Machine Learning Algorithms. Moreover, we have tried to increase the accuracy level of all Machine Learning Algorithm to get the accurate predicted result. Hence, we have done performance analysis in between the algorithms by using different confusion matrix.

CHAPTER 2

BACKGROUND STUDY

The heart is one of the most important parts as it pumps blood around your body, delivering oxygen and nutrients to your cells and removing waste products. 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. A heart attack occurs when an artery supplying your heart with blood and oxygen becomes blocked. Fatty deposits build up over time, forming plaques in your heart's arteries. If a plaque ruptures, a blood clot can form and block your arteries, causing a heart attack. In this regard, machine learning algorithms are efficient and reliable sources to detect and categorize persons suffering from heart disease and those who are healthy.

2.1 Project Overview

In this project, we propose machine learning models to predict the possibility of having heart disease using various algorithms. The model is executed using seven algorithms Logistic Regression, K-Nearest Neighbor (KNN), Support Vector Machine (SVM), Gaussian Naive Bayes, Decision Tree, Random Forest, and AdaBoost. We used Kaggle dataset [4] for training and testing the model. The dataset is preprocessed followed by feature selection to select most prominent features. The resultant dataset is then used for training the framework. The results are combined and show that Random forest gives maximum accuracy.

The first step is to use different data visualization methods to represent the text-based data into a visual format for identifying undetected trends. Next second step is to use feature selection to reduce redundant and trivial data which improves the prediction rate significantly. The third step is to use classification techniques to train the model and predict on the testing dataset. The fourth step is to propose a method for boosting the prediction rate for technique.

Objective:

- To demonstrate algorithms like Logistic Regression, K-Nearest Neighbor (KNN), Support Vector Machine (SVM), Gaussian Naive Bayes, Decision Tree, Random Forest, and AdaBoost.
- To demonstrate prediction boosting for each machine learning technique.
- To evaluate the prediction based on performance in the final statement.

2.2 Data Collection and Feature Analysis

We have used Kaggle dataset [4] which has 14 variables including 9 categorical and 5 continuous variables. According to the study, we have identified and predict human heart disease using a variety of machine learning algorithms and used the heart disease dataset to evaluate its performance using different metrics for evaluation, such as sensitivity, specificity, F-measure, and classification accuracy. For this purpose, we have used nine classifiers of machine learning to the final dataset before and after the hyper parameter tuning of the machine learning classifiers, such as AB, LR, ETC, KNN, SVM, GNB, DT, RF, and GB. Furthermore, we check their accuracy on the standard heart disease dataset by performing certain preprocessing, standardization of dataset. The goal is to identify the parameters that influence the heart attack and build a ML model for the prediction of heart attack.

First step for predication system is data collection, data collect from net [4] after that used data wrangling for data cleaning and then determining about the training and testing dataset. In this project we have used 80% training dataset and 20% dataset used as testing dataset. After Cleaning, in this data set there are 14 columns and 1025 rows as shown in Fig. 1

age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
52	1	0	125	212	0	1	168	0	1	2	2	3	0
53	1	0	140	203	1	0	155	1	3.1	0	0	3	0
70	1	0	145	174	0	1	125	1	2.6	0	0	3	0
61	1	0	148	203	0	1	161	0	0	2	1	3	0
62	0	0	138	294	1	1	106	0	1.9	1	3	2	0
58	0	0	100	248	0	0	122	0	1	1	0	2	1
58	1	0	114	318	0	2	140	0	4.4	0	3	1	0
55	1	0	160	289	0	0	145	1	0.8	1	1	3	0
46	1	0	120	249	0	0	144	0	0.8	2	0	3	0
54	1	0	122	286	0	0	116	1	3.2	1	2	2	0
71	0	0	112	149	0	1	125	0	1.6	1	0	2	1
43	0	0	132	341	1	0	136	1	3	1	0	3	0
34	0	1	118	210	0	1	192	0	0.7	2	0	2	1
51	1	0	140	298	0	1	122	1	4.2	1	3	3	0
52	1	0	128	204	1	1	156	1	1	1	0	0	0
34	0	1	118	210	0	1	192	0	0.7	2	0	2	1
51	0	2	140	308	0	0	142	0	1.5	2	1	2	1
54	1	0	124	266	0	0	109	1	2.2	1	1	3	0
50	0	1	120	244	0	1	162	0	1.1	2	0	2	1
58	1	2	140	211	1	0	165	0	0	2	0	2	1
60	1	2	140	185	0	0	155	0	3	1	0	2	0

Figure 1: Dataset Info

2.3 Classification Model

2.3.1 Logistic Regression

Logistic regression essentially uses a logistic function defined below to model a binary output variable. The primary difference between linear regression and logistic regression is that logistic regression's range is bounded between 0 and 1. This is due to applying a nonlinear log transformation to the odds ratio.

$$\text{Logistic function} = \frac{1}{1+e^{-x}}$$

In order to map predicted values to probabilities, we use the sigmoid function. The function maps any real value into another value between 0 and 1. In machine learning, we use sigmoid to map predictions to probabilities. As figure 2, the predicted class label is depending on final threshold function.

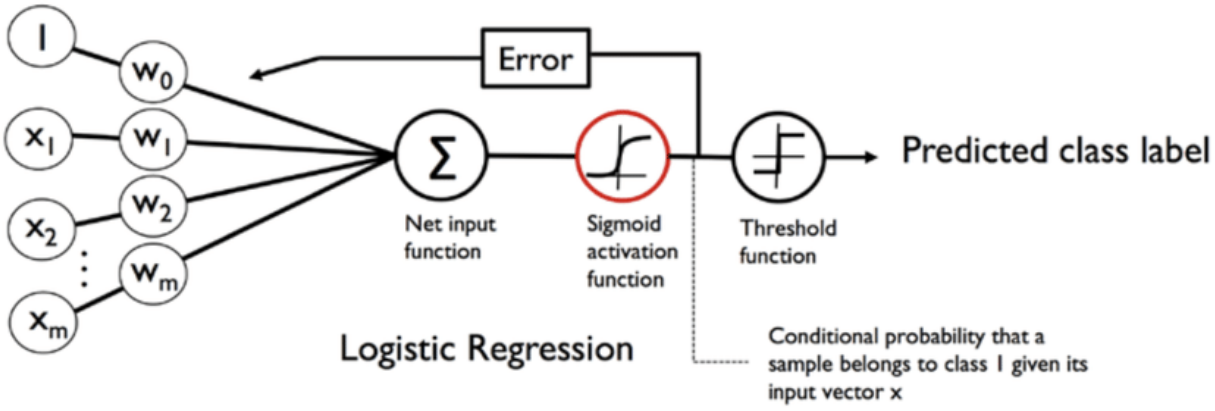


Figure 2: Architecture Diagram of Logistic Regression

2.3.2 K Means Neighbor (KNN)

KNN is a classification algorithm that belongs to supervise learning. It categorizes the entity that reliant on nearest neighbor. The output result of the algorithmic program depends on K-nearest neighbor class that enforced by finding K- variety of coaching points nearest to the specified character and contemplate the votes among the K object [5].

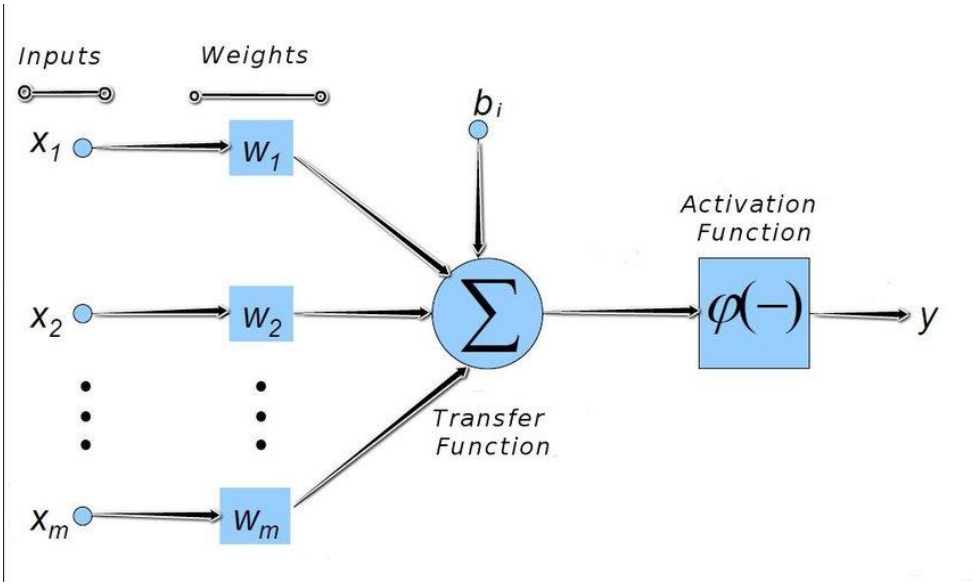


Figure 3: Architecture Diagram of K Means Algorithm

The ANN consist of various number of hidden layers with different number of units besides input and output layers. The first layer receives the inputs from outside and transmits to hidden layers. Hidden layers process the data in their turns and transmit to the output layer. Figure 3 shows the basic architecture of an ANN network [6].

2.3.3 Support Vector Machine (SVM)

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 [7]. The architecture of SVM is shown in Fig. 4. However, SVM cannot deal efficiently with large data samples, as in the case of this study. In SSVM, smoothing techniques are applied to solve important mathematical programming problems and the ε− in-sensitive loss function is replaced by the squares of 2-norm ε− insensitive loss function.

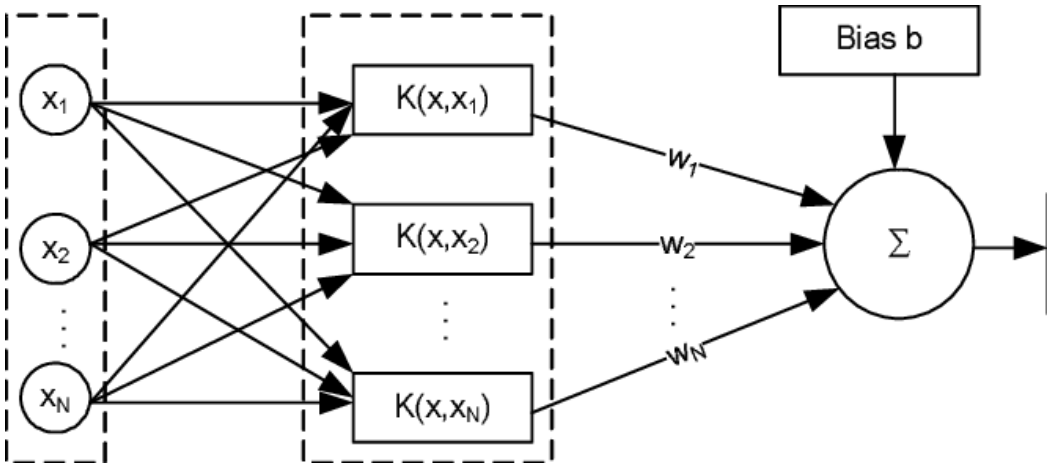


Figure 4: Architecture Diagram of SVM

2.3.4 Gaussian Naive Bayes

Naive Bayes Classifiers are based on the Bayes Theorem. When working with continuous data, an assumption often taken is that the continuous values associated with each class are distributed according to a normal (or Gaussian) distribution [8]. The likelihood of the features is assumed to be-

$$\frac{1}{\sqrt{2\pi_{y}^{\sigma^2}}}\exp\left(-\frac{\left(x_i-\mu_y\right)^2}{2\sigma_y^2}\right)$$

An approach to create a simple model is to assume that the data is described by a Gaussian distribution with no co-variance (independent dimensions) between dimensions. This model can be fit by simply finding the mean and standard deviation of the points within each label, which is all what is needed to define such a distribution.

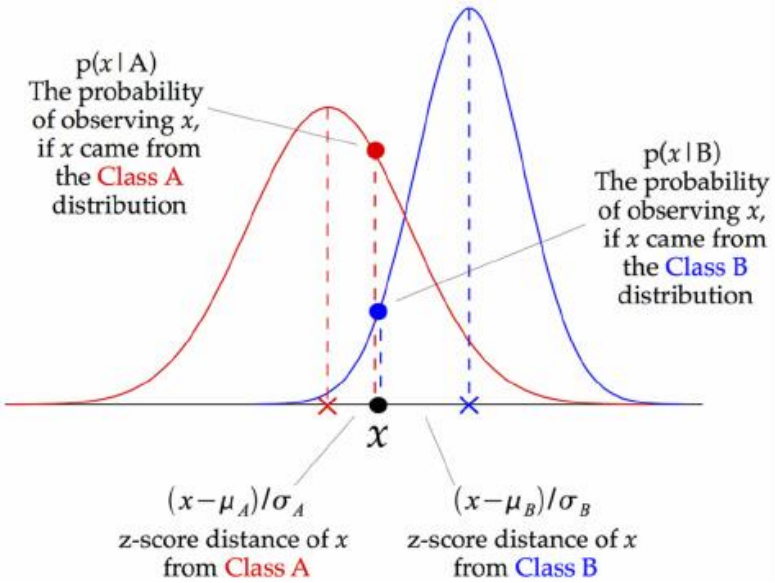


Figure 5: Architecture Diagram of Gaussian Naïve Bayes

The above illustration indicates how a Gaussian Naive Bayes (GNB) classifier works. At every data point, the z-score distance between that point and each class-mean is calculated, namely the distance from the class mean divided by the standard deviation of that class. Thus, we see that the Gaussian Naive Bayes has a slightly different approach and can be used efficiently.

2.3.5 Decision Tree

Decision Tree is an algorithm that classifies parameters in categorical form in spite of arithmetic data. Tree like structure is created by DT. Many large data set related to medical have analyzed by DT due to its simple nature. It works on tree node for analysis. Leaf Node: Signify the solution of every Test Interior Node: Handle numerous element Main Node [Root Node]: Other nodes work based on main node Data is to be divided into two or more parallel set by applying this algorithm. Then entropy of each parameter is calculated. After that divide the data with predictor having extreme information gain that means minimum entropy.

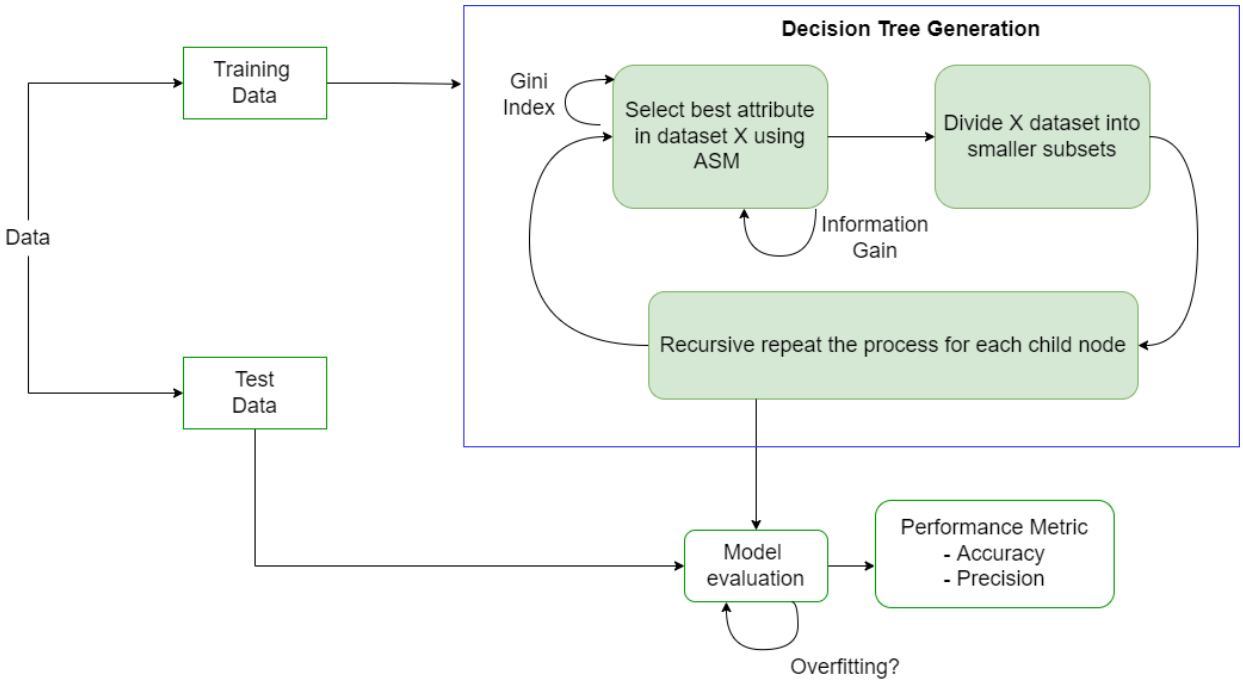


Figure 6: Architecture diagram of Decision Tree

For classification problems, we using entropy which is defines the randomness in the processed information and measures the amount of uncertainty in it. More the entropy, the more complex the scenario to draw conclusions. The overall objective is to minimize entropy and have more homogeneous decision regions wherein data points belong to a similar class. The metric measures the chances or likelihood of a randomly selected data point misclassified by a particular node. The cost function for evaluating feature splits in a dataset is the Gini index. The IG metric measures the reduction in entropy or Gini index due to a feature split. Informative splits are achieved when tree-based algorithms use Entropy or Gini index as criteria. In other words, such a split reduces the requirements by a maximum amount [9].

2.3.6 Random Forest

Random Forest is a classifier that contains several decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset. It is based on the concept of ensemble learning which is a process of combining multiple classifiers to solve a complex problem and improve the performance of the model [10].

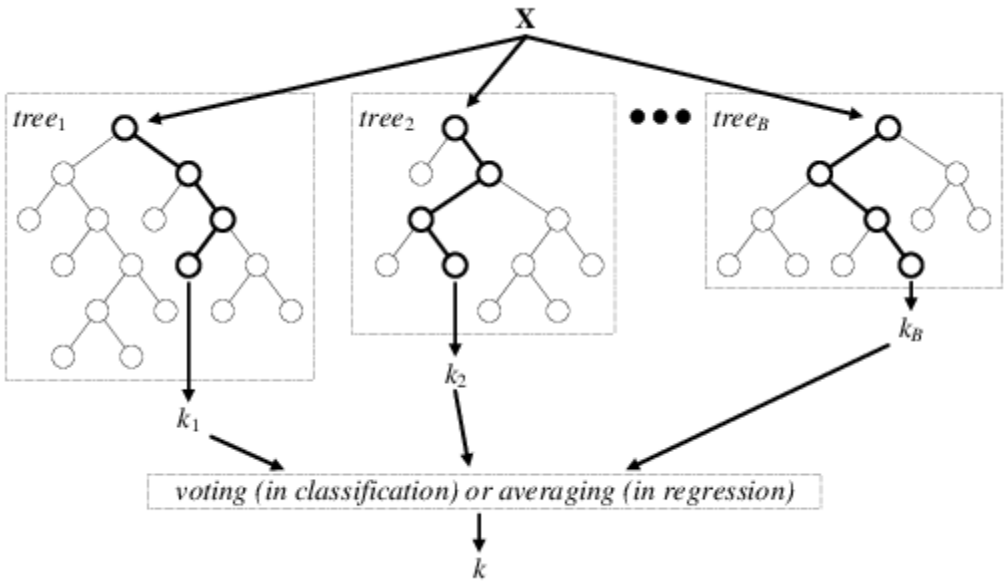


Figure 7: Architecture Diagram of Random Forest

From the above figure, we used random forest (RF) for predictive modeling. In classification, RF is a committee of decision trees, where the final output is derived from voting. The core idea of RF is to combine many (Bin total) decision trees, built using different bootstrap samples of the original dataset and a random subset (of predetermined size q) of features x^1,. . .,x^p Voting will take place by averaging the decision tree. Finally, select the most voted prediction result as the final prediction result.

2.3.7 AdaBoost

The AdaBoost method to increase accuracy. AdaBoost divides the training dataset into a number of instances. These instances are then labeled with a weighted value. Error value and therefore stage value is calculated for each instance. So that weak instances are classified. It uses a loss function to minimize loss and converge upon a final output value. The loss function optimization is done using gradient descent, and hence the name gradient boosting. Further, gradient boosting uses short, less-complex decision trees instead of decision stumps.

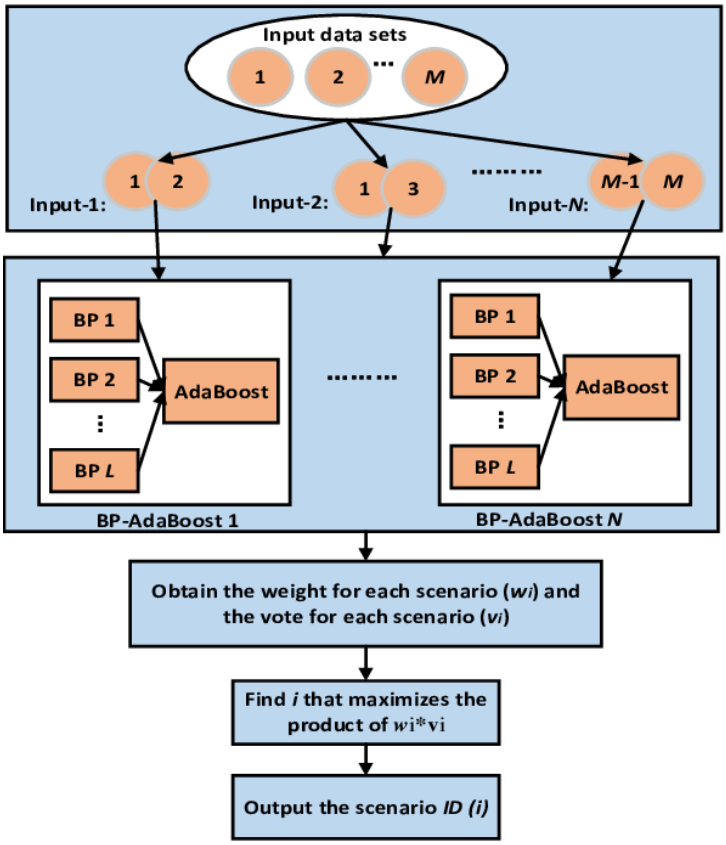


Figure 8: Architecture Diagram of AdaBoost

This figure shows the proposed advanced BP-AdaBoost classification algorithm, which employs a number of BP-AdaBoost sub-classifiers in parallel. To make more reliable radio scenario recognition, both the offline training performance for each scenario and the online classification results from each sub-classifier are taken into account during the decision fusion [11].

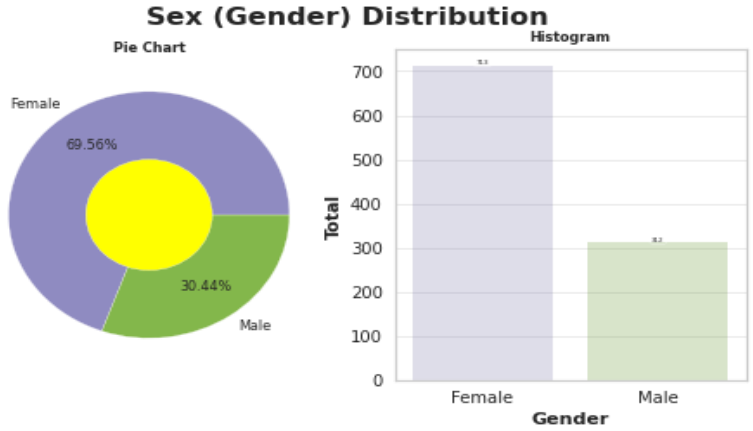
CHAPTER 3

IMPLEMENTATION

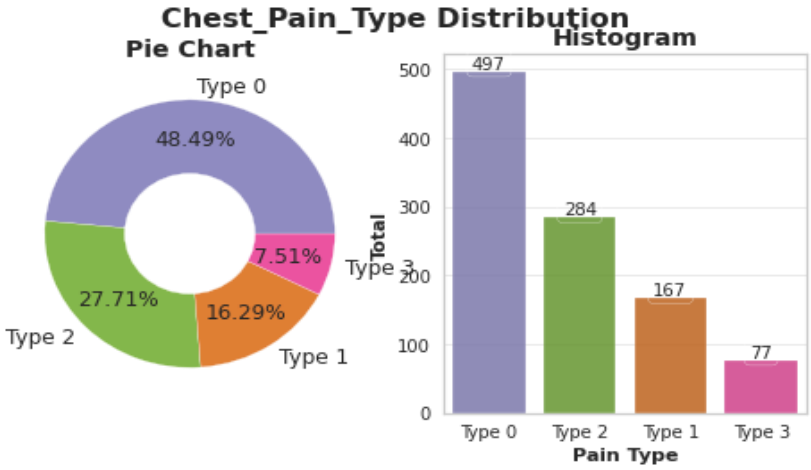
3.1 Overview of the Experiment

Since our project is into two parts, we did Exploratory Data Analysis and accuracy level based on machine learning algorithms. Initially we did data exploration based on features from the dataset which we provide visual representation. Then we did one-hot encoding in feature engineering for better prediction.

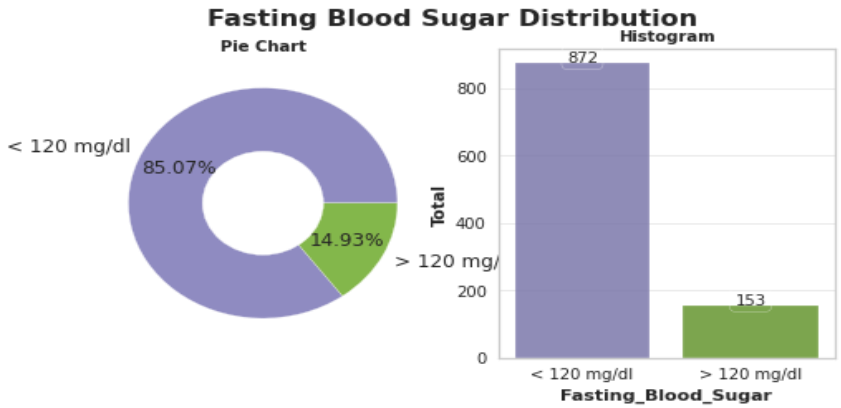
The very first variable “sex(Gender)” that we explore as categorical value here:



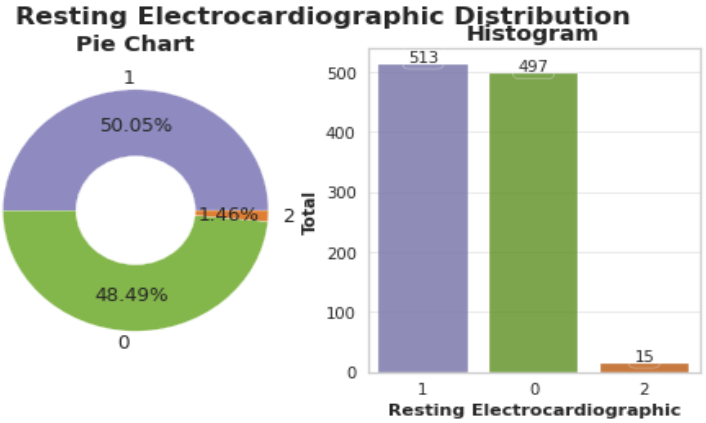
Comparing male patients to females which are the highest.



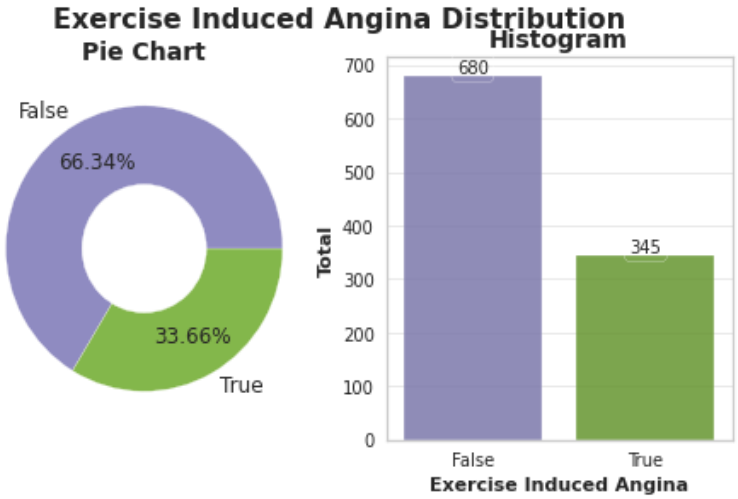
Various types of chest pain are shown where type 0 has the highest number.



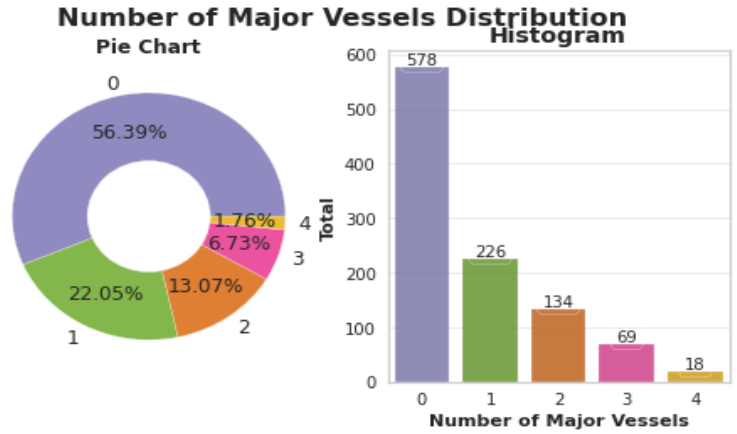
Graphical representation of Fasting Blood Sugar level



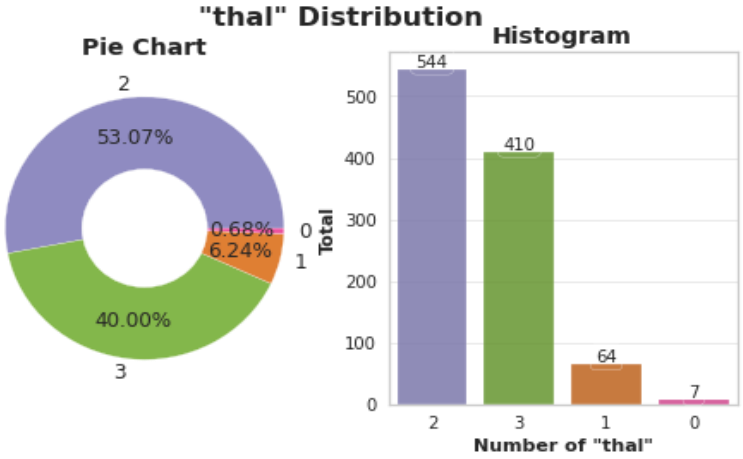
Visualization of Resting Electrocardiographic distribution.



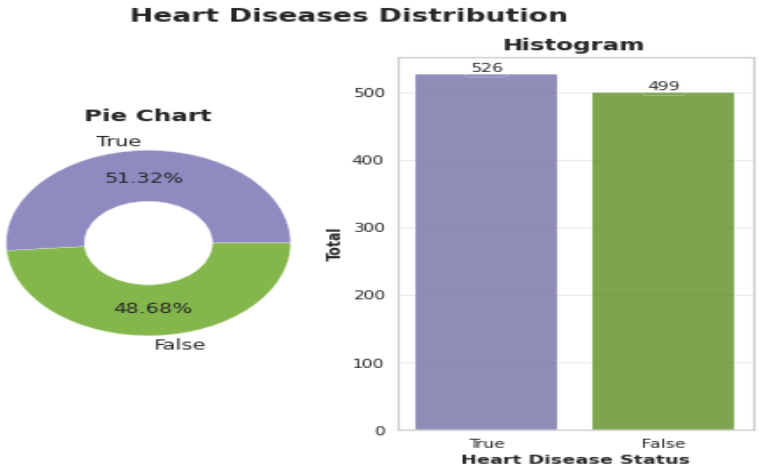
Patients with no exercise induced angina are the highest number.



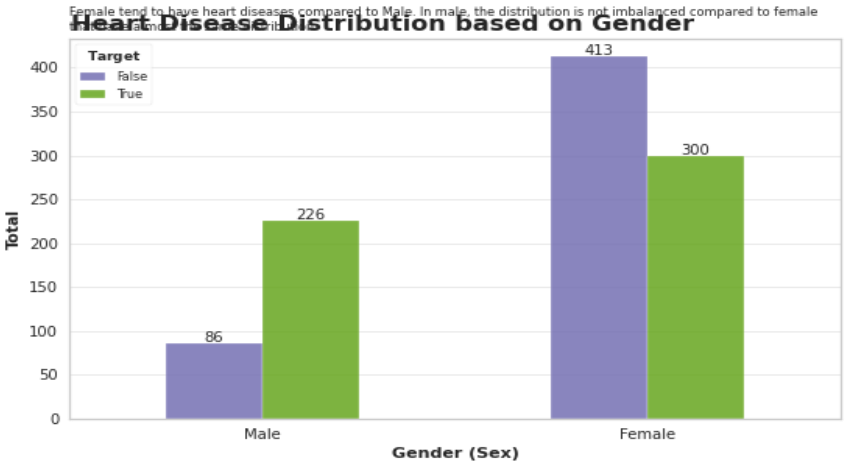
Graphical representation of Major Vessels Distribution



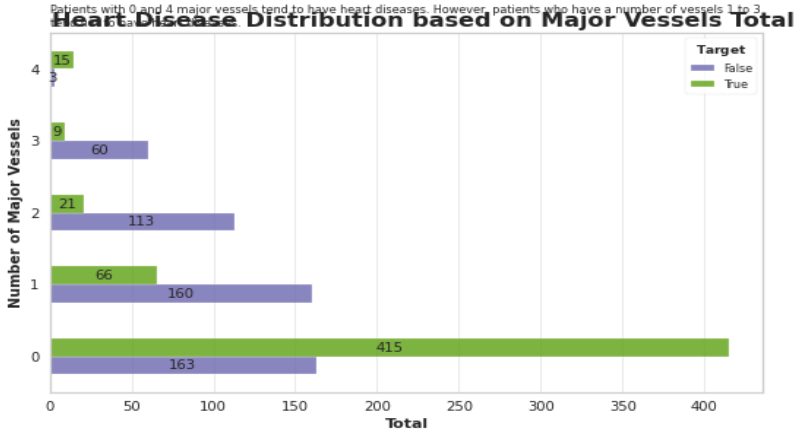
Thal distribution



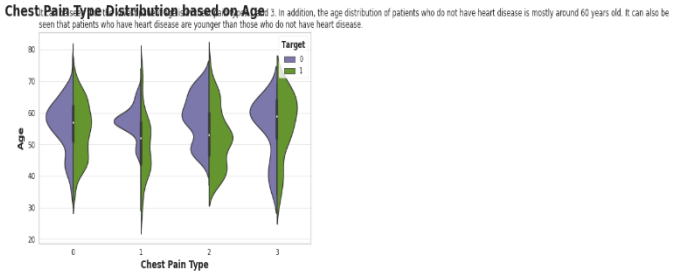
The graph shows the highest number of patients who have heart disease.



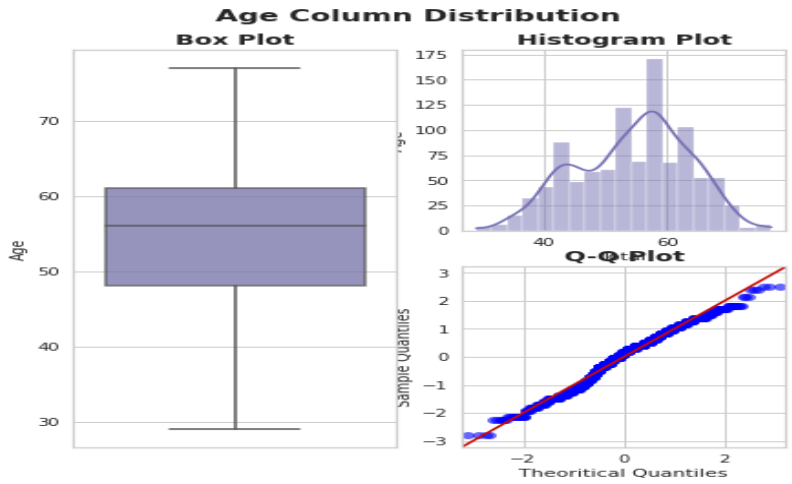
This graph shows heart disease, based on Gender



Heart disease based on Major Vessels

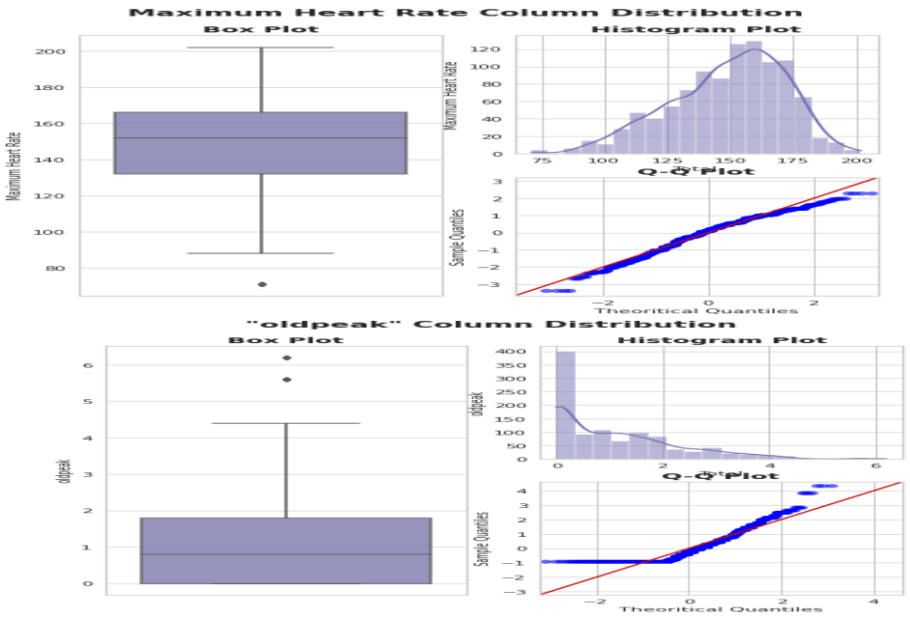


Chest pain type based on age

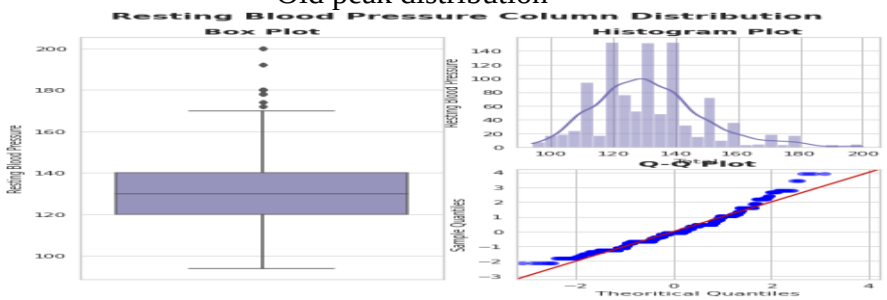


Age Column Distribution

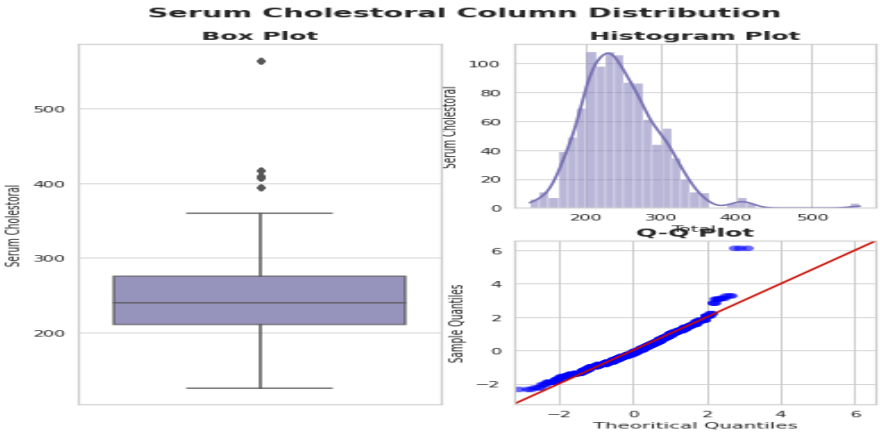
From the histogram and boxplot, it can be seen that this column is normally distributed. From the Q-Q plot, the data values tend to closely follow the 45-degree, which means the data is likely normally distributed.



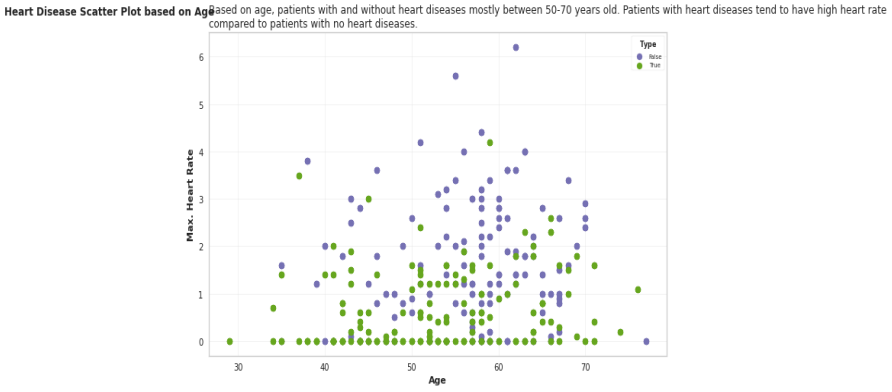
Old peak distribution



Resting blood pressure column distribution



Serum Cholesterol Distribution



Heart Disease based on disease

3.2 Feature Engineering

In this segment, we work with one-hot encoding for better results.

```
# --- Creating Dummy Variables for cp, thal and slope ---
cp = pd.get_dummies(df['cp'], prefix='cp')
thal = pd.get_dummies(df['oldpeak'], prefix='oldpeak')
slope = pd.get_dummies(df['slope'], prefix='slope')

# --- Merge Dummy Variables to Main Data Frame ---
frames = [df, cp, thal, slope]
df = pd.concat(frames, axis = 1)
```

One-hot encoding for CP, Old Peak and slope

3.3 Training and test generation

For generating training, testing and splitting we did implement.

```
[ ]  # --- Data Normalization using Min-Max Method ---
     x = MinMaxScaler().fit_transform(x)
```

```
[ ]  # --- Splitting Dataset into 80:20 ---
     x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=4)
```

```
[ ]  # --- Applying Logistic Regression ---
     LRclassifier = LogisticRegression(max_iter=1000, random_state=1, solver='liblinear', penalty='l1')
     LRclassifier.fit(x_train, y_train)

     y_pred_LR = LRclassifier.predict(x_test)
```

3.4 Running the classifier

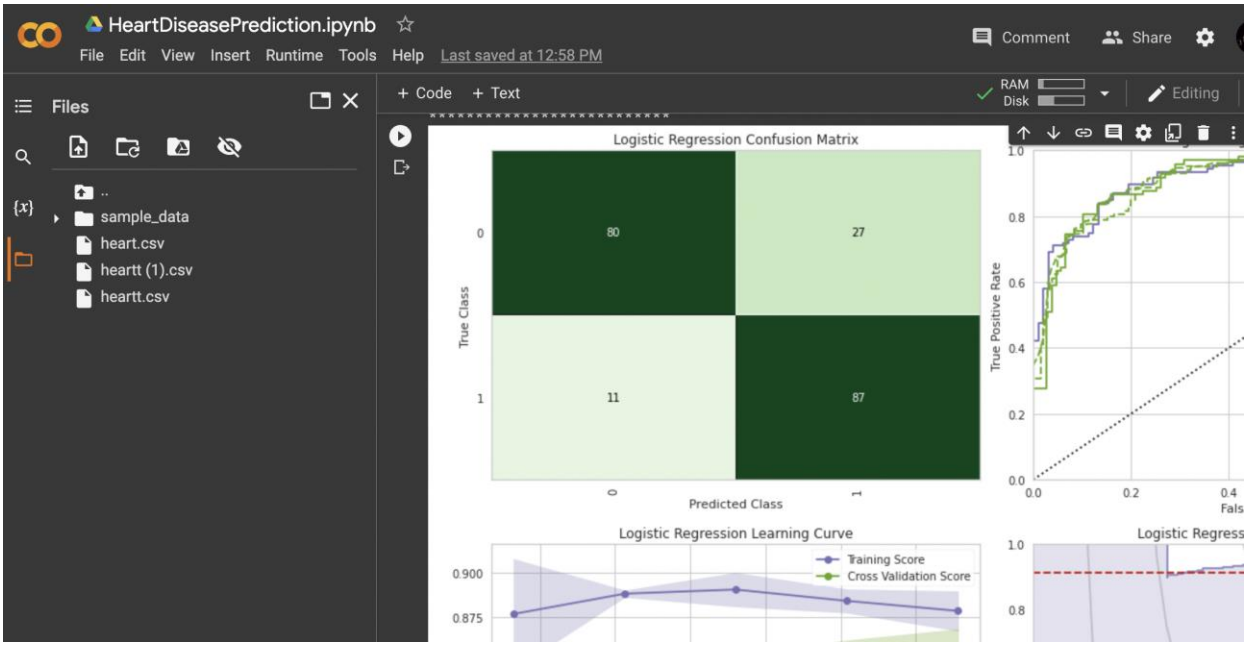
We have already used the models that are previously described.

CHAPTER 4

RESULT ANALYSIS

4.1 About Google Colab

Colab is a free Jupyter notebook environment that runs entirely in the cloud. Most importantly, it does not require a setup and the notebooks that we create can be simultaneously edited by our team members - just the way we edit documents in Google Docs. Colab supports many popular machine learning libraries which can be easily loaded in our notebook. It is a very comfortable tool. By importing various libraries of python programming, we can work on larger data set and analysis and visualize with different graphs of data in real time. With Colab, data cleaning, numerical



imitation, statistical modeling and many work have done.

4.2 Confusion Matrix

The confusion matrix is created by various classifier and it includes expected and real classifications information. The confusion matrix provides analysis to judge the effectiveness of proposed methodology [12].
It is one of the most intuitive and easiest metrics used for finding the correctness and accuracy of

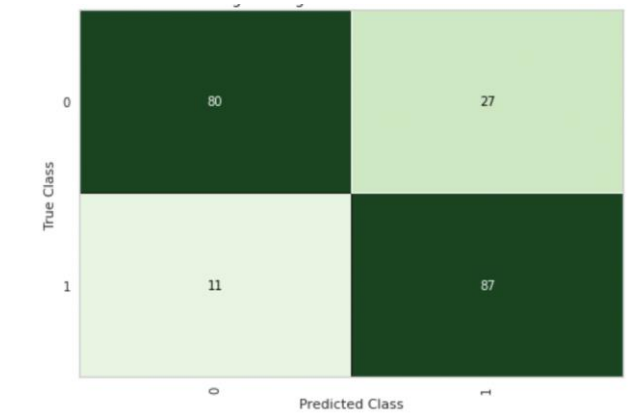
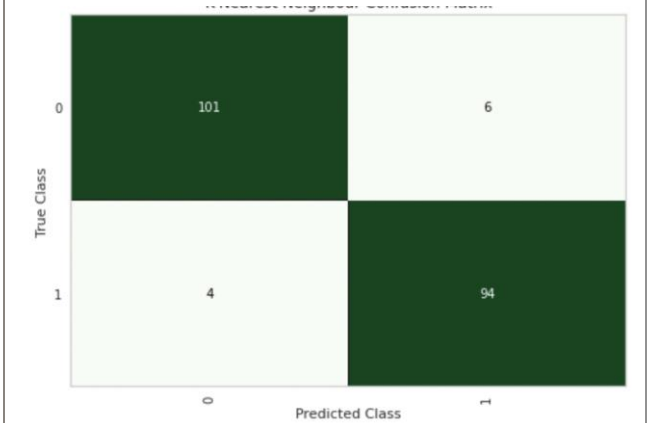
the model. It is used for Classification problem where the output can be of two or more types of classes. Where,
The number of actual negative cases in the data = Condition Negative (N)
Condition Negative (N) = Total number of negative cases
Condition Positive (P) = Total number of positive cases
True Positive (TP) = number of correct positive prediction
True Negative (TN) = number of correct negative prediction
False Positive (FP) = Type I Error, No. of incorrect positive prediction
False Negative (FN) = Type II Error, No. of incorrect negative prediction

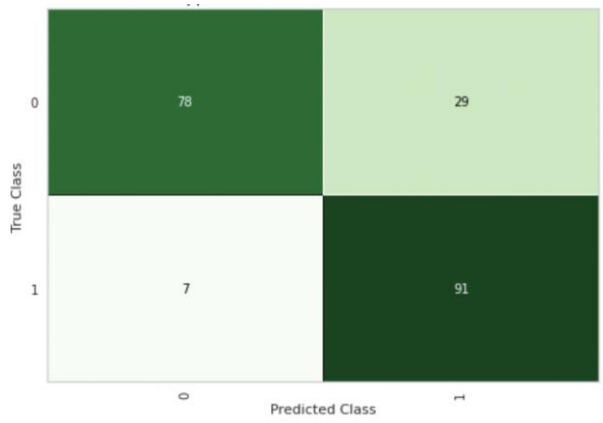
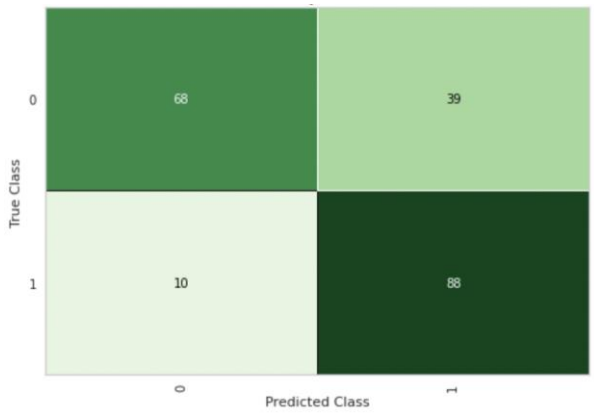
Accuracy: The accuracy of classification process is based on correct and incorrect predictions. Accuracy of the classification can be calculated by

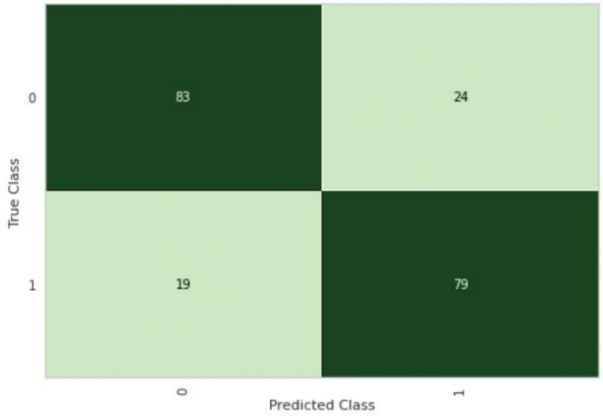
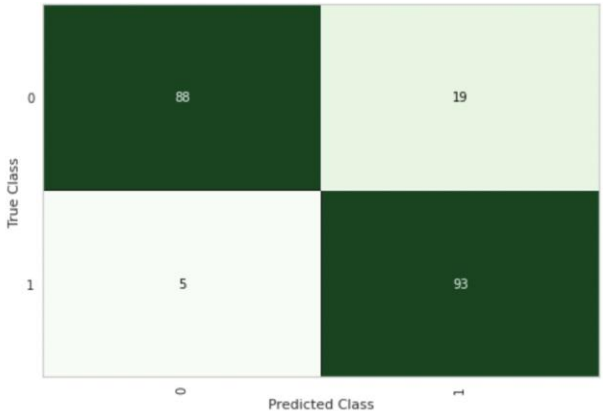
Accuracy(ACC)

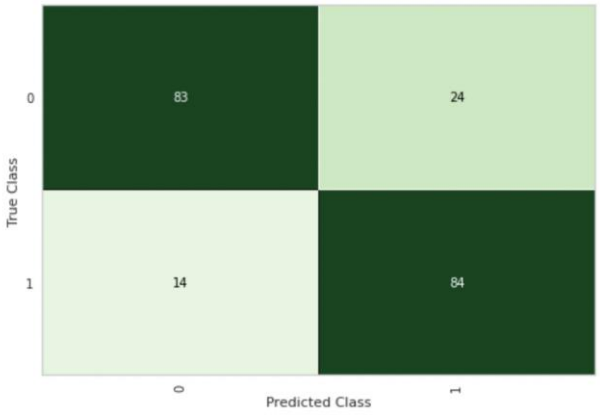
$$ACC = \frac{TP + TN}{P + N} = \frac{TP + TN}{TP + TN + FP + FN}$$

The confusion matrix of various ML algorithm

CM for Logistic Regression	CM for KNN
	

CM for Support Vector Machine	CM for Gaussian Naive Bayes
	

CM for Decision Tree	CM for Random Forest
	

CM for Adaboost	
	

4.3 Accuracy of Various Models

After Training and Testing by using various ML approach we get that accurateness of the K-Nearest Neighbor is far proficient as relate to other algorithms. To Find the accuracy of each algorithm we use confusion matrix as shown in Table before. And it is concluding that KNN is best among them with 95.121% accuracy and the comparison is shown in Table below

SL.NO	Model	Accuracy
1	**K – Nearest Neighbor**	**95.121**
2	**Random Forrest**	**88.292**
3	**Support Vector Machine**	**82.43**
4	**Logistic Regression**	**81.46**
5	**AdaBoost**	**81.46**
6	**Decision Tree**	**79.02**
7	**Gaussian Naïve Bayes**	**76.09**

CHAPTER 5

CONCLUSION

Heart acts a major role in corporeal organism. The diseases of heart want more perfection and exactness for diagnose and analyses. In real time heart diseases may not be detect in early stage. This need further analysis. In proposed work, an accurate and early heart diseases prediction is presented by using data set of heart diseases. The presented methodology requires various ML algorithms. The analysis is carried out based on Confusion matrix and comparing accuracy among them and get SVM is finest algorithm. Thus the efficacy of presented work has been verified. This technique may be used as a support for early and accurate prediction of heart disease. There are many more ML algorithms that can be used for finest exploration and for earlier prediction of heart diseases for the upcoming possibility. This needs further diagnosis.

References

[1] E.Taylor,P.s.Ezekiel,F.B.Deedam. (2019). “A Model to Detect Heart Disease using Machine Learning algorithm” International journal of Computer Science and engineering.vol-7, issue-11

[2] R. Goel and A. Jain. (2018) "The Implementation of Image Enhancement Techniques on Color and Gray Scale IMAGEs," 2018 Fifth International Conference on Parallel, Distributed and Grid Computing (PDGC), , pp. 204-209, doi: 10.1109/PDGC.2018.8745782

[3] Archana Singh, Rakesh k. (2020). ”Heart disease Prediction Using machine Learning Algorithms” International Conferences On Electrical and electronics Engineering (ICE3)

[4] https://archive.ics.uci.edu/ml/datasets/heart+disease

[5] Devansh Shahet.al.(2020) “HeartDiseasePredictionusingMachineLearningTechniques” © Springer Nature Singapore Pte Ltd

[6]https://www.researchgate.net/publication/325209169_An_Intelligent_Software_for_Measurements_of_Biological_Materials_BioMorph#pf4

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

[8] https://iq.opengenus.org/gaussian-naive-bayes

[9] https://www.spiceworks.com/tech/artificial-intelligence/articles/what-is-decision-tree

[10] https://www.simplilearn.com/tutorials/machine-learning-tutorial/random-forest-algorithm

[11]https://www.researchgate.net/publication/323600319_Machine_LearningAided_Radio_Scenario_Recognition_for_Cognitive_Radio_Networks_in_Millimeter-Wave_Bands#pf9

[12] Goel R., Jain A. (2020) Improved Detection of Kidney Stone in Ultrasound Images Using Segmentation Techniques. In: Kolhe M., Tiwari S., Trivedi M., Mishra K. (eds) Advances in Data and Information Sciences. Lecture Notes in Networks and Systems, vol 94. Springer, Singapore. https://doi.org/10.1007/978-981-15-0694-9_58