

**AMERICAN INTERNATIONAL UNIVERSITY BANGLADESH**

Final Term Project

**Introduction To Data Science**

**Name: Shakib Sadat Shanto**

**ID: 20-43074-1**

**Section: D**

**Project Overview:**

In this project, basic data analysis and preprocessing was conducted on selected dataset and K Nearest Neighbors algorithm was applied on the dataset using R programming language. Confusion matrix of the model have also been discussed in the Confusion Matrix section of the report.

**Dataset Overview:**

The dataset used for the project was collected from Kaggle. The name of the dataset is **Heart Failure Prediction Dataset**. Here is the URL of the dataset:

<https://www.kaggle.com/datasets/fedesoriano/heart-failure-prediction>

The dataset contains 918 observations. It has 11 independent variables.

1. Age: age of the patient [years]
2. Sex: sex of the patient [M: Male, F: Female]
3. ChestPainType: chest pain type [TA: Typical Angina, ATA: Atypical Angina, NAP: Non-Anginal Pain, ASY: Asymptomatic]
4. RestingBP: resting blood pressure [mm Hg]
5. Cholesterol: serum cholesterol [mm/dl]
6. FastingBS: fasting blood sugar [1: if FastingBS > 120 mg/dl, 0: otherwise]
7. RestingECG: resting electrocardiogram results [Normal: Normal, ST: having ST-T wave abnormality (T wave inversions and/or ST elevation or depression of > 0.05 mV), LVH: showing probable or definite left ventricular hypertrophy by Estes' criteria]
8. MaxHR: maximum heart rate achieved [Numeric value between 60 and 202]
9. ExerciseAngina: exercise-induced angina [Y: Yes, N: No]
10. Oldpeak: oldpeak = ST [Numeric value measured in depression]
11. ST\_Slope: the slope of the peak exercise ST segment [Up: upsloping, Flat: flat, Down: downsloping]

1 dependent or class variable.

1. HeartDisease: output class [1: heart disease, 0: Normal]

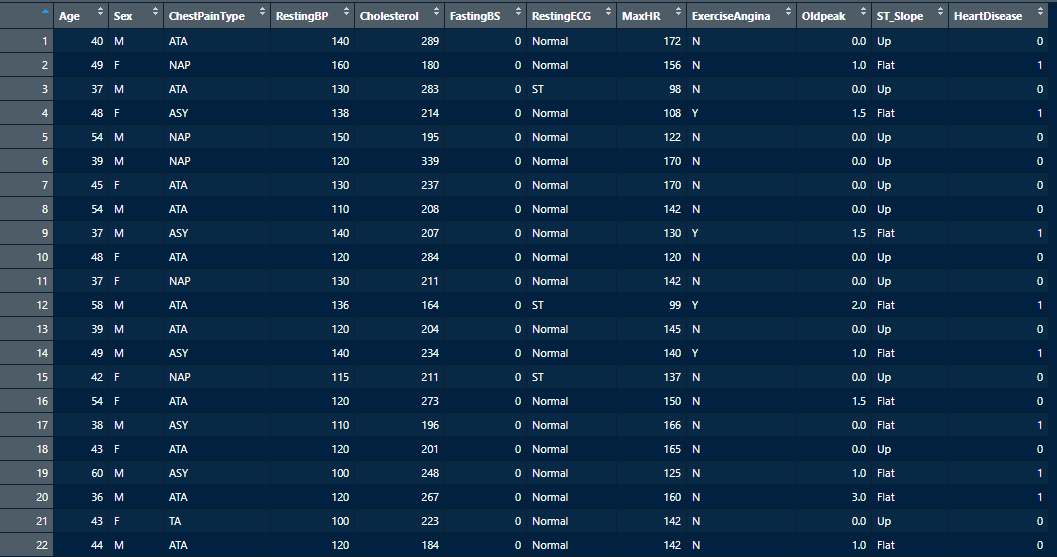
**Data Analysis and Preprocess:**

1. Importing Dataset:

Code:

dataset = read.csv('heart.csv')

Output:

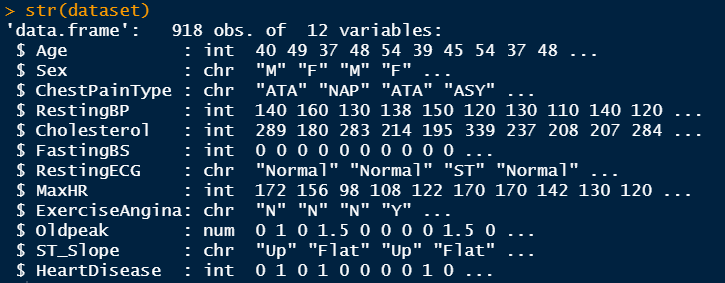


1. Structure of the Dataset:

Code:

str(dataset)

Output:

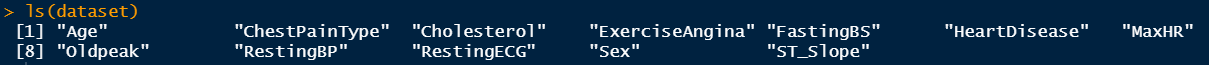


1. Attributes:

Code:

ls(dataset)

Output:



1. Unique Values of Categorical Attributes:

Code:

unique(dataset$Sex)

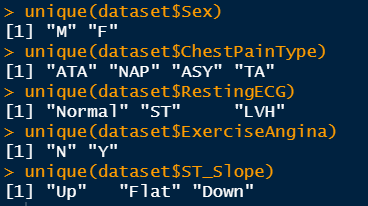
unique(dataset$ChestPainType)

unique(dataset$RestingECG)

unique(dataset$ExerciseAngina)

unique(dataset$ST\_Slope)

Output:

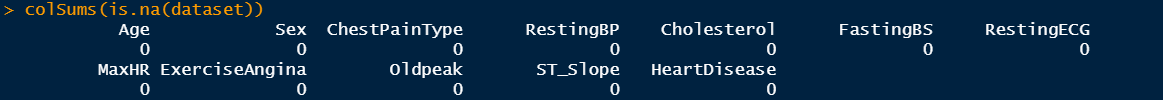


1. Check for missing values:

Code:

colSums(is.na(dataset))

Output:



1. Encoding Categorical Values into Numeric Values:

Code:

dataset$Sex = factor(dataset$Sex,

levels = c("M","F"),

labels = c(0,1))

dataset$ChestPainType = factor(dataset$ChestPainType,

levels = c("ATA","NAP","ASY","TA"),

labels = c(1,2,3,4))

dataset$RestingECG = factor(dataset$RestingECG,

levels = c("Normal","ST","LVH"),

labels = c(1,2,3))

dataset$ExerciseAngina = factor(dataset$ExerciseAngina,

levels = c("N","Y"),

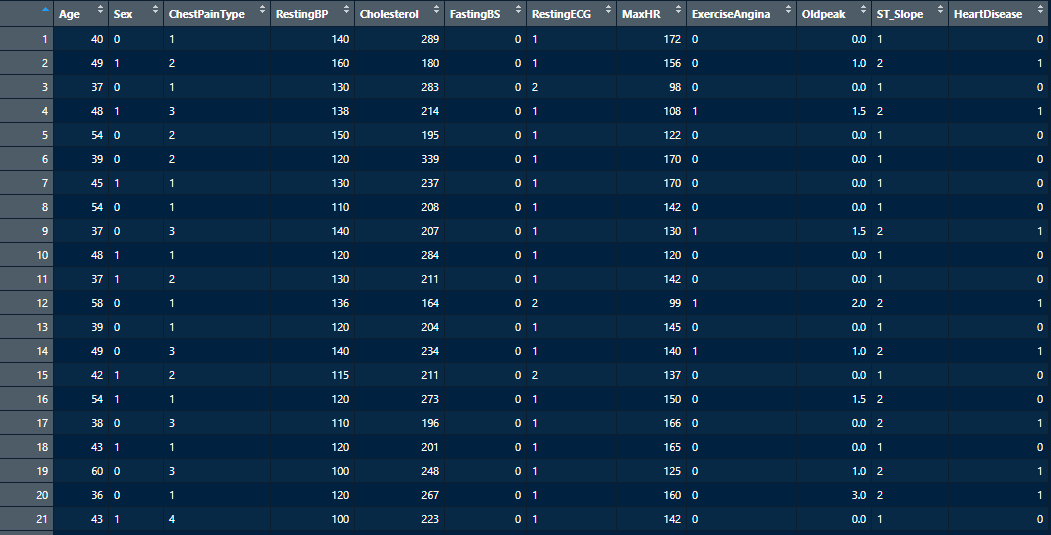
labels = c(0,1))

dataset$ST\_Slope = factor(dataset$ST\_Slope,

levels = c("Up","Flat","Down"),

labels = c(1,2,3))

Output:



1. Feature Scaling (Normalization):

Code:

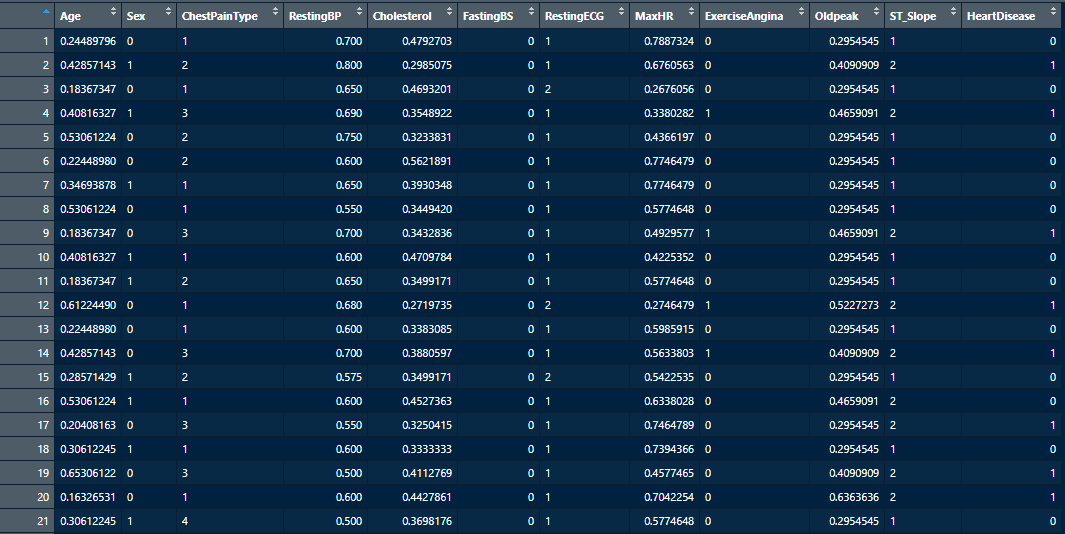
install.packages("caret")

library(caret)

process = preProcess(as.data.frame(dataset), method=c("range"))

dataset = predict(process, as.data.frame(dataset))

Output:

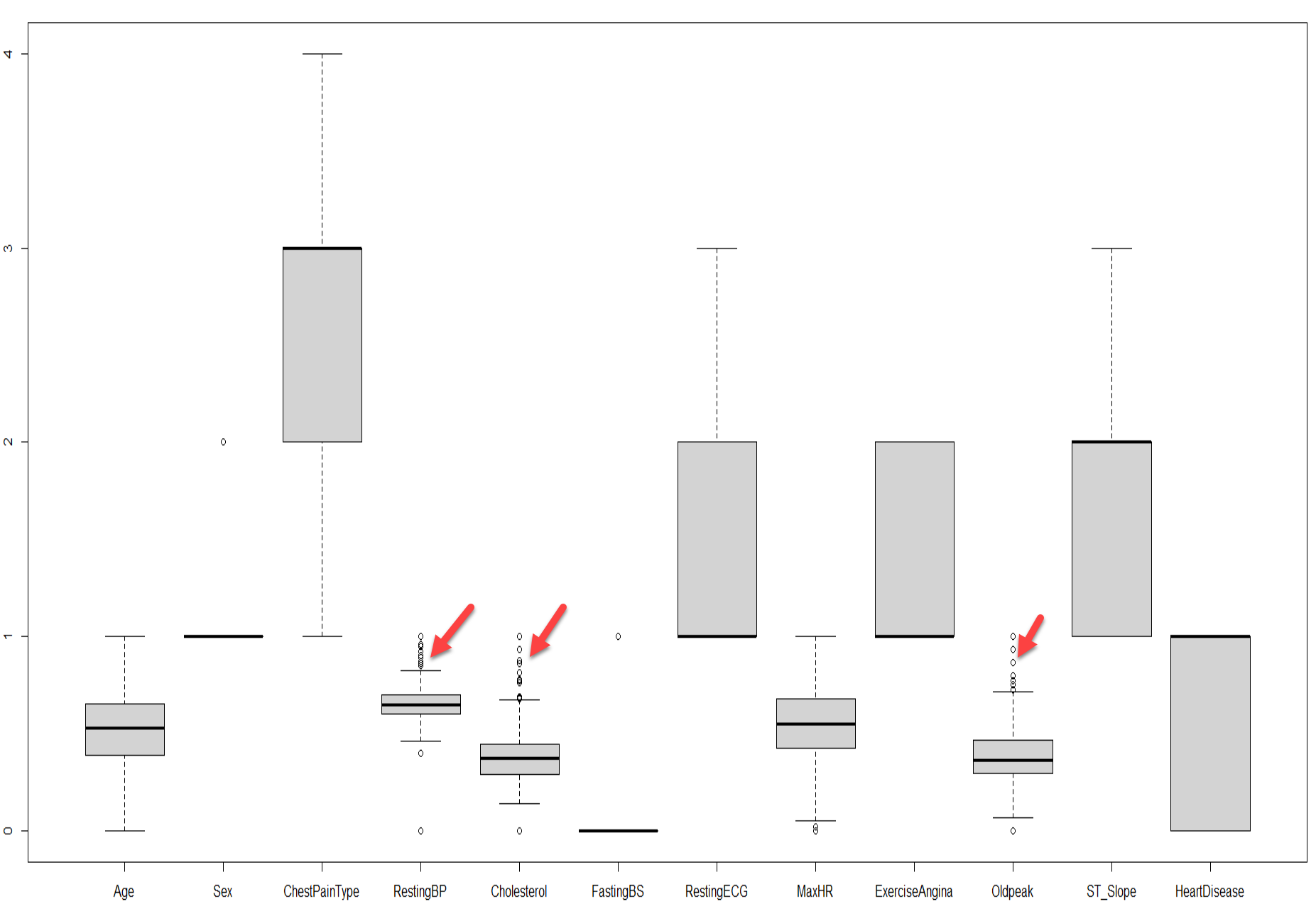


1. Outlier Detection and Removing:

Code:

boxplot(dataset)

Output:



Code:

summary(dataset$RestingBP)

Iqr\_restingbp = .70-.60

upfen\_restingbp = .70+1.5\*Iqr\_restingbp

low\_restingbp = .60-1.5\*Iqr\_restingbp

upfen\_restingbp

low\_restingbp

summary(dataset$Cholesterol)

Iqr\_Cholesterol = 0.4428-0.2873

upfen\_Cholesterol = .4428+1.5\*Iqr\_Cholesterol

low\_Cholesterol = 0.2873 -1.5\*Iqr\_Cholesterol

upfen\_Cholesterol

low\_Cholesterol

summary(dataset$Oldpeak)

Iqr\_Oldpeak = 0.4659-0.2955

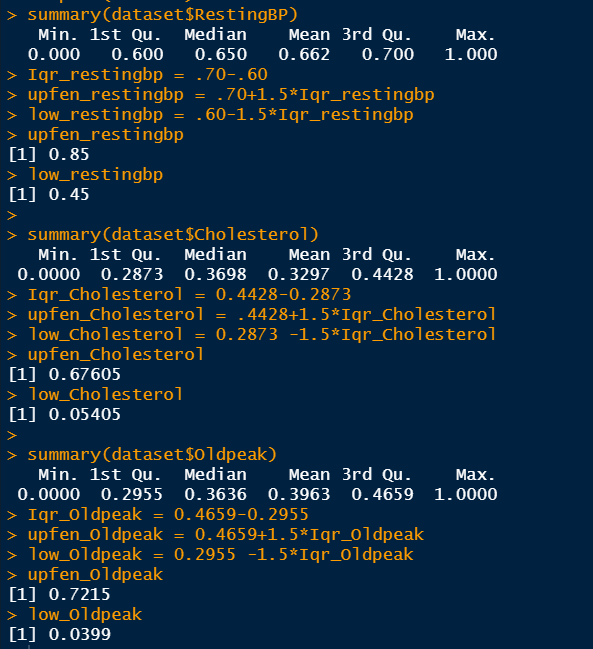
upfen\_Oldpeak = 0.4659+1.5\*Iqr\_Oldpeak

low\_Oldpeak = 0.2955 -1.5\*Iqr\_Oldpeak

upfen\_Oldpeak

low\_Oldpeak

Output:



Code:

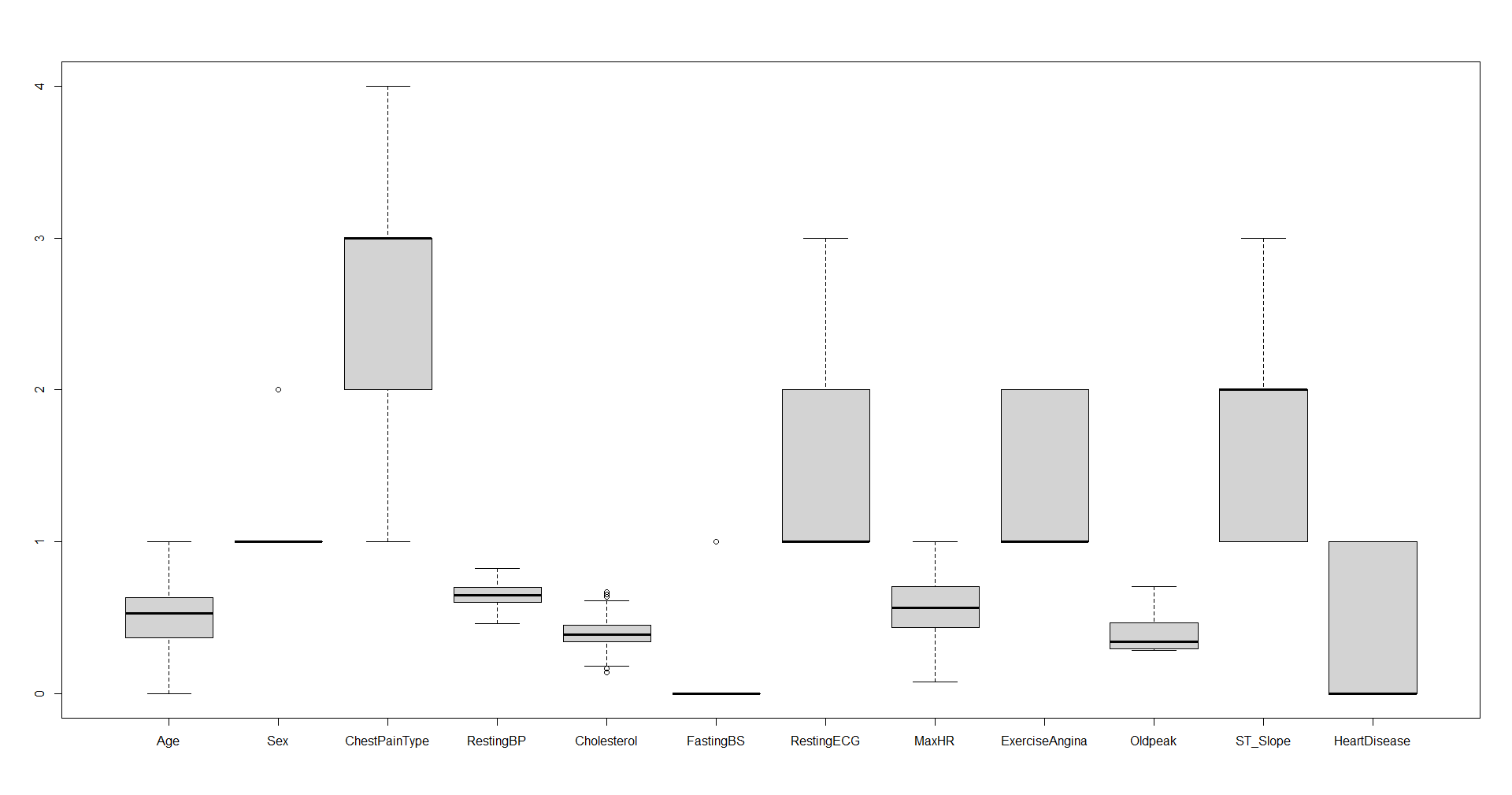
dataset\_rm\_outlier = subset(dataset, dataset$RestingBP > low\_restingbp & dataset$RestingBP < upfen\_restingbp & dataset$Cholesterol > low\_Cholesterol & dataset$Cholesterol < upfen\_Cholesterol

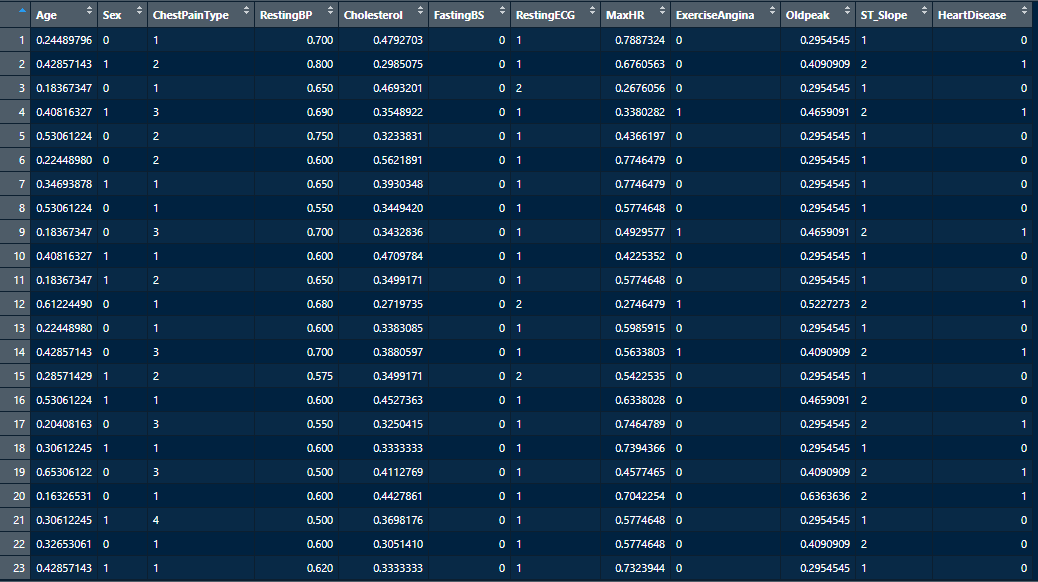
& dataset$Oldpeak > low\_Oldpeak & dataset$Oldpeak < upfen\_Oldpeak )

boxplot(dataset\_rm\_outlier)

dataset = dataset\_rm\_outlier

Output:



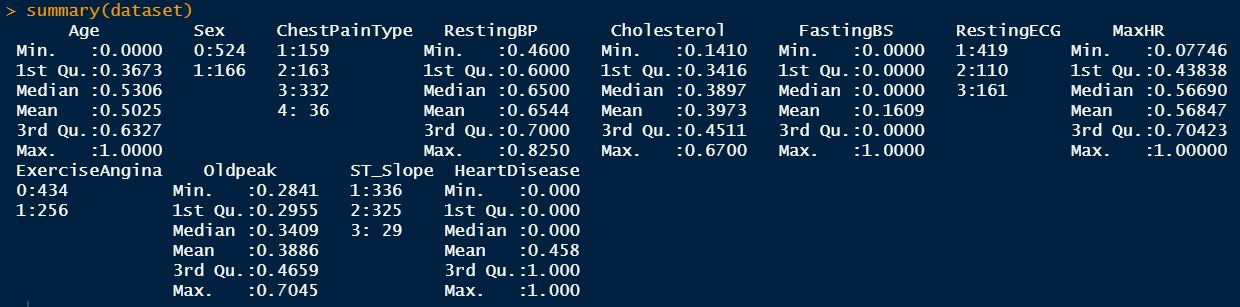


1. Summary and Histogram:

Code:

summary(dataset)

Output:



Code:

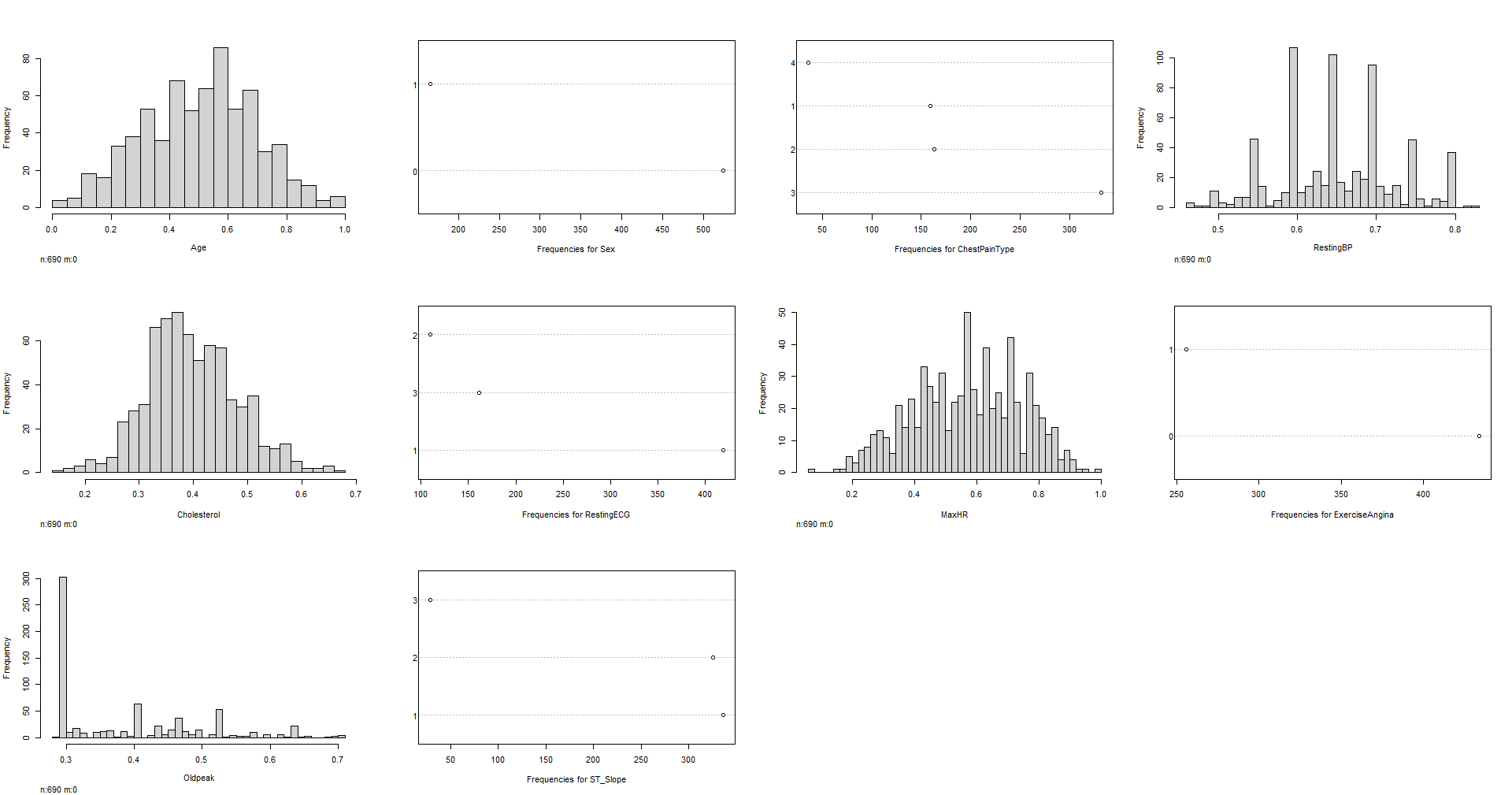
install.packages("Hmisc")

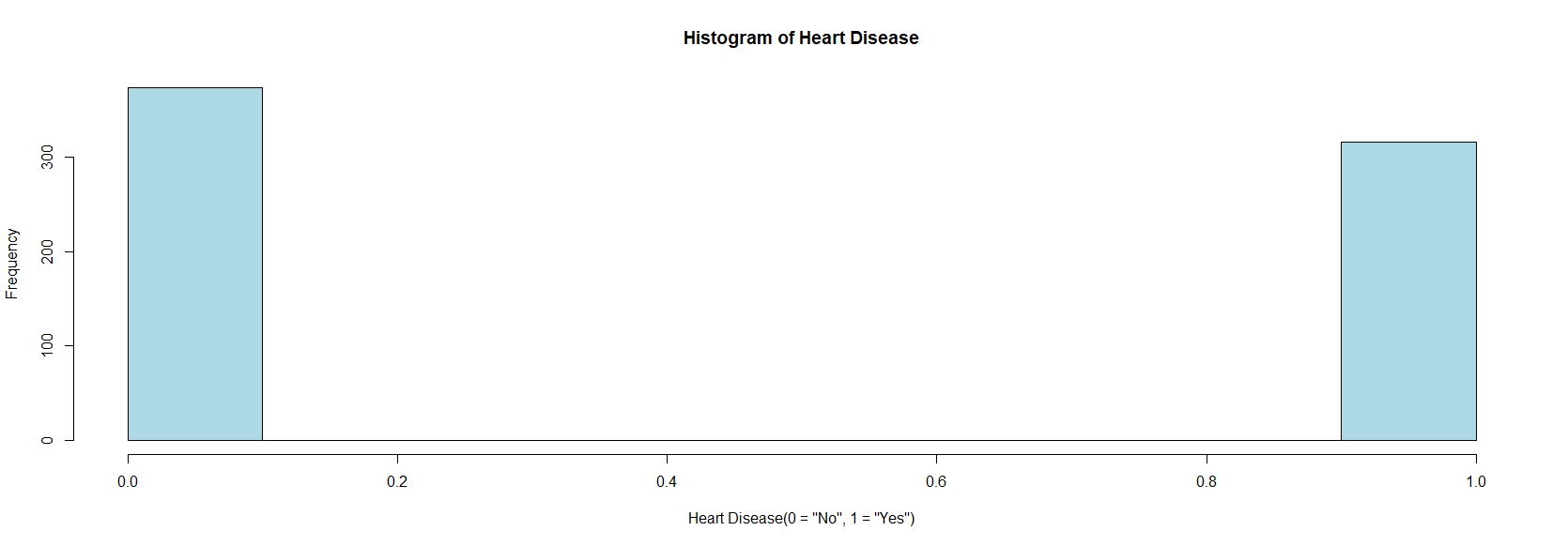
library(Hmisc)

hist.data.frame(dataset)

hist(dataset$HeartDisease,xlab = 'Heart Disease(0 = "No", 1 = "Yes")',main = paste("Histogram of Heart Disease"), col=("lightblue"))

Output:





**Model Building:**

1. Splitting Dataset into Training and Test Set:

Code:

install.packages("caTools")

library(caTools)

set.seed(123)

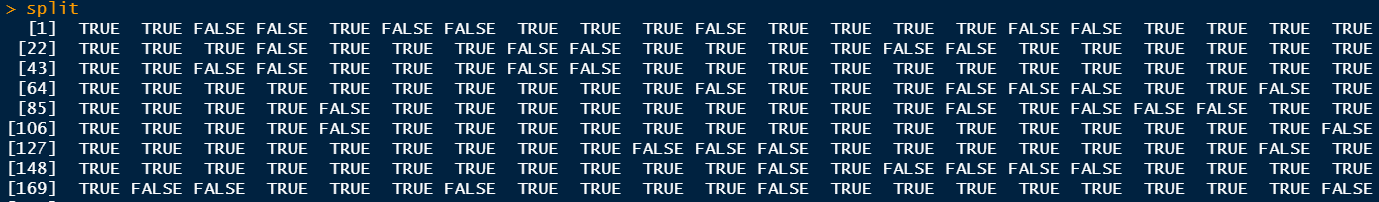
split = sample.split(dataset$HeartDisease, SplitRatio = 0.80)

training\_set = subset(dataset, split==TRUE)

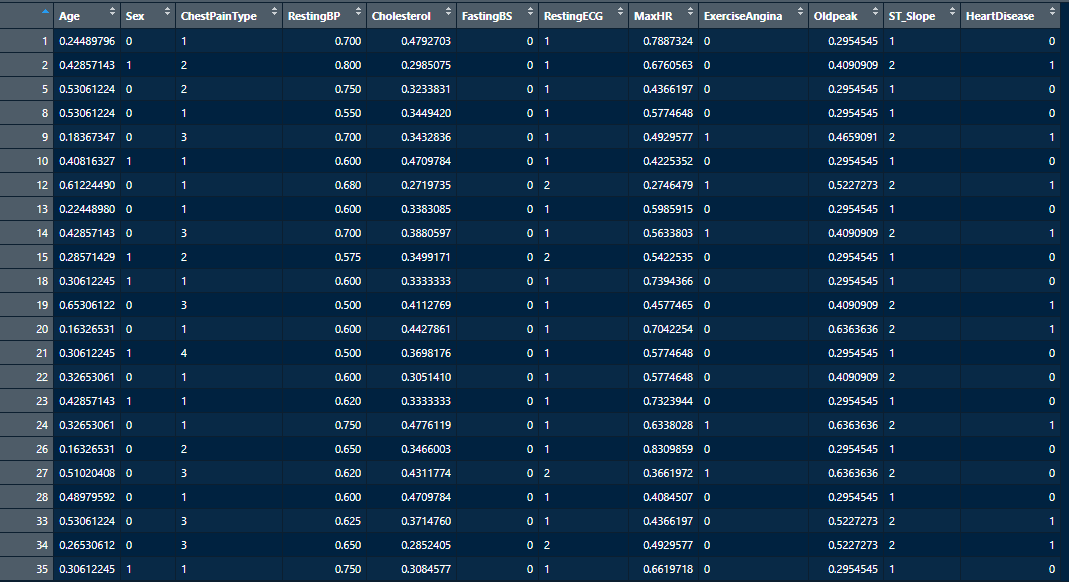
test\_set = subset(dataset, split==FALSE)

split

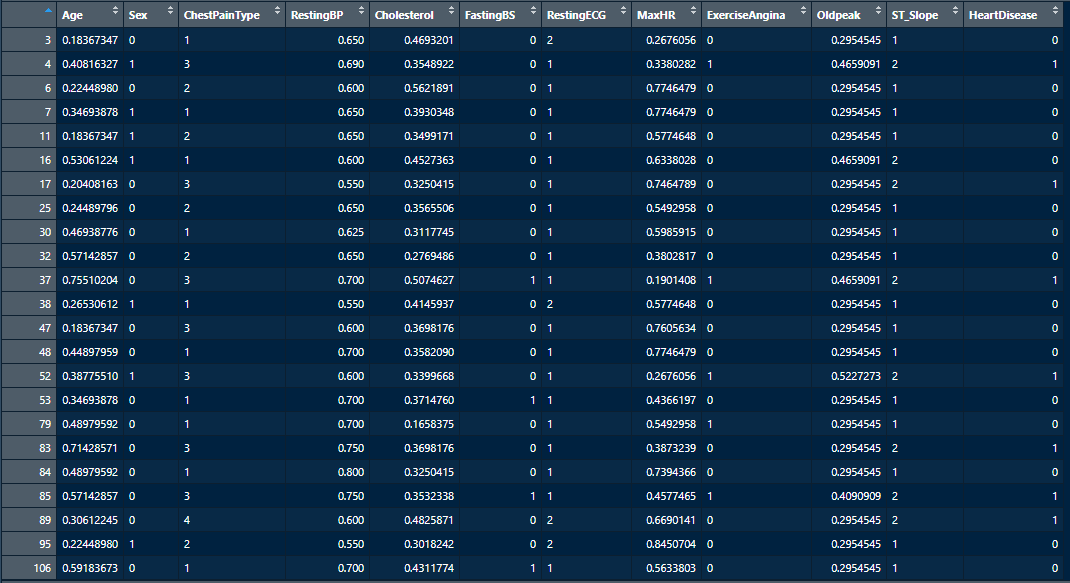
Output:



Training Set:



Test Set:



1. Applying KNN Classifier:

Code:

install.packages("class")

library(class)

y\_pred = knn(train = training\_set[,-12],

test = test\_set[,-12],

cl = training\_set[,12],

k = 7)

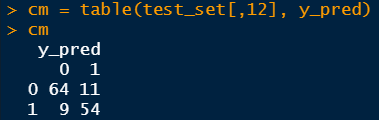
Confusion Matrix:

Code:

cm = table(test\_set[,12], y\_pred)

cm

Output:



Accuracy:

Code:

accuracy = sum(diag(cm))/nrow(test\_set)

accuracy

Output:



**Confusion Matrix:**

|  |  |
| --- | --- |
| Correctly Classified Instances | **118** |
| Incorrectly Classified Instances | **20** |

|  |  |  |
| --- | --- | --- |
| Correct Classification | Classified As | |
| **0** | **1** |
| **0** | **64 (True Negatives)** | **11 (False Positives)** |
| **1** | **9 (False Negatives)** | **54 (True Positives)** |

After, applying KNN with value of K = 7, we can observe that it classifies 118 instances correctly from the available 138 instances of test dataset and classifies 20 instances incorrectly. The model was able to classify 64 instances as 0 (Normal) which were actually 0 (Normal) and classify 11 instances as 1 (heart diseases) which were actually 0 (Normal). The model predicts 9 instances as 0 (Normal) which were actually 1 (heart diseases) and predicts 54 instances as 1 (heart diseases) which were actually 1 (heart diseases).

So, finally the accuracy of the model would be, = 118/138 = .855 \* 100% = 85.5%.