Heart Disease Prediction

Haya Abdeh

9th July 2021

Abstract

This is the second project for the Harvard Data Science Professional Program by Prof. of Biostatistics Rafael Irizarry from Harvard University. In this capstone project, we will choose our own data to make analysis and apply machine learning methods.

Summary

As we will choose our own Data set to analyze, we will make a study about the Heart Disease Data set, we will analyze it, visualize it, and apply machine learning methods to make prediction if an individual will have a heart disease or not.
The data set has been updated about four months ago, and published on kaggle site. the data set can be downloaded from this
link https://www.kaggle.com/rashikrahmanpritom/heart-attack-analysis-prediction-dataset
or from this site
https://archive.ics.uci.edu/ml/datasets/heart+disease
we will work with (heart.csv) file, we will explore the date set and all its variables and analyze the relationship between them and apply machine learning methods to make our predictions.

Introduction

Data science is the gate to introduce machine learning as a technique to describe big data and extract knowledge by applying algorithms that analyze and process data into helpful information and naturally intuitive solutions.

Machine learning methods have become useful in the development of medicine in terms of improving methods for diagnosing diseases and predicting their occurrence by knowing some important data about individuals.
In this study, we will explore the Heart Data set and we are going to predict if an individual will develop heart disease or not by applying machine learning algorithms to make our predictions and to compare between each results to find the appropriate technique which make our prediction more accurate.
We will compute the accuracy of our prediction using these machine learning methods:
Logistic Regression
Regression and Decision Trees

Quadrant Discriminant Analysis (QDA)
Linear Discriminamt Analaysis (LDA)
K-Nearest Neighbours Classifier (KNN)
Support Vector Machine (SVM)
Random Forest (RF)
Gradient Boosting Machine (GBM)

Executive Summary
We start with loading all needed packages and loading the Heart data set (heart.csv) from this link:
https://www.kaggle.com/rashikrahmanpritom/heart-attack-analysis-prediction-dataset
Then we will start to explore our data set and analyze it.
Let us start

```
# Loading all needed libraries


library(dplyr)
library(tidyverse)
library(kableExtra)
library(tidyr)
library(ggplot2)
library(plotly)
library(gbm)
library(caret)
library(xgboost)
library(e1071)
library(class)
library(lightgbm)
library(ROCR)
library(randomForest)
library(PRROC)
library(reshape2)
library(data.table)
library(lubridate)
```

```
library(knitr)
library(recosystem)
library(tinytex)
library(webshot)
library(Hmisc)
library(GGally )
library(rpart)
library(rpart.plot)
```

Exploratory Analysis for Data Set

Introduce The Dataset

First we load the dataset, and show the first 6 rows of it

```
heart_df <- read.csv('/Users/hammar/Documents/RGitProjects/Heart_attack_proj/heart.csv')
heart_df %>% head()
```

```
##   age sex cp trtbps chol fbs restecg thalachh exng oldpeak slp caa thall output
## 1  63   1  3    145  233   1       0      150    0     2.3   0   0     1      1
## 2  37   1  2    130  250   0       1      187    0     3.5   0   0     2      1
## 3  41   0  1    130  204   0       0      172    0     1.4   2   0     2      1
## 4  56   1  1    120  236   0       1      178    0     0.8   2   0     2      1
## 5  57   0  0    120  354   0       1      163    1     0.6   2   0     2      1
## 6  57   1  0    140  192   0       1      148    0     0.4   1   0     1      1
```

The class of the data set is Data frame

```
class(heart_df)
```

```
## [1] "data.frame"
```

Now we show the structure of our Heart dataset, so we can see that it has 303 observations and 14 variables.

```
str(heart_df)
```

```
## 'data.frame':    303 obs. of  14 variables:
##  $ age     : int  63 37 41 56 57 57 56 44 52 57 ...
##  $ sex     : int  1 1 0 1 0 1 0 1 1 1 ...
##  $ cp      : int  3 2 1 1 0 0 1 1 2 2 ...
##  $ trtbps  : int  145 130 130 120 120 140 140 120 172 150 ...
##  $ chol    : int  233 250 204 236 354 192 294 263 199 168 ...
##  $ fbs     : int  1 0 0 0 0 0 0 0 1 0 ...
##  $ restecg : int  0 1 0 1 1 1 0 1 1 1 ...
##  $ thalachh: int  150 187 172 178 163 148 153 173 162 174 ...
##  $ exng    : int  0 0 0 0 1 0 0 0 0 0 ...
##  $ oldpeak : num  2.3 3.5 1.4 0.8 0.6 0.4 1.3 0 0.5 1.6 ...
##  $ slp     : int  0 0 2 2 2 1 1 2 2 2 ...
##  $ caa     : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ thall   : int  1 2 2 2 2 1 2 3 3 2 ...
##  $ output  : int  1 1 1 1 1 1 1 1 1 1 ...
```

Let us explain the meaning of the variable's name:

1.age : displays age of individual

2. sex : Gender of subject: 0 = female 1 = male
3. cp : Chest-pain type for individual, with the following formate:
   0 = typical angina
   1 = atypical angina
   2 = non-angina pain
   3 = asymptomatic angina
4. trtbps : Resting blood pressure value of an individual in mm Hg (unit)

5. chol : displays Serum cholesterol in mg/dl (unit)
6. fbs : Fasting blood sugar of an individual ,level relative to 120 mg/dl: 0 = fasting blood sugar <= 120 mg/dl And 1 = fasting blood sugar > 120 mg/dl
7. restecg - Resting ECG: Resting electrocardiographic results
   0 = normal
   1 = ST-T wave abnormality
   2 = left ventricle hyperthrophy
8. thalachh : Maximum heart rate of an individual
9. exng : Exercise Induced Angina, 0 = no 1 = yes
10. oldpeak : previous peack - ST Depression Induced by Exercise Relative to Rest, value is integer or float
11. slp - slope - Peak Exercise ST Segment:
    1 = Up-sloaping
    2 = flat
    3 = downsloping
12. caa : Number of major vessels (0-3) colored by flourosopy, displays value as integer.
13. thall : displays thalassemia :
    0 = normal
    1 = silent carrier but normal
    2 = fixed defect
    3 = reversable defect
14. output : Diagnosis of heart disease which Displays whether the individual is suffering from heart disease or not :
    0 = absence
    1 = present.

Let us rename the columns to more meaningful names

```
#defining meanningful names
```

```
names <- c("Age",
           "Sex",
           "Chest_Pain_Type",
           "Resting_Blood_Pressure",
           "Cholesterol_serum",
           "Fasting_Blood_Sugar",
           "Resting_ECG",
           "Maximum_Heart_Rate",
           "Exercise_Induced_Angina",
           "ST_Depression_Exercise",
           "Peak_Exercise_ST_Segment",
           "Num_Major_Vessels_Flourosopy",
           "Thalassemia",
           "Diagnosis_Heart_Disease")

#Lets keep the old names in another data frame
heart_df_oldnames <- heart_df

#now rename the columns of the dataframe

colnames(heart_df) <- names

#show the new names
names(heart_df)
```

```
##  [1] "Age"                                "Sex"
##  [3] "Chest_Pain_Type"                    "Resting_Blood_Press
ure"
##  [5] "Cholesterol_serum"                  "Fasting_Blood_Sugar
"
##  [7] "Resting_ECG"                        "Maximum_Heart_Rate"
##  [9] "Exercise_Induced_Angina"            "ST_Depression_Exerc
ise"
## [11] "Peak_Exercise_ST_Segment"           "Num_Major_Vessels_F
lourosopy"
## [13] "Thalassemia"                        "Diagnosis_Heart_Dis
ease"
```

Data Visualization

```
#LEt us show the Data frame summary
summary(heart_df)
```

```
##       Age              Sex         Chest_Pain_Type Restin
g_Blood_Pressure
##  Min.   :29.00   Min.   :0.0000   Min.   :0.000   Min.
: 94.0
##  1st Qu.:47.50   1st Qu.:0.0000   1st Qu.:0.000   1st Qu
.:120.0
##  Median :55.00   Median :1.0000   Median :1.000   Median
:130.0
##  Mean   :54.37   Mean   :0.6832   Mean   :0.967   Mean
:131.6
##  3rd Qu.:61.00   3rd Qu.:1.0000   3rd Qu.:2.000   3rd Qu
.:140.0
##  Max.   :77.00   Max.   :1.0000   Max.   :3.000   Max.
:200.0
##  Cholesterol_serum Fasting_Blood_Sugar  Resting_ECG
Maximum_Heart_Rate
##  Min.   :126.0     Min.   :0.0000      Min.   :0.0000
Min.   : 71.0
##  1st Qu.:211.0     1st Qu.:0.0000      1st Qu.:0.0000
```

```
 1st Qu.:133.5
##  Median :240.0     Median :0.0000      Median :1.0000
Median :153.0
##  Mean   :246.3     Mean   :0.1485      Mean   :0.5281
Mean   :149.6
##  3rd Qu.:274.5     3rd Qu.:0.0000      3rd Qu.:1.0000
3rd Qu.:166.0
##  Max.   :564.0     Max.   :1.0000      Max.   :2.0000
Max.   :202.0
##  Exercise_Induced_Angina ST_Depression_Exercise Peak_Exe
rcise_ST_Segment
##  Min.   :0.0000          Min.   :0.00           Min.   :
0.000
##  1st Qu.:0.0000          1st Qu.:0.00           1st Qu.:
1.000
##  Median :0.0000          Median :0.80           Median :
1.000
##  Mean   :0.3267          Mean   :1.04           Mean   :
1.399
##  3rd Qu.:1.0000          3rd Qu.:1.60           3rd Qu.:
2.000
##  Max.   :1.0000          Max.   :6.20           Max.   :
2.000
##  Num_Major_Vessels_Flourosopy  Thalassemia    Diagnosis_
Heart_Disease
##  Min.   :0.0000               Min.   :0.000   Min.   :0.
0000
##  1st Qu.:0.0000               1st Qu.:2.000   1st Qu.:0.
0000
##  Median :0.0000               Median :2.000   Median :1.
0000
##  Mean   :0.7294               Mean   :2.314   Mean   :0.
5446
##  3rd Qu.:1.0000               3rd Qu.:3.000   3rd Qu.:1.
0000
```

```
##  Max.   :4.0000                Max.   :3.000   Max.   :1.
0000
```

Let’s calculate the distinct values and types for all 14 variables

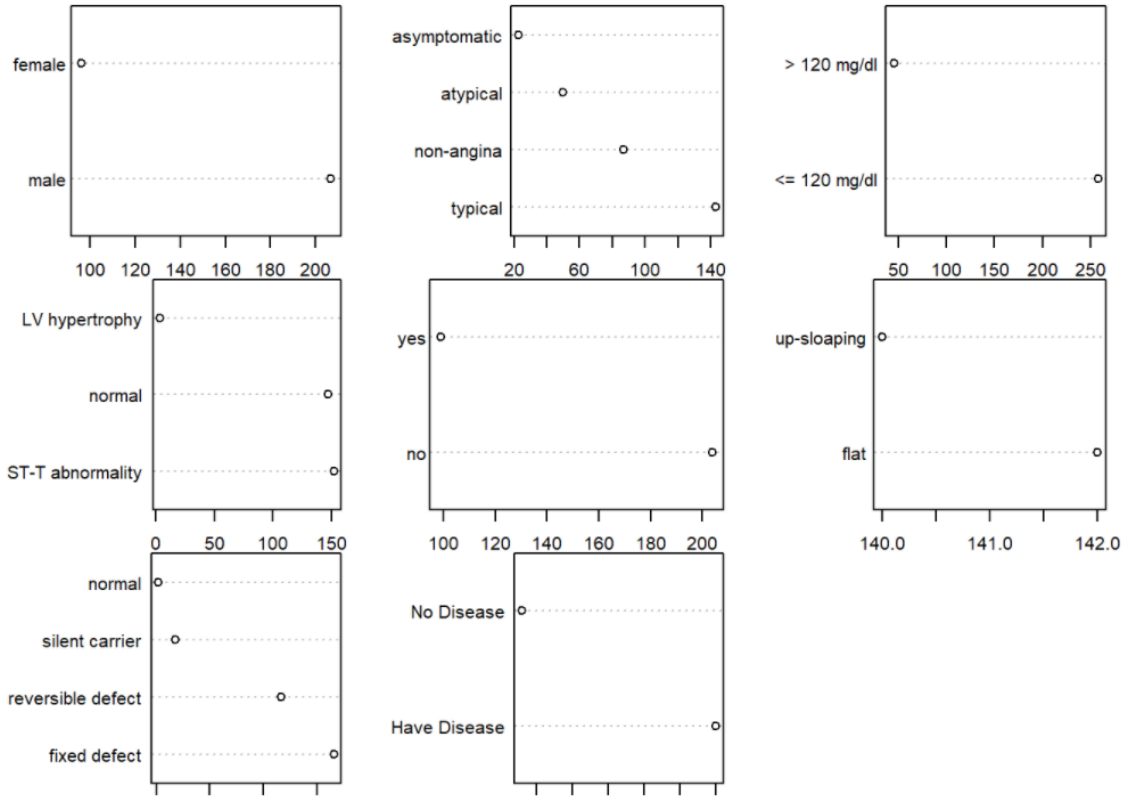
```
heart_df %>%
summarise(age_ranges = n_distinct(Age),
          sex_types = n_distinct(Sex),
          cp_types = n_distinct(Chest_Pain_Type),
          nom_trestbps = n_distinct(Resting_Blood_Pressure)
,
          nom_chol = n_distinct(Cholesterol_serum),
          nom_fbs = n_distinct(Fasting_Blood_Sugar),
          types_restecg = n_distinct(Resting_ECG),
          nom_thalach = n_distinct(Maximum_Heart_Rate),
          nom_exang = n_distinct(Exercise_Induced_Angina),
          nom_oldpeak = n_distinct(ST_Depression_Exercise),
          types_slope = n_distinct(Peak_Exercise_ST_Segment
),
          nom_caa = n_distinct(Num_Major_Vessels_Flourosopy
),
          types_thal = n_distinct(Thalassemia),
Diagnosis_types = n_distinct(Diagnosis_Heart_Disease))
```

```
##   age_ranges sex_types cp_types nom_trestbps nom_chol no
m_fbs types_restecg
## 1         41         2        4           49      152
2             3
##   nom_thalach nom_exang nom_oldpeak types_slope nom_caa
types_thal
## 1          91         2          40           3       5
4
##   Diagnosis_types
## 1               2
```

Let us visualize the categorical variables in the Heart Dataset (Sex , Chest_Pain_Type, Fasting_Blood_Sugar , Resting_ECG, Exercise_Induced_Angina , Peak_Exercise_ST_Segment , Thalassemia , Diagnosis_Heart_Disease)

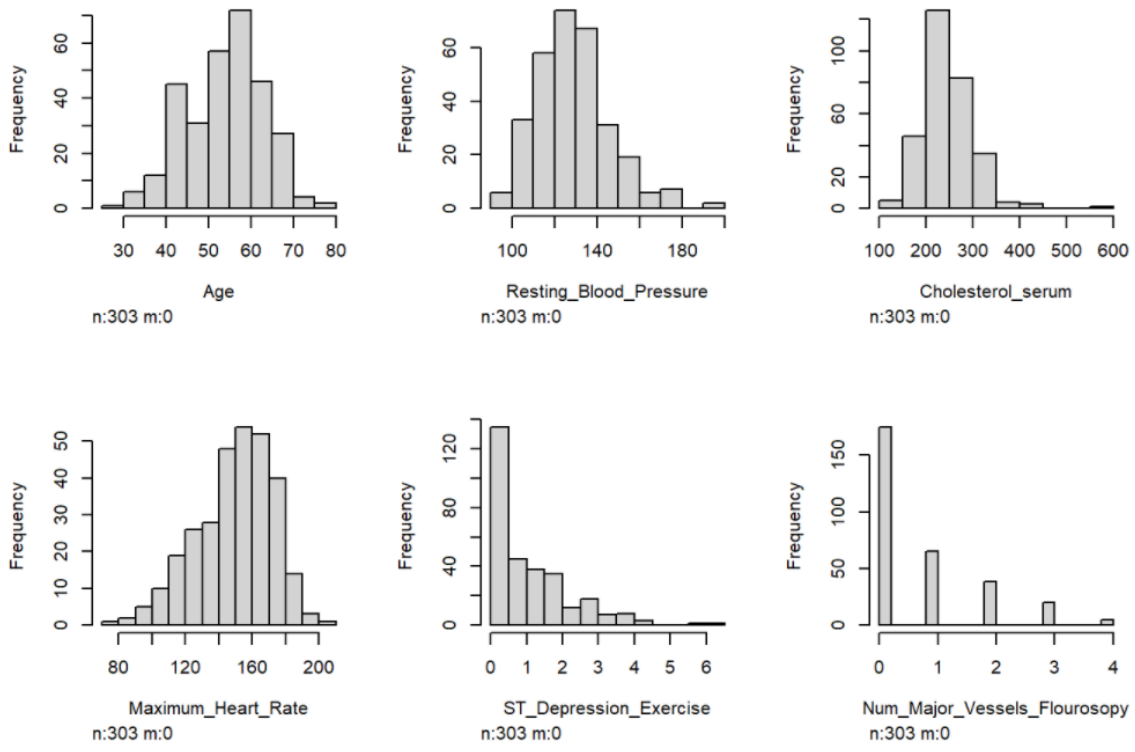
```
# Histogram for all  Categorical Variables in The Heart Dat
aset each column individually.



heart_df_cat1 <- heart_df %>%
  select(Sex , Chest_Pain_Type, Fasting_Blood_Sugar ,
         Resting_ECG, Exercise_Induced_Angina ,
         Peak_Exercise_ST_Segment , Thalassemia ,
         Diagnosis_Heart_Disease)%>%
         mutate(Sex = recode_factor(Sex, `0` = "female", `1
` = "male" ),
         Chest_Pain_Type = recode_factor(Chest_Pain_Type,`0
` = "typical",
                                                          `1
` = "atypical",
                                                          `2
` = "non-angina",
                                                          `3
`="asymptomatic"),
         Fasting_Blood_Sugar =
         recode_factor(Fasting_Blood_Sugar,`0`="<= 120 mg/d
l",
                                             `1`="> 120 mg/dl
"),
          Resting_ECG = recode_factor(Resting_ECG, `0` = "n
ormal",
                                                      `1` = "S
T-T abnormality",
                                                      `2` = "L
V hypertrophy"),
          Exercise_Induced_Angina = recode_factor(Exercise_
Induced_Angina,`0`="no",`1` = "yes"),
```

```
            Peak_Exercise_ST_Segment = recode_factor(Peak_Ex
ercise_ST_Segment,`1`= "up-sloaping",
                                          `2` = "flat",
                                          `3` = "down-sloaping",
    Thalassemia = recode_factor(Thalassemia,`0` = "normal",
                                        `1` = "silent carrier",
                                        `2` = "fixed defect",
                                      `3` = "reversible defect"),
Diagnosis_Heart_Disease = recode_factor(Diagnosis_Heart_Dis
ease, `0` = "No Disease",`1` = "Have Disease") )


par(mar=c(1, 1, 1, 1))

hist.data.frame(heart_df_cat1)
```



And now visualize the numiric variables in the Heart Dataset (Age,Resting_Blood_Pressure,Cholesterol_serum, Maximum_Heart_Rate,ST_Depression_Exercise,Num_Major_Vessels_Flouroso py)

```
# Histogram for all numeric Variables in The Heart Dataset each column individually.

heart_df_num <- heart_df %>% select(Age,Resting_Blood_Press ure,Cholesterol_serum,

        Maximum_Heart_Rate, ST_Depression_Exercise, Num_Maj or_Vessels_Flourosopy)


hist.data.frame(heart_df_num)
```



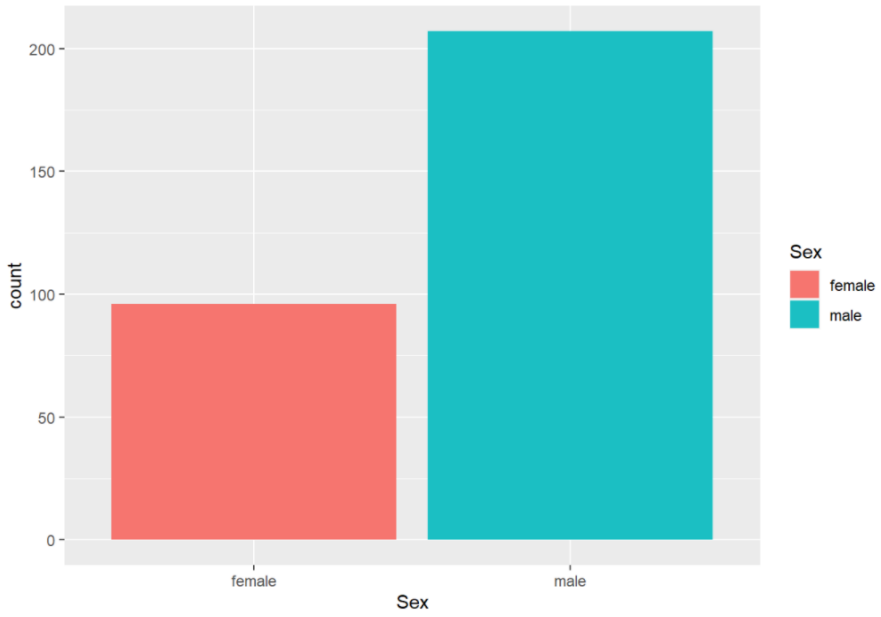
When visualizing the previous plots we can have an idea of the high rates of variables may cause a heart disease, may this can give us an idea of the relations between these variables.
Highly correlated variables can lead to overly complicated models or wonky predictions. we will find the correlations between the variables after we moved in our analysis.

Lets visualize more in our variables.
Let us compute each gender, and plot it.

```
heart_df %>%
  drop_na() %>%
  group_by(Sex) %>%
  count() %>%
  ungroup()
```

```
## # A tibble: 2 x 2
##     Sex     n
##   <int> <int>
## 1     0    96
## 2     1   207
```

```
heart_df_gender <- heart_df %>%
    select(Sex)%>%
    mutate(Sex = recode_factor(Sex, `0` = "female", `1` = "male" )
    )
ggplot(heart_df_gender) +
                  geom_bar(aes(x = Sex ,fill=Sex ) )
```

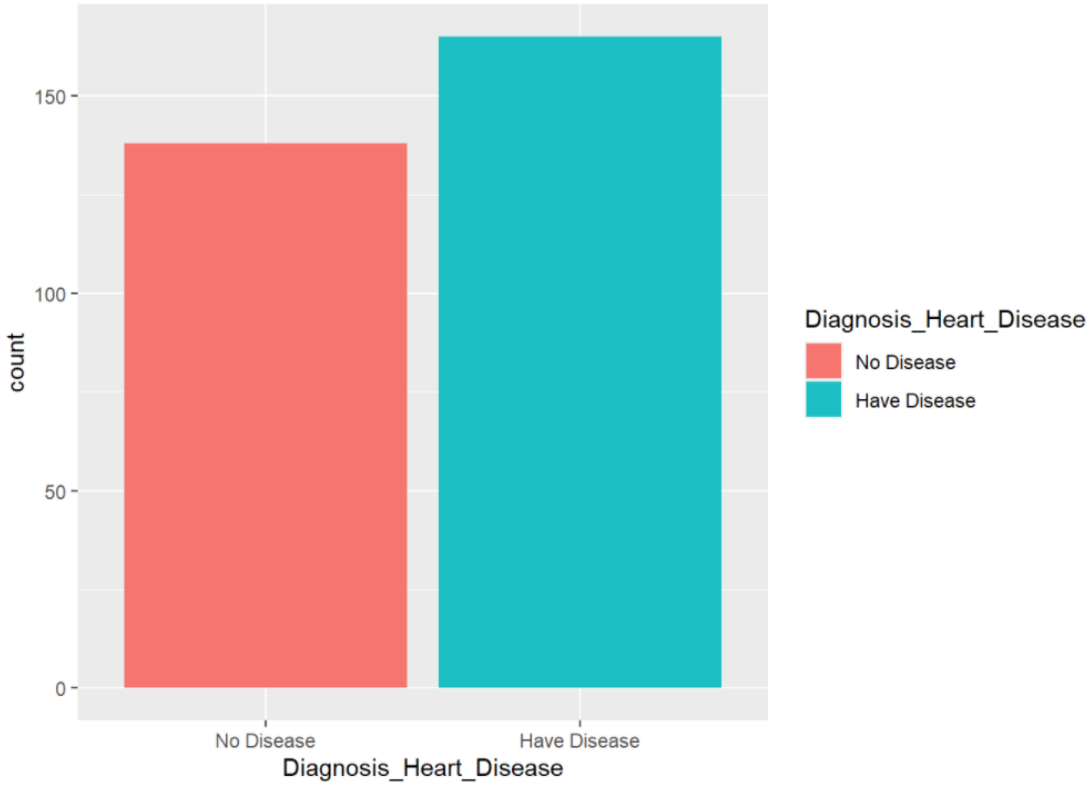


Now let us calculate how many individuals that could be Diagnosed to suffer from heart disease

```
heart_df %>%
  drop_na() %>%
  group_by(Diagnosis_Heart_Disease) %>%
  count() %>%
  ungroup()
```

```
## # A tibble: 2 x 2
##   Diagnosis_Heart_Disease     n
##                     <int> <int>
## 1                       0   138
## 2                       1   165
```

```
heart_df_diseased <- heart_df %>%
    select(Diagnosis_Heart_Disease)%>%
    mutate(Diagnosis_Heart_Disease = recode_factor(Diagnosis_Heart_Disease, `0` = "No Disease", `1` = "Have Disease" )
    )


ggplot(heart_df_diseased) +
                   geom_bar(aes(x = Diagnosis_Heart_Disease,fill=Diagnosis_Heart_Disease ) )
```

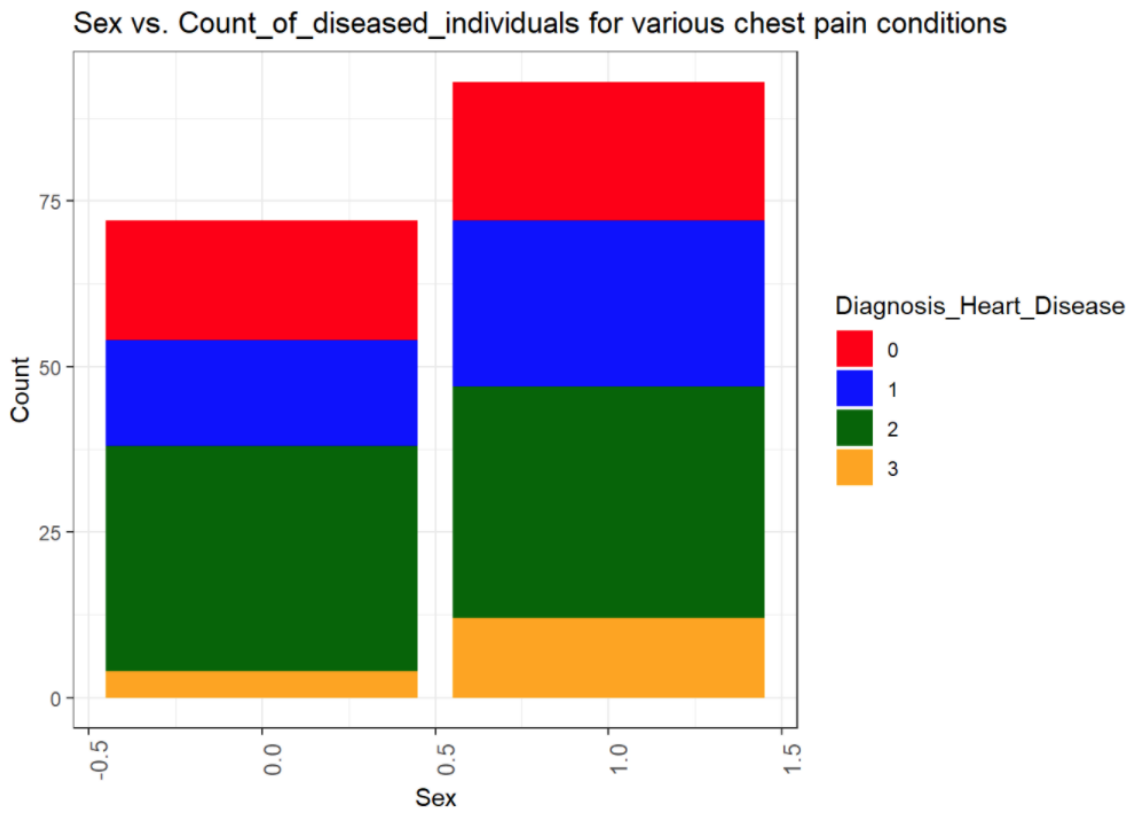


As we see the results the total count of having heart disease (1 = present of disease) is 165 which is higher than not having a heart disease (0 = absence) 138.

Now let us show Diagnosis Heart Disease by gender for all types of chest pain

```
heart_df %>% filter(Diagnosis_Heart_Disease == 1) %>% group
_by(Sex, Chest_Pain_Type) %>% summarise(count = n()) %>%

  ggplot() + geom_bar(aes(Sex, count,   fill = as.factor(Ch
est_Pain_Type)),stat = "Identity") +

  theme_bw() +

  theme(axis.text.x = element_text(angle = 90, size = 10))
+

  ylab("Count") + xlab("Sex") + labs(fill = "Diagnosis_Hear
t_Disease") +

  ggtitle("Sex vs. Count_of_diseased_individuals for variou
s chest pain conditions") +
```

```
  scale_fill_manual(values=c("red", "blue", "darkgreen", "o
range"))
```



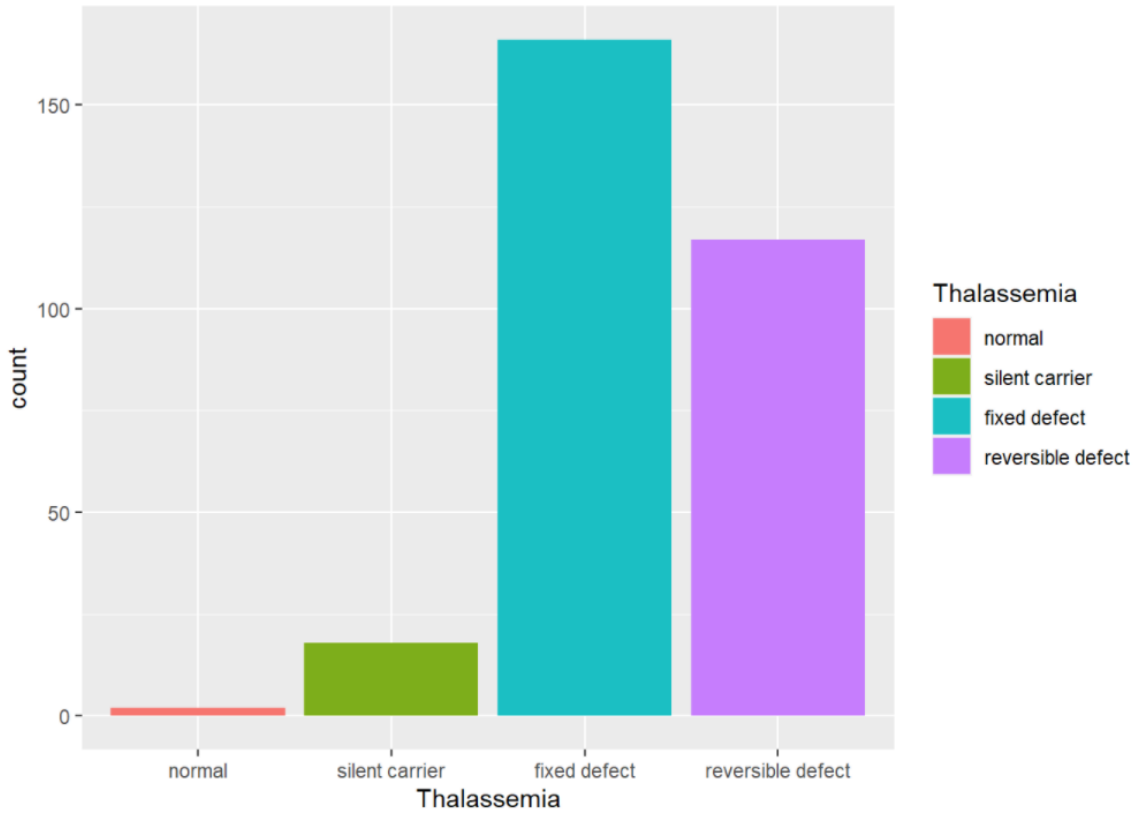
Individuals having Thalassemia may have a higher chance of having heart disease , if this is true lets make visualization on this to know.

Show levels of Thalassemia

```
heart_df %>%
  drop_na() %>%
  group_by(Thalassemia) %>%
  count() %>%
  ungroup()
```

```
## # A tibble: 4 x 2
##   Thalassemia     n
##         <int> <int>
## 1           0     2
## 2           1    18
## 3           2   166
## 4           3   117
```

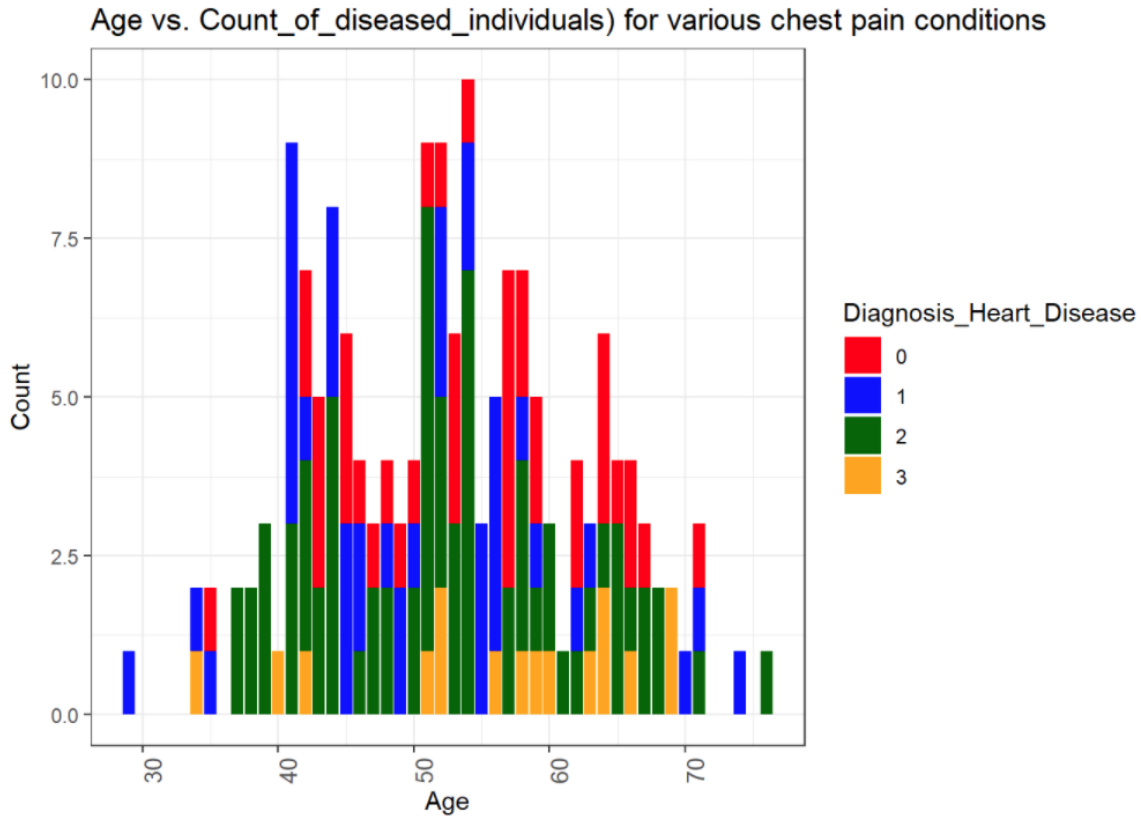
```
heart_df_Thalassemia <- heart_df %>%
    select(Thalassemia)%>%
    mutate(Thalassemia =recode_factor(Thalassemia,`0` = "normal",
                                                          `1` = "silent carrier",
                                                          `2` = "fixed defect",
                                                          `3` = "reversible defect"))


ggplot(heart_df_Thalassemia) +
                    geom_bar(aes(x = Thalassemia ,fill=Thalassemia ) )
```



Let us show a diagram explains Chest pain type for diseased people with Age classifications

```
heart_df %>% filter(Diagnosis_Heart_Disease == 1) %>% group
_by(Age, Chest_Pain_Type) %>% summarise(count = n()) %>%

  ggplot() + geom_bar(aes(Age, count,   fill = as.factor(Ch
est_Pain_Type)),stat = "Identity") +

  theme_bw() +

  theme(axis.text.x = element_text(angle = 90, size = 10))
+
```

```
  ylab("Count") + xlab("Age") + labs(fill = "Diagnosis_Heart_Disease") +
  ggtitle("Age vs. Count_of_diseased_individuals) for various chest pain conditions") +
  scale_fill_manual(values=c("red", "blue", "darkgreen", "orange"))
```



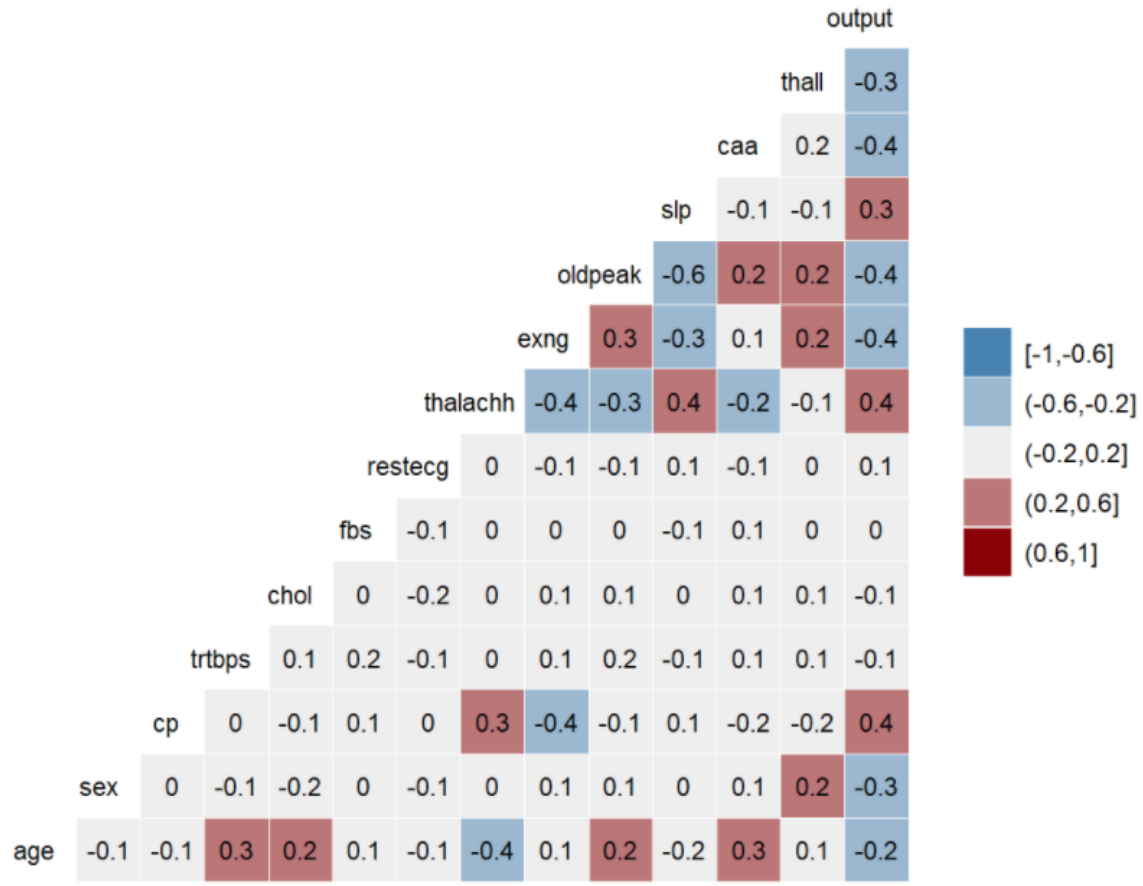
As we mintioned before we have 4 types of Chest pain
cp : Chest-pain type for individual, with the following formate:
type 0 = typical angina
type 1 = atypical angina
type 2 = non-angina pain
type 3 = asymptomatic angina
We can see - Majority of individuals has the type-2 of Chest_Pain (non-angina pain) with ages about (36-75)

Methods of Machine Learning

Variables that are highly correlated could give us correct predictions or incorrect predictions,we are going to start with finding the correlated variables so we can have a high prediction.

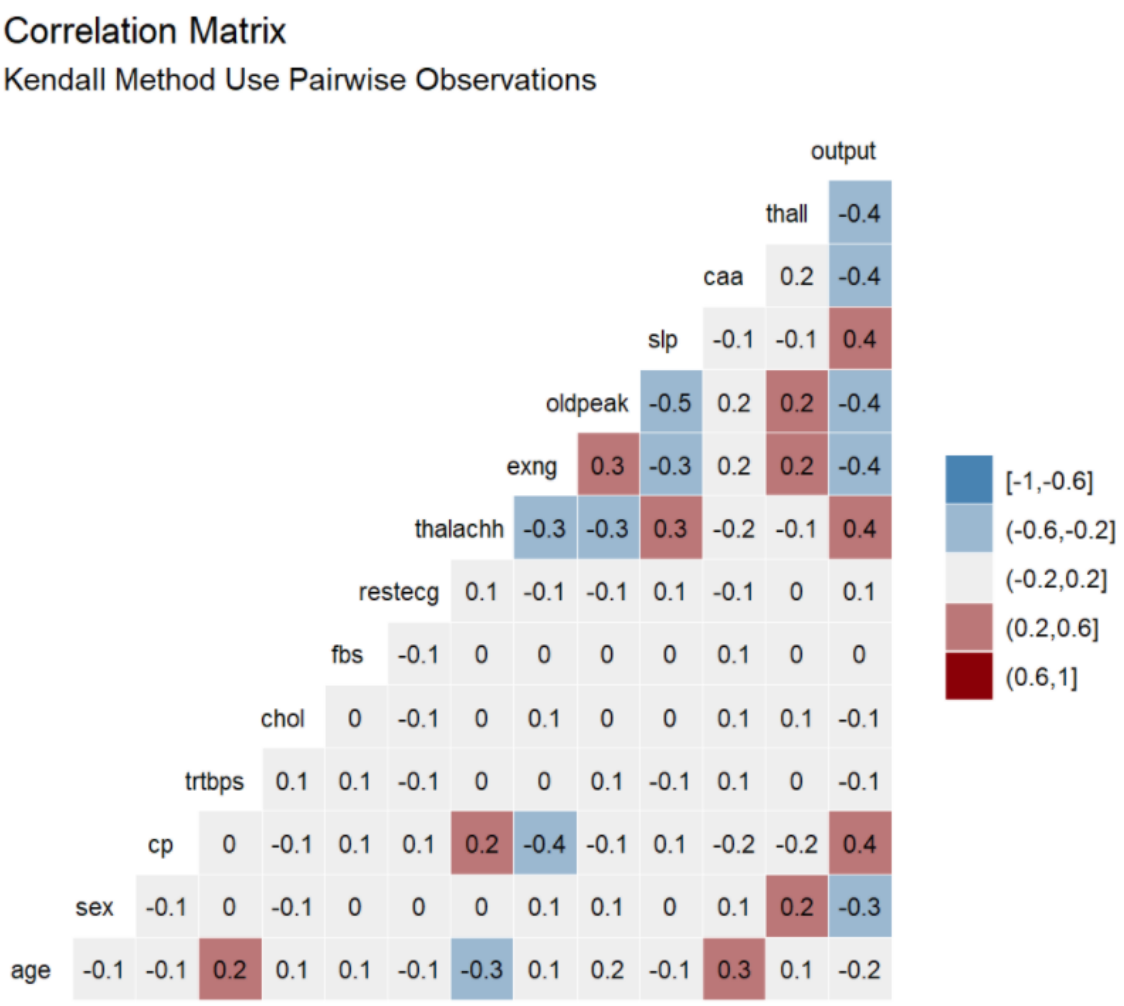
Let us use function from GGally library which is ggcorr() to make a correlation matrix of the numeric variables, we have two methods to apply, the first one is Pearson which is not that ideal method if the data has too much outliers, the second method is Kendall, which is more suitable for our data.
Let us check both of them.

```
# to use the short names of the data frame, we use the data
frame that we initiate previously with shortcut names.


#Correlation matrix using Pearson method, default method is
Pearson
heart_df_oldnames %>% ggcorr(high       = "darkred",
                             low        = "steelblue",
                             label      = TRUE,
                             hjust      = .75,
                             size       = 3,
                             label_size = 3,
                             nbreaks    = 5
                                          ) +
  labs(title = "Correlation Matrix",
  subtitle = "Pearson Method Use Pairwise Obervations")
```




```
#Correlation matrix using Kendall method
heart_df_oldnames %>% ggcorr(method     = c("pairwise", "ke
ndall"),
                             high       = "darkred",
                             low        = "steelblue",
                             label      = TRUE,
                             hjust      = .75,
                             size       = 3,
                             label_size = 3,
                             nbreaks    = 5
                                   ) +
```

```
  labs(title = "Correlation Matrix",
  subtitle = "Kendall Method Use Pairwise Observations")
```



There are a slight differences between the Pearson and Kendall results, the variables are not highly correlated.

Machine Learning Basics

We will apply machine learning methods to compute the accuracy of our prediction using these machine learning methods:
Logistic Regression
Regression and Decision Trees
Quadrant Discriminant Analysis (QDA)
Linear Discriminamt Analaysis (LDA)
K-Nearest Neighbours Classifier (KNN)
Support Vector Machine (SVM)
Random Forest (RF)
Gradient Boosting Machine (GBM)

Before we start with the algorithms, we will split our dataset to training and test set, to compare our results.

Training and Test sets, and overall accuracy

```
#create Data Partition

#set seed for reproducible results

set.seed(1)


test_index <- createDataPartition(y = heart_df$Diagnosis_He
art_Disease, times = 1, p = 0.1, list = FALSE)

train_heart_df <- heart_df[-test_index, ]

test_heart_df <- heart_df[test_index, ]
```

After we partition our dataset lets check the dimension of each training and test set

```
#dimension of the training data set


dim(train_heart_df)
```

```
## [1] 272  14
```

```
#dimension of the test data set


dim(test_heart_df)
```

```
## [1] 31 14
```

Applying Methods of Machine Learning

It is convenient to start working with Logistic regression model since it is relatively easy to implement and yields results that have intuitive meaning.

Logistic Regression Logistic regression is a statistical model that in its basic form uses a logistic function to model a binary dependent variable, Mathematically, a binary logistic model has a dependent variable with two possible values, such as pass/fail which is represented by an indicator variable, where the two values are labeled “0” and “1”.

```
#Logistic Regression model

set.seed(1)


    log_regr_hd_model = glm(Diagnosis_Heart_Disease~., data
=train_heart_df, family='binomial')


 summary(log_regr_hd_model)
```

```
##
## Call:
## glm(formula = Diagnosis_Heart_Disease ~ ., family = "bin
omial",
##     data = train_heart_df)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.5633  -0.3960   0.1469   0.5876   2.5025
##
## Coefficients:
##                              Estimate Std. Error z valu
e Pr(>|z|)
## (Intercept)                  3.973030   2.775290   1.43
2  0.15227
## Age                         -0.009467   0.024845  -0.38
1  0.70317
```

```
## Sex                            -1.577591   0.482908  -3.26
7  0.00109 **
## Chest_Pain_Type                 0.804998   0.195659   4.11
4 3.88e-05 ***
## Resting_Blood_Pressure         -0.020459   0.011019  -1.85
7  0.06334 .
## Cholesterol_serum              -0.004707   0.003816  -1.23
3  0.21745
## Fasting_Blood_Sugar             0.222651   0.547886   0.40
6  0.68446
## Resting_ECG                     0.588453   0.361832   1.62
6  0.10388
## Maximum_Heart_Rate              0.021039   0.011166   1.88
4  0.05952 .
## Exercise_Induced_Angina        -1.108258   0.441805  -2.50
8  0.01213 *
## ST_Depression_Exercise         -0.591779   0.224294  -2.63
8  0.00833 **
## Peak_Exercise_ST_Segment        0.329832   0.384015   0.85
9  0.39039
## Num_Major_Vessels_Flourosopy -0.811153   0.205034  -3.95
6 7.62e-05 ***
## Thalassemia                    -0.703026   0.303807  -2.31
4  0.02066 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1
' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 375.88  on 271  degrees of freedom
## Residual deviance: 194.68  on 258  degrees of freedom
## AIC: 222.68


##
```

```
## Number of Fisher Scoring iterations: 6
```

Some variables are not significant. So let us check the Multi collinearity

```
cor(train_heart_df)
```

```
##                                    Age          Sex Che
st_Pain_Type
## Age                         1.00000000 -0.05870629
-0.06804040
## Sex                        -0.05870629  1.00000000
-0.04377826
## Chest_Pain_Type            -0.06804040 -0.04377826
1.00000000
## Resting_Blood_Pressure      0.28396173 -0.04375402
0.04203868
## Cholesterol_serum           0.20658040 -0.18134287
-0.06816631
## Fasting_Blood_Sugar         0.13143747  0.03045931
0.11803110
## Resting_ECG                -0.13568375 -0.05752715
0.04857083
## Maximum_Heart_Rate         -0.39893973 -0.07514824
0.28072394
## Exercise_Induced_Angina     0.07380707  0.13371352
-0.41976875
## ST_Depression_Exercise      0.21859497  0.08746082
-0.15593697
## Peak_Exercise_ST_Segment   -0.19643157 -0.01481500
0.09172991
## Num_Major_Vessels_Flourosopy  0.28647423  0.10479789
-0.19415542
## Thalassemia                 0.08699454  0.19848152
-0.14624579
## Diagnosis_Heart_Disease    -0.24448346 -0.26153760
```

```
   0.42576193
##                                Resting_Blood_Pressure Chol
esterol_serum
## Age                                        0.28396173
0.206580398
## Sex                                       -0.04375402
-0.181342874
## Chest_Pain_Type                            0.04203868
-0.068166309
## Resting_Blood_Pressure                     1.00000000
0.100325028
## Cholesterol_serum                          0.10032503
1.000000000
## Fasting_Blood_Sugar                        0.20139639
0.004711190
## Resting_ECG                               -0.14443945
-0.148402911
## Maximum_Heart_Rate                        -0.05858395
0.015706402
## Exercise_Induced_Angina                    0.06103037
0.065650539
## ST_Depression_Exercise                     0.22734189
0.058825454
## Peak_Exercise_ST_Segment                  -0.17508757
-0.009920242
## Num_Major_Vessels_Flourosopy               0.10094911
0.089192150
## Thalassemia                                0.09198073
0.116601563
## Diagnosis_Heart_Disease                   -0.16998535
-0.096735010
##                                Fasting_Blood_Sugar Resting
_ECG Maximum_Heart_Rate
## Age                                     0.13143747 -0.1356
8375        -0.39893973
## Sex                                     0.03045931 -0.0575
```

```
2715        -0.07514824

## Chest_Pain_Type                          0.11803110  0.0485
7083         0.28072394

## Resting_Blood_Pressure                   0.20139639 -0.1444
3945        -0.05858395

## Cholesterol_serum                        0.00471119 -0.1484
0291         0.01570640

## Fasting_Blood_Sugar                      1.00000000 -0.0992
3581        -0.02722403

## Resting_ECG                             -0.09923581  1.0000
0000         0.05454858

## Maximum_Heart_Rate                      -0.02722403  0.0545
4858         1.00000000

## Exercise_Induced_Angina                  0.03470234 -0.0713
1224        -0.37732003

## ST_Depression_Exercise                   0.01727857 -0.0614
4549        -0.33844962

## Peak_Exercise_ST_Segment                -0.05718833  0.0891
5277         0.37815782

## Num_Major_Vessels_Flourosopy             0.16028345 -0.1091
3750        -0.20717269

## Thalassemia                             -0.04421980 -0.0251
4528        -0.07541872

## Diagnosis_Heart_Disease                 -0.01764299  0.1724
8328         0.40938532

##                               Exercise_Induced_Angina ST_
Depression_Exercise

## Age                                          0.07380707
0.21859497

## Sex                                          0.13371352
0.08746082

## Chest_Pain_Type                             -0.41976875
-0.15593697

## Resting_Blood_Pressure                       0.06103037

           0.22734189

## Cholesterol_serum                            0.06565054
```

```
            0.05882545
## Fasting_Blood_Sugar                          0.03470234
0.01727857
## Resting_ECG                                 -0.07131224
-0.06144549
## Maximum_Heart_Rate                          -0.37732003
-0.33844962
## Exercise_Induced_Angina                      1.00000000
0.29186460
## ST_Depression_Exercise                       0.29186460
1.00000000
## Peak_Exercise_ST_Segment                    -0.27544795
-0.59614158
## Num_Major_Vessels_Flourosopy                 0.11224900
0.22619379
## Thalassemia                                  0.20921619
0.19757932
## Diagnosis_Heart_Disease                     -0.44675167
-0.43356195
##                               Peak_Exercise_ST_Segment
## Age                                      -0.196431565
## Sex                                      -0.014814999
## Chest_Pain_Type                           0.091729907
## Resting_Blood_Pressure                   -0.175087567
## Cholesterol_serum                        -0.009920242
## Fasting_Blood_Sugar                      -0.057188326
## Resting_ECG                               0.089152766
## Maximum_Heart_Rate                        0.378157822
## Exercise_Induced_Angina                  -0.275447954
## ST_Depression_Exercise                   -0.596141583
## Peak_Exercise_ST_Segment                  1.000000000
## Num_Major_Vessels_Flourosopy             -0.088905519
## Thalassemia                              -0.083536262
```

```
## Diagnosis_Heart_Disease                  0.321552675
##                                Num_Major_Vessels_Flourosop
y Thalassemia
## Age                                          0.2864742
3  0.08699454
## Sex                                          0.1047978
9  0.19848152
## Chest_Pain_Type                             -0.1941554
2 -0.14624579
## Resting_Blood_Pressure                       0.1009491
1  0.09198073
## Cholesterol_serum                            0.0891921
5  0.11660156
## Fasting_Blood_Sugar                          0.1602834
5 -0.04421980
## Resting_ECG                                 -0.1091375
0 -0.02514528
## Maximum_Heart_Rate                          -0.2071726
9 -0.07541872
## Exercise_Induced_Angina                      0.1122490
0  0.20921619
## ST_Depression_Exercise                       0.2261937
9  0.19757932
## Peak_Exercise_ST_Segment                    -0.0889055
2 -0.08353626
## Num_Major_Vessels_Flourosopy                 1.0000000
0  0.17683452
## Thalassemia                                  0.1768345
2  1.00000000
## Diagnosis_Heart_Disease                     -0.4032770
5 -0.31905104


##                                Diagnosis_Heart_Disease
## Age                                        -0.24448346
## Sex                                        -0.26153760
```

```
## Chest_Pain_Type                          0.42576193

## Resting_Blood_Pressure                  -0.16998535
## Cholesterol_serum                       -0.09673501
## Fasting_Blood_Sugar                     -0.01764299
## Resting_ECG                              0.17248328
## Maximum_Heart_Rate                       0.40938532
## Exercise_Induced_Angina                 -0.44675167
## ST_Depression_Exercise                  -0.43356195
## Peak_Exercise_ST_Segment                 0.32155268
## Num_Major_Vessels_Flourosopy            -0.40327705
## Thalassemia                             -0.31905104
## Diagnosis_Heart_Disease                  1.00000000
```

Since it's hard to find which variables are highly correlated. We will see only the variables with correlation > 0.7 or < -0.7

```
abs(cor(train_heart_df))>0.7
```

```
##                                Age   Sex Chest_Pain_Type
Resting_Blood_Pressure
## Age                           TRUE FALSE           FALSE
FALSE
## Sex                          FALSE  TRUE           FALSE
FALSE
## Chest_Pain_Type              FALSE FALSE            TRUE
               FALSE
## Resting_Blood_Pressure       FALSE FALSE           FALSE
TRUE
## Cholesterol_serum            FALSE FALSE           FALSE
FALSE
## Fasting_Blood_Sugar          FALSE FALSE           FALSE
FALSE
```

```
## Resting_ECG                   FALSE FALSE           FALSE
                FALSE
## Maximum_Heart_Rate            FALSE FALSE           FALSE
FALSE
## Exercise_Induced_Angina       FALSE FALSE           FALSE
FALSE
## ST_Depression_Exercise        FALSE FALSE           FALSE
FALSE
## Peak_Exercise_ST_Segment      FALSE FALSE           FALSE
FALSE
## Num_Major_Vessels_Flourosopy FALSE FALSE           FALSE
FALSE
## Thalassemia                   FALSE FALSE           FALSE
FALSE
## Diagnosis_Heart_Disease       FALSE FALSE           FALSE
FALSE
##                               Cholesterol_serum Fasting_B
lood_Sugar Resting_ECG
## Age                                        FALSE
FALSE       FALSE
## Sex                                        FALSE
FALSE       FALSE
## Chest_Pain_Type                            FALSE
FALSE       FALSE
## Resting_Blood_Pressure                     FALSE
FALSE       FALSE
## Cholesterol_serum                           TRUE
   FALSE       FALSE
## Fasting_Blood_Sugar                        FALSE
TRUE       FALSE
## Resting_ECG                                FALSE
FALSE        TRUE
## Maximum_Heart_Rate                         FALSE
FALSE       FALSE
## Exercise_Induced_Angina                    FALSE
           FALSE       FALSE
```

```
## ST_Depression_Exercise                     FALSE
FALSE       FALSE

## Peak_Exercise_ST_Segment                   FALSE
FALSE       FALSE

## Num_Major_Vessels_Flourosopy               FALSE
FALSE       FALSE

## Thalassemia                                FALSE
FALSE       FALSE

## Diagnosis_Heart_Disease                    FALSE
FALSE       FALSE

##                                Maximum_Heart_Rate Exercise
_Induced_Angina

## Age                                         FALSE
FALSE

## Sex                                         FALSE
FALSE

## Chest_Pain_Type                             FALSE
FALSE

## Resting_Blood_Pressure                      FALSE
FALSE

## Cholesterol_serum                           FALSE
FALSE

## Fasting_Blood_Sugar                         FALSE
FALSE

## Resting_ECG                                 FALSE
                FALSE

## Maximum_Heart_Rate                           TRUE
FALSE

## Exercise_Induced_Angina                     FALSE
TRUE

## ST_Depression_Exercise                      FALSE
FALSE

## Peak_Exercise_ST_Segment                    FALSE
FALSE

## Num_Major_Vessels_Flourosopy                FALSE
                 FALSE
```

```
## Thalassemia                                    FALSE
FALSE

## Diagnosis_Heart_Disease                        FALSE
FALSE

##                              ST_Depression_Exercise Peak
_Exercise_ST_Segment

## Age                                              FALSE
FALSE

## Sex                                              FALSE
FALSE

## Chest_Pain_Type                                  FALSE
FALSE

## Resting_Blood_Pressure                           FALSE
FALSE

## Cholesterol_serum                                FALSE
FALSE

## Fasting_Blood_Sugar                              FALSE
FALSE

## Resting_ECG                                      FALSE
FALSE

## Maximum_Heart_Rate                               FALSE
FALSE

## Exercise_Induced_Angina                          FALSE
                   FALSE

## ST_Depression_Exercise                            TRUE
FALSE

## Peak_Exercise_ST_Segment                         FALSE
TRUE

## Num_Major_Vessels_Flourosopy                     FALSE
FALSE

## Thalassemia                                      FALSE
FALSE

## Diagnosis_Heart_Disease                          FALSE
FALSE
```

```
##                                Num_Major_Vessels_Flourosop
y Thalassemia
## Age                                                    FALS
E       FALSE
## Sex                                                    FALS
E       FALSE
## Chest_Pain_Type                                        FALS
E       FALSE
## Resting_Blood_Pressure                                 FALS
E       FALSE
## Cholesterol_serum                                      FALS
E       FALSE
## Fasting_Blood_Sugar                                    FALS
E       FALSE
## Resting_ECG                                            FALS
E       FALSE
## Maximum_Heart_Rate                                     FALS
E       FALSE
## Exercise_Induced_Angina                                FALS
E       FALSE
## ST_Depression_Exercise                                 FALS
E       FALSE
## Peak_Exercise_ST_Segment                               FALS
E       FALSE
## Num_Major_Vessels_Flourosopy                            TRU
E       FALSE
## Thalassemia                                            FALS
E        TRUE
## Diagnosis_Heart_Disease                                FALS
E       FALSE
##                               Diagnosis_Heart_Disease
## Age                                             FALSE
## Sex                                             FALSE
## Chest_Pain_Type                                 FALSE
## Resting_Blood_Pressure                          FALSE
```

```
## Cholesterol_serum                                 FALSE
## Fasting_Blood_Sugar                               FALSE
## Resting_ECG                                       FALSE
## Maximum_Heart_Rate                                FALSE
## Exercise_Induced_Angina                           FALSE
## ST_Depression_Exercise                            FALSE
## Peak_Exercise_ST_Segment                          FALSE
## Num_Major_Vessels_Flourosopy                      FALSE
## Thalassemia                                       FALSE
## Diagnosis_Heart_Disease                            TRUE
```

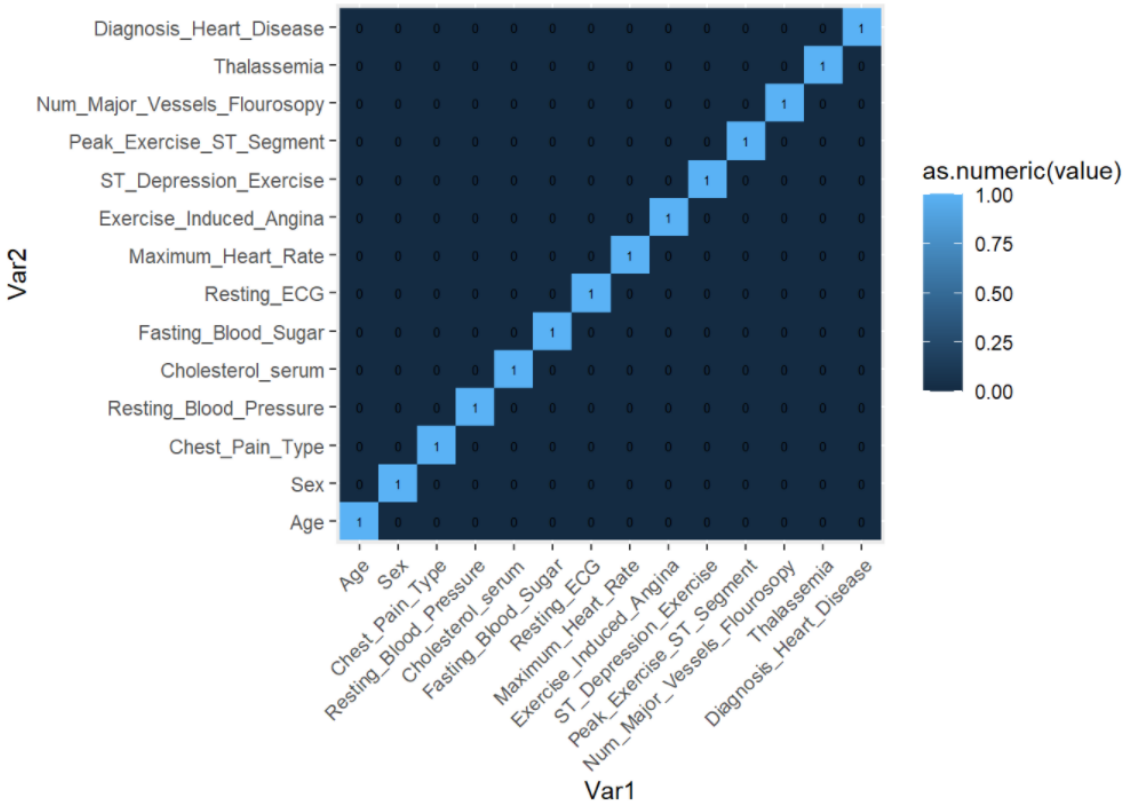
As we find the columns which are highly correlated with each other. Let us view the above information as a heatmap.

```
m_cor = melt(abs(cor(train_heart_df))>0.7)
head(m_cor)
```

```
##                      Var1 Var2 value
## 1                     Age  Age  TRUE
## 2                     Sex  Age FALSE
## 3        Chest_Pain_Type  Age FALSE
## 4 Resting_Blood_Pressure  Age FALSE
## 5      Cholesterol_serum  Age FALSE
## 6    Fasting_Blood_Sugar  Age FALSE
```

```
ggplot(m_cor, aes(x = Var1, y=Var2, fill=as.numeric(value))
) + geom_tile() +
geom_text(aes(Var1, Var2, label=as.numeric(value)),color='b
lack',size=2)+


scale_color_gradient(low='blue',high='red') +
```

```
theme(axis.text.x = element_text(angle=45, vjust=1, hjust=1
))
```



The heat map show's that there are no highly correlated variables.
So as there is no multi collinearity we will removing Variables based on Significance Level.

```
log_regr_hd_model2 = glm(Diagnosis_Heart_Disease~
                         Age+ Sex+
                         Chest_Pain_Type+
                         Resting_Blood_Pressure +
                         Cholesterol_serum +
                         Fasting_Blood_Sugar +
                         Resting_ECG  +
```

```
                              Maximum_Heart_Rate  +
                              Exercise_Induced_Angina  +
                              ST_Depression_Exercise  +
                              Peak_Exercise_ST_Segment +
                              Num_Major_Vessels_Flourosopy+
                              Thalassemia  ,
                              data=train_heart_df, family='b
inomial')


 summary(log_regr_hd_model2)
```

```
##
## Call:
## glm(formula = Diagnosis_Heart_Disease ~ Age + Sex + Ches
t_Pain_Type +
##     Resting_Blood_Pressure + Cholesterol_serum + Fasting
_Blood_Sugar +
##     Resting_ECG + Maximum_Heart_Rate + Exercise_Induced_
Angina +
##     ST_Depression_Exercise + Peak_Exercise_ST_Segment +
Num_Major_Vessels_Flourosopy +
##     Thalassemia, family = "binomial", data = train_heart
_df)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.5633  -0.3960   0.1469   0.5876   2.5025
##
## Coefficients:
##                               Estimate Std. Error z valu
e Pr(>|z|)
## (Intercept)                   3.973030   2.775290   1.43
2  0.15227
```

```
## Age                             -0.009467   0.024845  -0.38
1  0.70317

## Sex                             -1.577591   0.482908  -3.26

7  0.00109 **

## Chest_Pain_Type                  0.804998   0.195659   4.11
4 3.88e-05 ***

## Resting_Blood_Pressure          -0.020459   0.011019  -1.85
7  0.06334 .

## Cholesterol_serum               -0.004707   0.003816  -1.23
3  0.21745

## Fasting_Blood_Sugar              0.222651   0.547886   0.40
6  0.68446

## Resting_ECG                      0.588453   0.361832   1.62
6  0.10388

## Maximum_Heart_Rate               0.021039   0.011166   1.88
4  0.05952 .

## Exercise_Induced_Angina         -1.108258   0.441805  -2.50
8  0.01213 *

## ST_Depression_Exercise          -0.591779   0.224294  -2.63
8  0.00833 **

## Peak_Exercise_ST_Segment         0.329832   0.384015   0.85
9  0.39039

## Num_Major_Vessels_Flourosopy -0.811153   0.205034  -3.95
6 7.62e-05 ***

## Thalassemia                     -0.703026   0.303807  -2.31
4  0.02066 *

## ---

## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1
' ' 1

##

## (Dispersion parameter for binomial family taken to be 1)

##

##     Null deviance: 375.88  on 271  degrees of freedom

## Residual deviance: 194.68  on 258  degrees of freedom

## AIC: 222.68
```

```
##
## Number of Fisher Scoring iterations: 6
```

```
    log_regr_hd_model2 = glm(Diagnosis_Heart_Disease~
                               Age+ Sex+
                               Chest_Pain_Type+
                               Resting_Blood_Pressure +
                               Cholesterol_serum +
                               Resting_ECG  +
                               Maximum_Heart_Rate  +
                               Exercise_Induced_Angina  +
                               ST_Depression_Exercise  +
                               Peak_Exercise_ST_Segment +
                               Num_Major_Vessels_Flourosopy+
                               Thalassemia  ,
                               data=train_heart_df, family='b
inomial')
```

```
 summary(log_regr_hd_model2)
```

```
##
## Call:
## glm(formula = Diagnosis_Heart_Disease ~ Age + Sex + Ches
t_Pain_Type +
##     Resting_Blood_Pressure + Cholesterol_serum + Resting
_ECG +
##     Maximum_Heart_Rate + Exercise_Induced_Angina + ST_De
pression_Exercise +
##     Peak_Exercise_ST_Segment + Num_Major_Vessels_Flouros
opy +
##     Thalassemia, family = "binomial", data = train_heart
_df)
```

```
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.5739  -0.3880   0.1457   0.5919   2.4911
##
## Coefficients:
##                                Estimate Std. Error z valu
e Pr(>|z|)
## (Intercept)                    3.918568   2.775170   1.41
2  0.15795
## Age                           -0.008547   0.024805  -0.34
5  0.73042
## Sex                           -1.565462   0.482564  -3.24
4  0.00118 **
## Chest_Pain_Type                0.821745   0.192280   4.27
4 1.92e-05 ***
## Resting_Blood_Pressure        -0.020022   0.010969  -1.82
5  0.06796 .
## Cholesterol_serum             -0.004677   0.003807  -1.22
8  0.21929
## Resting_ECG                    0.582427   0.361349   1.61
2  0.10700
## Maximum_Heart_Rate             0.021138   0.011169   1.89
3  0.05841 .
## Exercise_Induced_Angina       -1.086876   0.439192  -2.47
5  0.01333 *
## ST_Depression_Exercise        -0.599451   0.223745  -2.67
9  0.00738 **
## Peak_Exercise_ST_Segment       0.320378   0.381555   0.84
0  0.40110
## Num_Major_Vessels_Flourosopy -0.800503   0.203310  -3.93
7 8.24e-05 ***
## Thalassemia                   -0.728334   0.296388  -2.45
7  0.01400 *
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1
' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 375.88  on 271  degrees of freedom
## Residual deviance: 194.84  on 259  degrees of freedom
## AIC: 220.84
##
## Number of Fisher Scoring iterations: 6
```

```
    log_regr_hd_model2 = glm(Diagnosis_Heart_Disease~
                               Age+ Sex+
                               Chest_Pain_Type+
                               Resting_Blood_Pressure +
                               Cholesterol_serum +
                               Resting_ECG  +
                               Exercise_Induced_Angina  +
                               ST_Depression_Exercise  +
                               Peak_Exercise_ST_Segment +
                               Num_Major_Vessels_Flourosopy+
                               Thalassemia  ,
                               data=train_heart_df, family='b
inomial')


 summary(log_regr_hd_model2)
```

```
##
## Call:
## glm(formula = Diagnosis_Heart_Disease ~ Age + Sex + Ches
t_Pain_Type +
##     Resting_Blood_Pressure + Cholesterol_serum + Resting
_ECG +
```

```
##     Exercise_Induced_Angina + ST_Depression_Exercise + P
eak_Exercise_ST_Segment +
##     Num_Major_Vessels_Flourosopy + Thalassemia, family =
"binomial",
##     data = train_heart_df)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.6944  -0.4309   0.1474   0.6051   2.2635
##
## Coefficients:
##                                Estimate Std. Error z valu
e Pr(>|z|)
## (Intercept)                    7.295146   2.183624   3.34
1 0.000835 ***
## Age                           -0.028588   0.022264  -1.28
4 0.199116
## Sex                           -1.566574   0.480370  -3.26
1 0.001109 **
## Chest_Pain_Type                0.877034   0.190754   4.59
8 4.27e-06 ***
## Resting_Blood_Pressure        -0.016599   0.010825  -1.53
3 0.125191
## Cholesterol_serum             -0.003352   0.003666  -0.91
4 0.360528
## Resting_ECG                    0.593102   0.353362   1.67
8 0.093259 .
## Exercise_Induced_Angina       -1.269138   0.422905  -3.00
1 0.002691 **
## ST_Depression_Exercise        -0.651583   0.221848  -2.93
7 0.003313 **
## Peak_Exercise_ST_Segment       0.445766   0.368035   1.21
1 0.225817
## Num_Major_Vessels_Flourosopy -0.818702   0.201514  -4.06
3 4.85e-05 ***
```

```
## Thalassemia                   -0.706691   0.289210  -2.44
4 0.014545 *


## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1
' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 375.88  on 271  degrees of freedom
## Residual deviance: 198.64  on 260  degrees of freedom
## AIC: 222.64
##
## Number of Fisher Scoring iterations: 6
```

```
    log_regr_hd_model2 = glm(Diagnosis_Heart_Disease~
                               Age+ Sex+
                               Chest_Pain_Type+
                               Resting_Blood_Pressure +
                               Resting_ECG  +
                               Exercise_Induced_Angina  +
                               ST_Depression_Exercise  +
                               Peak_Exercise_ST_Segment +
                               Num_Major_Vessels_Flourosopy+
                               Thalassemia  ,
                               data=train_heart_df, family='b
inomial')


 summary(log_regr_hd_model2)
```

```
##
## Call:
```

```
## glm(formula = Diagnosis_Heart_Disease ~ Age + Sex + Chest_Pain_Type +
##     Resting_Blood_Pressure + Resting_ECG + Exercise_Induced_Angina +
##     ST_Depression_Exercise + Peak_Exercise_ST_Segment + Num_Major_Vessels_Flourosopy +
##     Thalassemia, family = "binomial", data = train_heart_df)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.7152  -0.4355   0.1562   0.6121   2.2544
##
## Coefficients:
##                              Estimate Std. Error z value Pr(>|z|)
## (Intercept)                   6.54928    1.99255   3.287 0.00101 **
## Age                          -0.03091    0.02190  -1.411 0.15821
## Sex                          -1.43910    0.45169  -3.186 0.00144 **
## Chest_Pain_Type               0.87637    0.19045   4.602 4.19e-06 ***
## Resting_Blood_Pressure       -0.01643    0.01074  -1.530 0.12611
## Resting_ECG                   0.64640    0.34749   1.860 0.06286 .
## Exercise_Induced_Angina      -1.26358    0.42059  -3.004 0.00266 **
## ST_Depression_Exercise       -0.66699    0.22143  -3.012 0.00259 **
## Peak_Exercise_ST_Segment      0.42274    0.36760   1.150 0.25014
```

```
## Num_Major_Vessels_Flourosopy -0.80033    0.19900  -4.022
5.78e-05 ***
## Thalassemia                  -0.73330    0.28792  -2.547
0.01087 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1
' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 375.88  on 271  degrees of freedom
## Residual deviance: 199.47  on 261  degrees of freedom
## AIC: 221.47
##
## Number of Fisher Scoring iterations: 6
```

```
    log_regr_hd_model2 = glm(Diagnosis_Heart_Disease~
                             Sex+
                             Chest_Pain_Type+
                             Resting_Blood_Pressure +
                             Resting_ECG  +
                             Exercise_Induced_Angina  +
                             ST_Depression_Exercise  +
                             Peak_Exercise_ST_Segment +
                             Num_Major_Vessels_Flourosopy+
                             Thalassemia  ,
                             data=train_heart_df, family='b
inomial')


 summary(log_regr_hd_model2)
```

```
##
```

```
## Call:
## glm(formula = Diagnosis_Heart_Disease ~ Sex + Chest_Pain
_Type +
##     Resting_Blood_Pressure + Resting_ECG + Exercise_Indu
ced_Angina +
##     ST_Depression_Exercise + Peak_Exercise_ST_Segment +
Num_Major_Vessels_Flourosopy +
##     Thalassemia, family = "binomial", data = train_heart
_df)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.6937  -0.4369   0.1705   0.6016   2.2708
##
## Coefficients:
##                              Estimate Std. Error z value
Pr(>|z|)
## (Intercept)                   5.10158    1.68056   3.036
0.00240 **
## Sex                          -1.35046    0.44685  -3.022
0.00251 **
## Chest_Pain_Type               0.88313    0.19023   4.643
3.44e-06 ***
## Resting_Blood_Pressure       -0.01934    0.01061  -1.823
0.06826 .
## Resting_ECG                   0.67496    0.34596   1.951
0.05106 .
## Exercise_Induced_Angina      -1.23112    0.41619  -2.958
0.00310 **
## ST_Depression_Exercise       -0.65605    0.21950  -2.989
0.00280 **
## Peak_Exercise_ST_Segment      0.49104    0.35982   1.365
```

```
  0.17235
## Num_Major_Vessels_Flourosopy -0.84821    0.19583  -4.331
1.48e-05 ***
## Thalassemia                  -0.74224    0.28795  -2.578
0.00995 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1
' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 375.88  on 271  degrees of freedom
## Residual deviance: 201.49  on 262  degrees of freedom
## AIC: 221.49
##
## Number of Fisher Scoring iterations: 6
```

All the variables are significant in the last model.

Let us Make Predictions

```
#Predictions on The Training Set


predictTrain_set = predict(log_regr_hd_model2, type='respon
se')
#Confusion matrix using threshold of 0.5
table(train_heart_df$Diagnosis_Heart_Disease, predictTrain_
set>0.5)
```

```
##
##     FALSE TRUE
##   0    98   29
##   1    16  129
```

Let's compute accuracy.

Accuracy is one metric for evaluating classification models, it is the fraction of predictions our model got right.

```
#Accuracy on The training set

accuracy_LR_train <- (97+137)/nrow(train_heart_df)


accuracy_LR_train
```

```
## [1] 0.8602941
```

```
#Predictions on Test set

predictTest_set = predict(log_regr_hd_model2, newdata=test_
heart_df , type='response')

#Confusion matrix using threshold of 0.5

table(test_heart_df$Diagnosis_Heart_Disease, predictTest_se
t>0.5)
```

```
##

##     FALSE TRUE

##   0     9    2

##   1     2   18
```

```
#Accuracy on The test set

accuracy_LR_test <- (8+15)/nrow(test_heart_df)


accuracy_LR_test
```

```
## [1] 0.7419355
```

Plotting ROCR curve

ROC curve (receiver operating characteristic curve) is a graph showing the performance of a classification model at all classification thresholds. This curve plots two parameters:

1. True Positive Rate
2. False Positive Rate

An ROC curve plots TPR vs. FPR at different classification thresholds. Lowering the classification threshold classifies more items as positive, thus increasing both False Positives and True Positives.

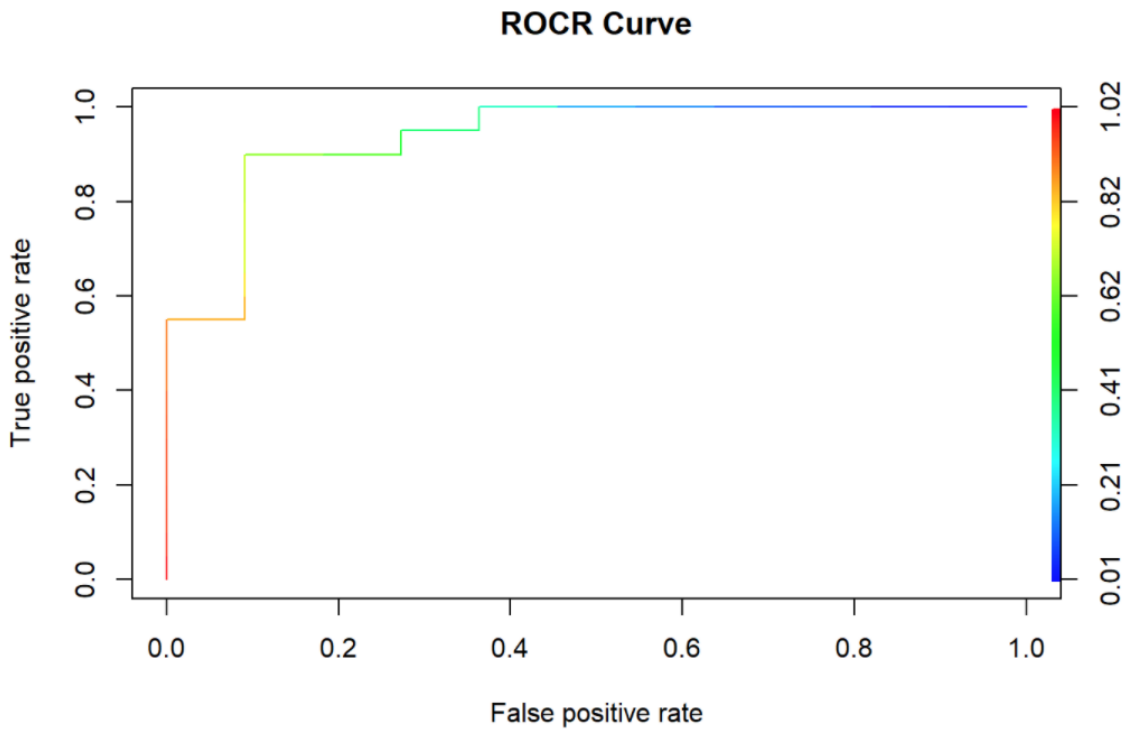
```
ROCRpred = prediction(predictTest_set, test_heart_df$Diagnosis_Heart_Disease)


#The area under curve
area = as.numeric(performance(ROCRpred, 'auc')@y.values)
area
```

```
## [1] 0.9363636
```

So we can see the value of The Area under the curve

Let us show the curve

```
ROCRperf = performance(ROCRpred, 'tpr','fpr')
plot(ROCRperf, colorize=TRUE, main='ROCR Curve')
```



From The curve it seems that true positives are maximized such that maximum number of patients with heart disease are not identified as healthy

```
# We make a data frame to save the results of accuracy

results_table <- data.frame(Methods="Logistic Regression Model", Accuracy_of_Train_Sets=accuracy_LR_train ,Accuracy_of_Test_Sets = accuracy_LR_test )


#results_table
```

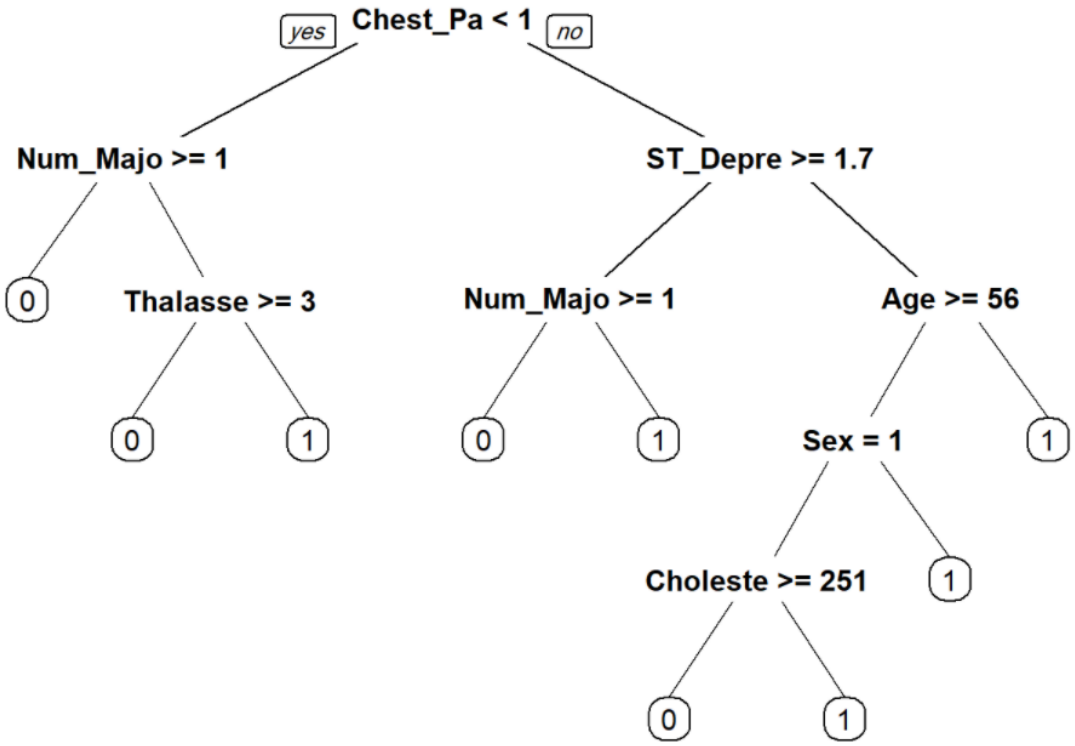
Regression Trees

A tree is basically a flow chart of yes or no questions. The general idea of the methods we are describing is to define an algorithm that uses data to create these trees with predictions at the ends, referred to as nodes.
The general idea is to define an algorithm that uses data to create trees, Regression trees operate by predicting an outcome variable by partitioning the predictors.
A doctor may can decide if a person at risk of having a heart attack by using a decision tree such as the one that we are going to build.

```
#initiate our tree


tree = rpart(Diagnosis_Heart_Disease ~ ., data=train_heart_df, method='class')
```

Let us show the tree that we build using prp function

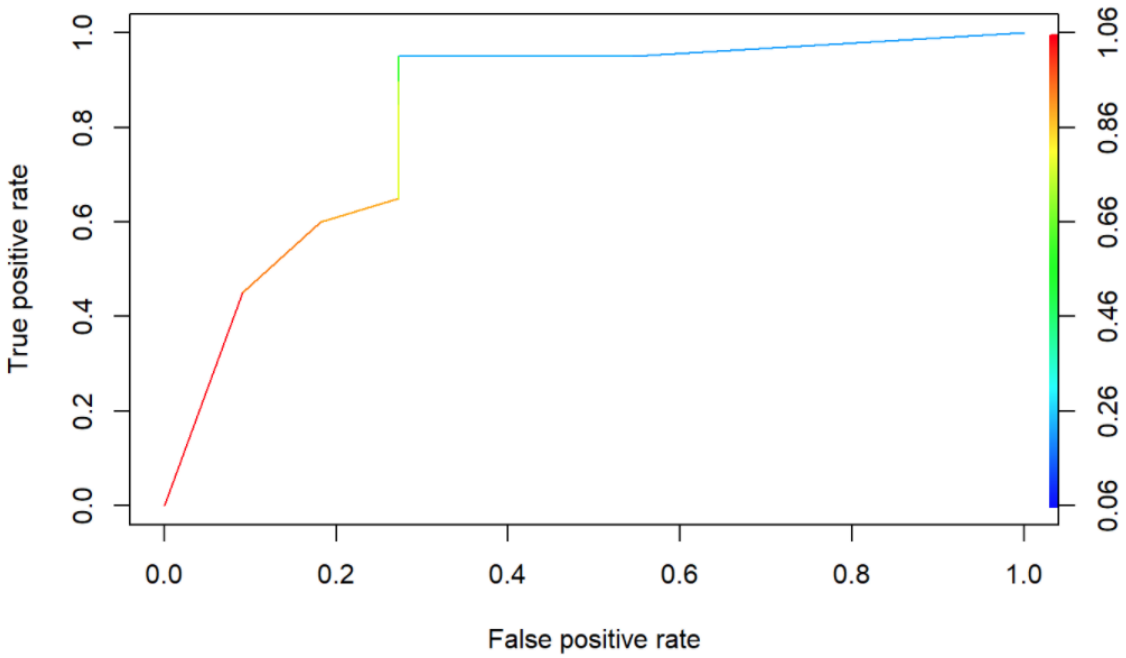
```
# show tree graph

prp(tree)
```




```
#predict_tree for train set

predict_tree_train = predict(tree, newdata=train_heart_df,
type='class')

#predict_tree for test set

predict_tree = predict(tree, newdata=test_heart_df, type='c
lass')

#confusion matrix for Trian set

table(train_heart_df$Diagnosis_Heart_Disease, predict_tree_
train)
```

```
##    predict_tree_train

##       0   1

##   0 102  25

##   1  12 133
```

```
#confusion matrix for Test set
table(test_heart_df$Diagnosis_Heart_Disease, predict_tree)
##    predict_tree
##       0  1
##   0   8  3
##   1   2 18
#Accuracy for train set
accuracy_train_tree <- (102+133)/nrow(train_heart_df)
#Accuracy for test set
accuracy_test_tree <- (7+13)/nrow(test_heart_df)



accuracy_train_tree
## [1] 0.8639706
accuracy_test_tree
## [1] 0.6451613
results_table <- results_table %>% add_row(Methods="Regress
ion Tree Model", Accuracy_of_Train_Sets=accuracy_train_tree
,Accuracy_of_Test_Sets = accuracy_test_tree)


#results_table
predict_tree = predict(tree, newdata=test_heart_df)
ROCR_tree_test = prediction(predict_tree[,2],test_heart_df$
Diagnosis_Heart_Disease)
ROCRperf = performance(ROCR_tree_test, 'tpr','fpr')
plot(ROCRperf,colorize=TRUE)
```




```
as.numeric(performance(ROCR_tree_test, 'auc')@y.values)
```

```
## [1] 0.8272727
```

The area under the curve for the regression tree is less than the logistic regression, which mean more items would be as positive, thus increasing both False Positives and True Positives in the logistic regression model.

Quadratic Discriminant Analysis (QDA)
QDA is a version of Naive Bayes in which we assume that the conditional probabilities for the predictors are multivariate normal.
the QDA method can work well with a few predictors.

Before start our analysis we should convert our variables to factors.

```
# Converting the dependent variables to factors

train_heart_df$Diagnosis_Heart_Disease <- as.factor(train_h
eart_df$Diagnosis_Heart_Disease)
```

```
test_heart_df$Diagnosis_Heart_Disease <- as.factor(test_hea
rt_df$Diagnosis_Heart_Disease)

qda_fit <- train( Diagnosis_Heart_Disease ~ ., method = "qd
a", data = train_heart_df)


qda_predict <- predict(qda_fit, test_heart_df)

confusionMatrix(qda_predict, test_heart_df$Diagnosis_Heart_
Disease)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0  8  3
##          1  3 17
##
##                Accuracy : 0.8065
##                  95% CI : (0.6253, 0.9255)
##     No Information Rate : 0.6452
##     P-Value [Acc > NIR] : 0.04116
##
##                   Kappa : 0.5773
##
##  Mcnemar's Test P-Value : 1.00000
##
##             Sensitivity : 0.7273
##             Specificity : 0.8500
##          Pos Pred Value : 0.7273
##          Neg Pred Value : 0.8500
##              Prevalence : 0.3548
```

```
##          Detection Rate : 0.2581
##    Detection Prevalence : 0.3548
##       Balanced Accuracy : 0.7886
##
##        'Positive' Class : 0
##
```

```
#Accuracy from the previous result
results_table <- results_table %>% add_row(Methods="QDA", A
ccuracy_of_Train_Sets= 0.8065 ,Accuracy_of_Test_Sets = 0.80
65)


#results_table
```

Linear Discriminant Analysis (LDA)
With assumption that all predictors share the same standard deviations and correlations, the boundary will be a line.
Let's start LDA

```
lda_fit <- train(Diagnosis_Heart_Disease ~ ., method = "lda
", data = train_heart_df)
lda_predict <- predict(lda_fit, test_heart_df)
confusionMatrix(lda_predict, test_heart_df$Diagnosis_Heart_
Disease)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0  9  1
##          1  2 19
##
```

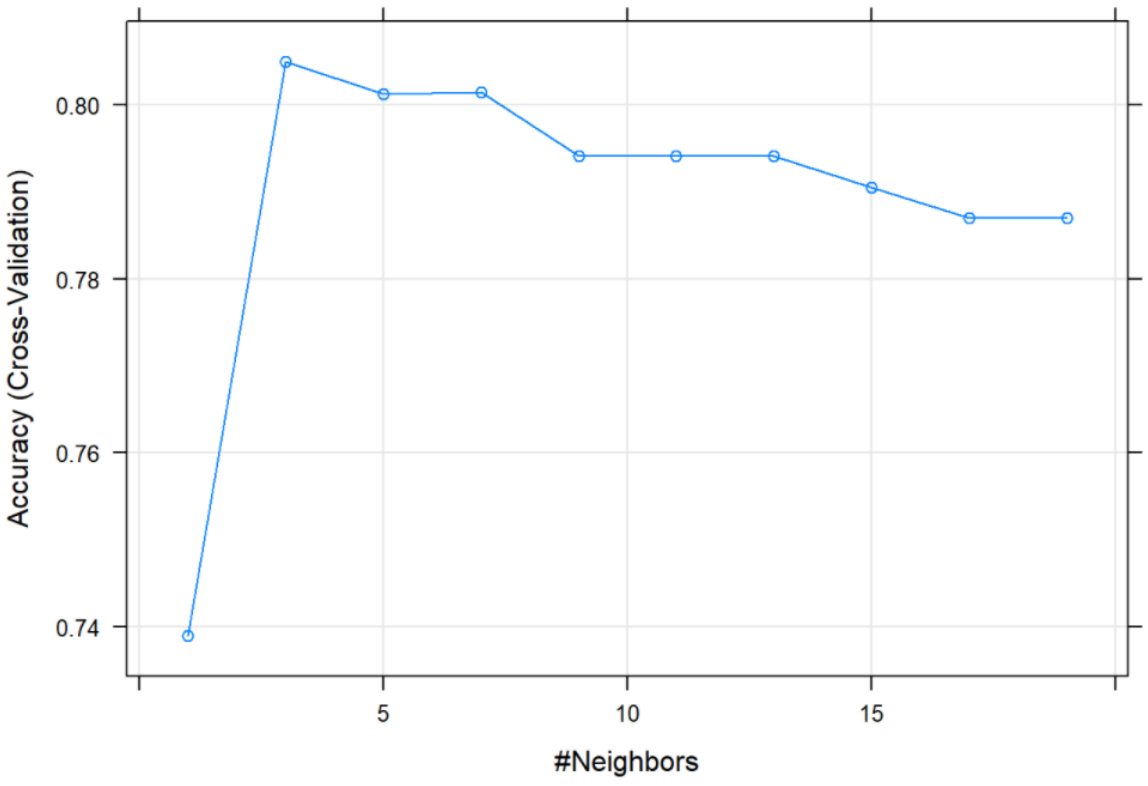
```
##                Accuracy : 0.9032
##                  95% CI : (0.7425, 0.9796)
##     No Information Rate : 0.6452
##     P-Value [Acc > NIR] : 0.001141
##
##                   Kappa : 0.7842
##
##  Mcnemar's Test P-Value : 1.000000
##
##             Sensitivity : 0.8182
##             Specificity : 0.9500
##          Pos Pred Value : 0.9000
##          Neg Pred Value : 0.9048
##              Prevalence : 0.3548
##          Detection Rate : 0.2903
##    Detection Prevalence : 0.3226
##       Balanced Accuracy : 0.8841
##
##        'Positive' Class : 0
##
```

```
#Accuracy from the previous result

results_table <- results_table %>% add_row(Methods="LDA", A
ccuracy_of_Train_Sets= 0.9032 ,Accuracy_of_Test_Sets = 0.90
32)


#results_table
```

KNN Classifier

K-nearest neighbors (KNN) estimates the conditional probabilities in a similar way to bin smoothing. However, KNN is easier to adapt to multiple dimensions.

```
ctrl <- trainControl(method = "cv", verboseIter = FALSE, nu
mber = 5)

knn_fit <- train(Diagnosis_Heart_Disease ~ .,

                 data = train_heart_df, method = "knn", preP
rocess = c("center","scale"),

                 trControl = ctrl , tuneGrid = expand.grid(k
= seq(1, 20, 2)))


plot(knn_fit)
```




```
knn_predict <- predict(knn_fit,newdata = test_heart_df )
```

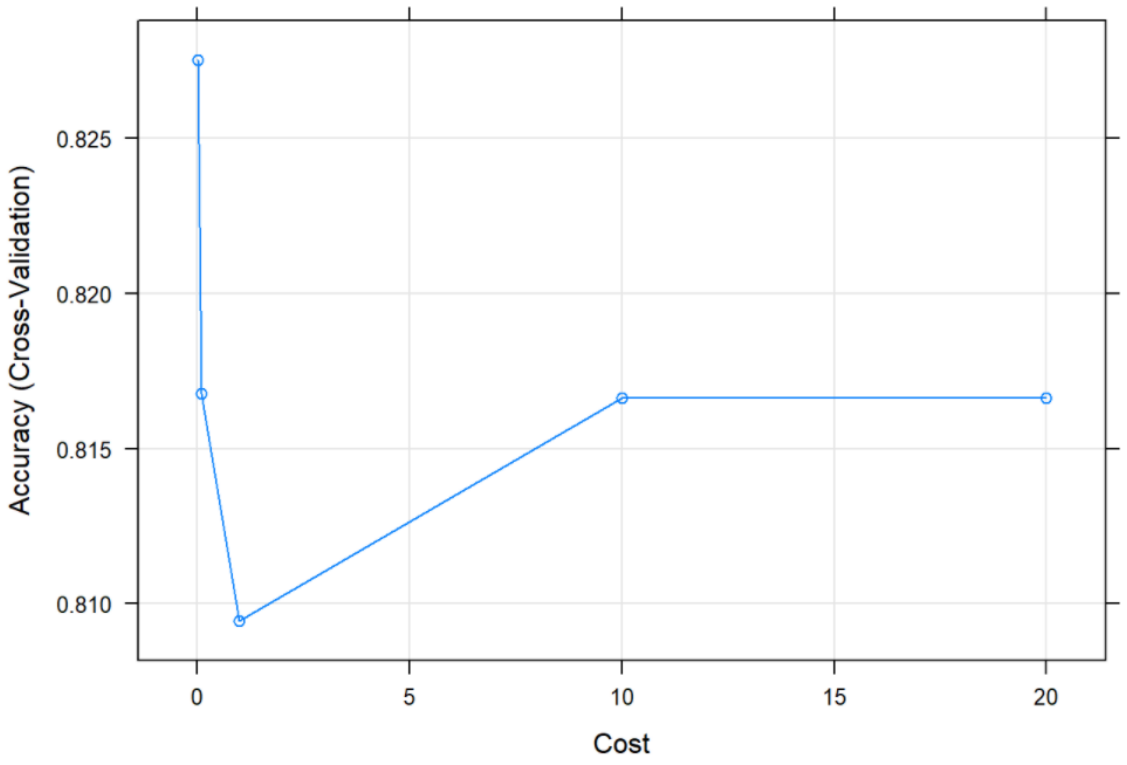
```
knn_results <- confusionMatrix(knn_predict, test_heart_df$D
iagnosis_Heart_Disease )


knn_results
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0  9  3
##          1  2 17
##
##                Accuracy : 0.8387
##                  95% CI : (0.6627, 0.9455)
##     No Information Rate : 0.6452
##     P-Value [Acc > NIR] : 0.01552
##
##                   Kappa : 0.6548
##
##  Mcnemar's Test P-Value : 1.00000
##
##             Sensitivity : 0.8182
##             Specificity : 0.8500
##          Pos Pred Value : 0.7500
##          Neg Pred Value : 0.8947
##              Prevalence : 0.3548
##          Detection Rate : 0.2903
##    Detection Prevalence : 0.3871
##       Balanced Accuracy : 0.8341
##
##        'Positive' Class : 0
```

```
##
#Accuracy from the previous result
results_table <- results_table %>% add_row(Methods="KNN", A
ccuracy_of_Train_Sets= 0.8387  ,Accuracy_of_Test_Sets = 0.8
387 )


#results_table
```

Support Vector Machine (SVM)
SVM is a supervised machine learning algorithm which can be used for both classification or regression challenges. However, it is mostly used in classification problems.
Support Vectors are simply the co-ordinates of individual observation. The SVM classifier is a frontier which best segregates the two classes (hyper-plane/ line).

```
ctrl <- trainControl(method = "cv", verboseIter = FALSE, nu
mber = 5)


grid_svm <- expand.grid(C = c(0.01, 0.1, 1, 10, 20))


svm_fit <- train(Diagnosis_Heart_Disease ~ ., data = train_
heart_df, method = "svmLinear", preProcess = c("center","sc
ale"), tuneGrid = grid_svm, trControl = ctrl)


plot(svm_fit)
```




```
svm_predict <- predict(svm_fit, newdata = test_heart_df)

svm_results <- confusionMatrix(svm_predict, test_heart_df$D
iagnosis_Heart_Disease)


svm_results
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0  8  2
##          1  3 18
##
##                Accuracy : 0.8387
##                  95% CI : (0.6627, 0.9455)
```

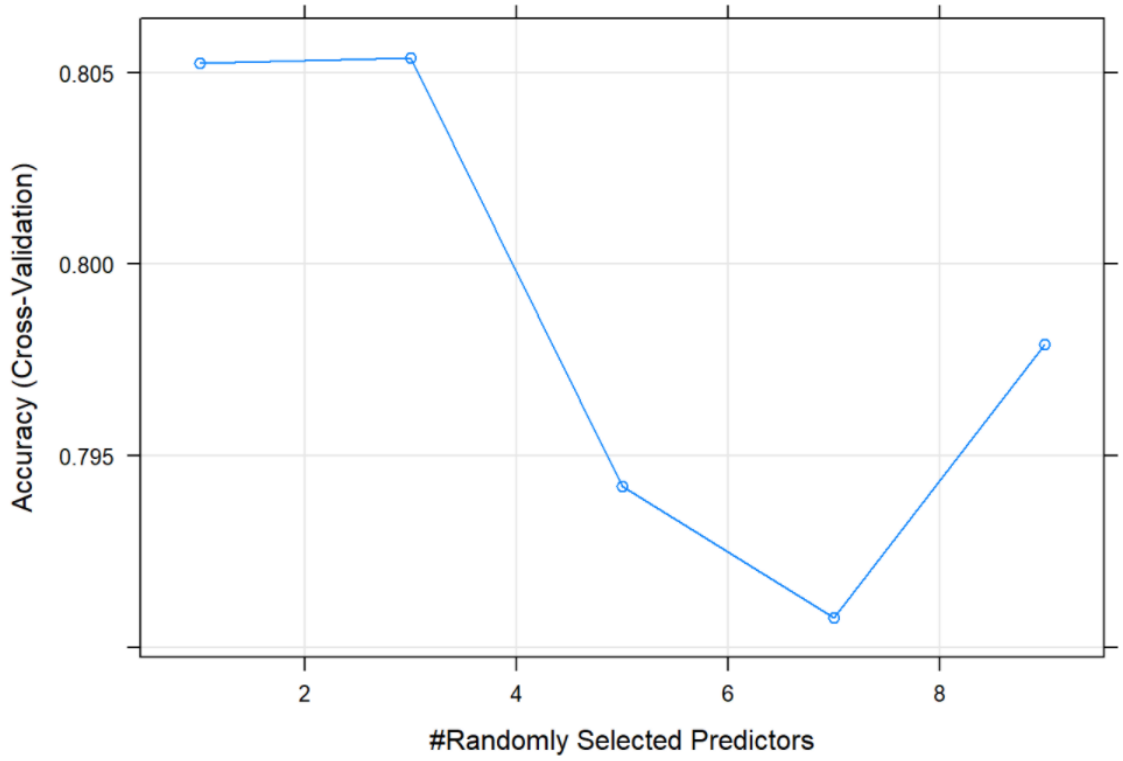
```
##      No Information Rate : 0.6452
##      P-Value [Acc > NIR] : 0.01552
##
##                    Kappa : 0.6404
##
##  Mcnemar's Test P-Value : 1.00000
##
##             Sensitivity : 0.7273
##             Specificity : 0.9000
##          Pos Pred Value : 0.8000
##          Neg Pred Value : 0.8571
##              Prevalence : 0.3548
##          Detection Rate : 0.2581
##    Detection Prevalence : 0.3226
##       Balanced Accuracy : 0.8136
##
##        'Positive' Class : 0
##
```

```
#Accuracy from the previous result

results_table <- results_table %>% add_row(Methods="SVM", A
ccuracy_of_Train_Sets= 0.8387  ,Accuracy_of_Test_Sets = 0.8
387 )


#results_table
```

Random Forest

Random forests are a very popular approach that address the shortcomings of decision trees using a clever idea.
The goal is to improve prediction performance and reduce instability by averaging multiple decision trees, a forest of trees constructed with randomness.

```
control<- trainControl(method = "cv", number = 5, verboseIt
er = FALSE)

grid <-data.frame(mtry = seq(1, 10, 2))

rf_fit <- train(Diagnosis_Heart_Disease ~ ., method = "rf",
data = train_heart_df, ntree = 20, trControl = control,

                  tuneGrid = grid)


plot(rf_fit)
```



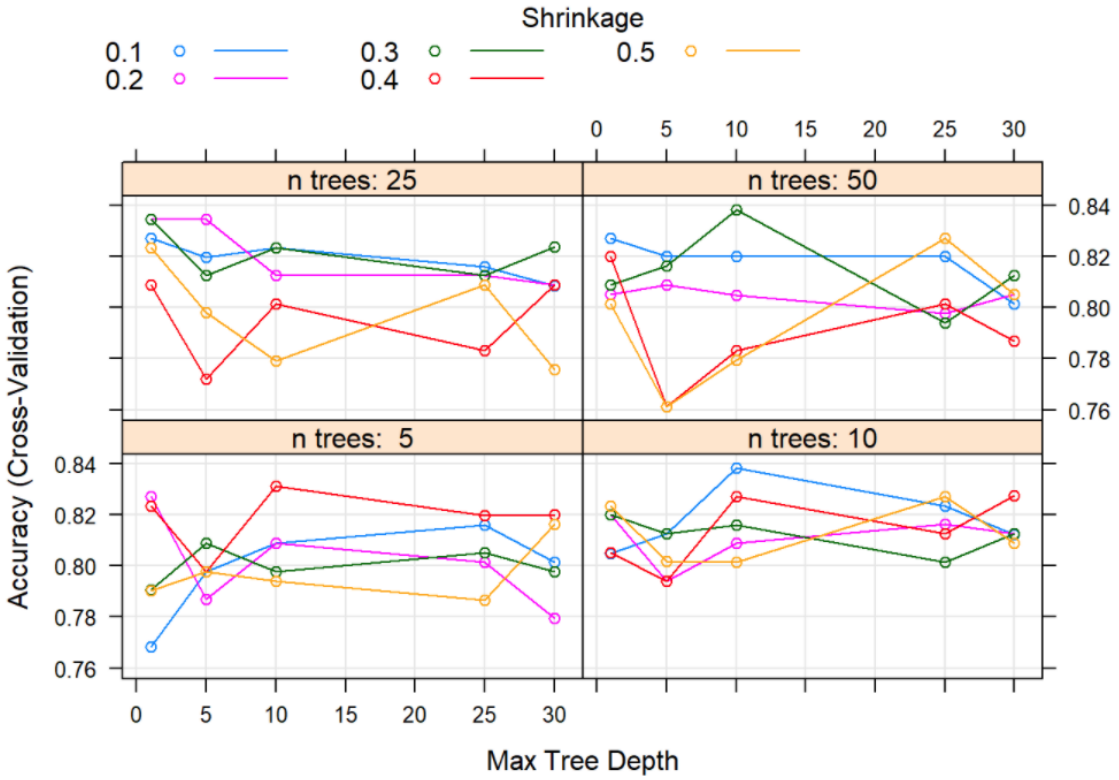

```
rf_predict <- predict(rf_fit, newdata = test_heart_df)


rf_results <- confusionMatrix(rf_predict, test_heart_df$Dia
gnosis_Heart_Disease)
```

```
rf_results
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0  8  2
##          1  3 18
##
##                Accuracy : 0.8387
##                  95% CI : (0.6627, 0.9455)
##     No Information Rate : 0.6452
##     P-Value [Acc > NIR] : 0.01552
##
##                   Kappa : 0.6404
##
##  Mcnemar's Test P-Value : 1.00000
##
##             Sensitivity : 0.7273
##             Specificity : 0.9000
##          Pos Pred Value : 0.8000
##          Neg Pred Value : 0.8571
##              Prevalence : 0.3548
##          Detection Rate : 0.2581
##    Detection Prevalence : 0.3226
##       Balanced Accuracy : 0.8136
##
##        'Positive' Class : 0
```

```
##
#Accuracy from the previous result

results_table <- results_table %>% add_row(Methods="RF", Ac
curacy_of_Train_Sets= 0.8387  ,Accuracy_of_Test_Sets = 0.83
87 )


#results_table
```

Gradient Boosting Machine (GBM)
GBM constructs a forward stage-wise additive model by implementing gradient descent in function space.
GBM build an ensemble of shallow and weak successive trees.

```
gbmGrid <-  expand.grid(interaction.depth = c(1, 5, 10, 25,
30),n.trees = c(5, 10, 25, 50), shrinkage = c(0.1, 0.2, 0.3
,  0.4, 0.5), n.minobsinnode = 20)


gbm_fit <- train(Diagnosis_Heart_Disease ~ ., method = "gbm
", data = train_heart_df,  trControl = control, verbose = F
ALSE,tuneGrid = gbmGrid)


plot(gbm_fit)
```




```
gbm_predict <- predict(gbm_fit, newdata = test_heart_df)


gbm_results <- confusionMatrix(gbm_predict, test_heart_df$D
iagnosis_Heart_Disease)


gbm_results
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0  8  3
##          1  3 17
##
##                Accuracy : 0.8065
```

```
##                  95% CI : (0.6253, 0.9255)
##     No Information Rate : 0.6452
##     P-Value [Acc > NIR] : 0.04116
##
##                   Kappa : 0.5773
##
##  Mcnemar's Test P-Value : 1.00000
##
##             Sensitivity : 0.7273
##             Specificity : 0.8500
##          Pos Pred Value : 0.7273
##          Neg Pred Value : 0.8500
##              Prevalence : 0.3548
##          Detection Rate : 0.2581
##    Detection Prevalence : 0.3548
##       Balanced Accuracy : 0.7886
##
##        'Positive' Class : 0
##
```

```
#Accuracy from the previous result

results_table <- results_table %>% add_row(Methods="GBM", A
ccuracy_of_Train_Sets= 0.8065  ,Accuracy_of_Test_Sets = 0.8
065 )


#results_table
```

Results of the accuracy of the predictions

As we see the resulting table which shows the overall accuracy for each model we build, the model that gives us the higher accuracy is LDA model.

```
#Print the final results table

results_table
```

```
##                     Methods Accuracy_of_Train_Sets Accuracy_of_Test_Sets
## 1 Logistic Regression Model              0.8602941             0.7419355
## 2     Regression Tree Model              0.8639706             0.6451613
## 3                       QDA              0.8065000             0.8065000
## 4                       LDA              0.9032000             0.9032000
## 5                       KNN              0.8387000             0.8387000
## 6                       SVM              0.8387000             0.8387000
## 7                        RF              0.8387000             0.8387000
## 8                       GBM              0.8065000             0.8065000
```

Conclusion

We can see that the LDA model gives us a good accuracy result 0.9032000 , it seems that LDA worked for this data set.
The other models gives us a good accuracy results approximately 0.80
We can't all be cardiologists but the models that we build is a very good methods to predict if individual would have a heart disease or not, this would improve the methods of predictions and diagnosing diseases in future.

Future Work

As a future plans more machine learning models would be built to find if we can get a higher rate of accuracy, also ensemble method should be considered to apply on such Data set, to combine the advantages of various models and enhance the overall performance of prediction.

References:

https://wanjirumaggie45.medium.com/data-science-for-good-machine-learning-for-heart-disease-prediction-289234651fed

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

https://www.kaggle.com/rashikrahmanpritom/heart-attack-analysis-prediction-dataset

https://www.kaggle.com/tentotheminus9/what-causes-heart-disease-explaining-the-model

https://towardsdatascience.com/heart-disease-uci-diagnosis-prediction-b1943ee835a7

https://www.kaggle.com/ronitf/heart-disease-uci

https://www.r-bloggers.com/2019/09/heart-disease-prediction-from-patient-data-in-r/

https://www.kaggle.com/snogard/heart-disease-uci-using-r

https://www.kaggle.com/faressayah/predicting-heart-disease-using-machine-learning

https://towardsdatascience.com/heart-disease-prediction-73468d630cfc

https://towardsdatascience.com/predicting-presence-of-heart-diseases-using-machine-learning-36f00f3edb2c

https://towardsdatascience.com/boosting-algorithm-gbm-97737c63daa3

https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc

https://en.wikipedia.org/wiki/Logistic_regression

https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code/

https://rpubs.com/phamdinhkhanh/389752

http://www.cookbook-r.com/Graphs/Colors_(ggplot2)/

https://ggplot2.tidyverse.org/reference/scale_manual.html