Topic: Heart Disease Prediction

Group Members:

1. Shubham Patil 17101A0027
2. Neha Mayekar 17101A0030
3. Shreya Pattewar 17101A0033

Link to the dataset: https://www.kaggle.com/ronitf/heart-disease-uci

Dataset Description

The dataset contains many medical indicators. This dataset contains 14 attributes.

- slope_of_peak_exercise_st_segment (type: int): the slope of the peak exercise ST segment, an electrocardiography read out indicating quality of blood flow to the heart
- thal (type: categorical): results of thallium stress test measuring blood flow to the heart, with possible values normal, fixed_defect, reversible_defect
- resting_blood_pressure (type: int): resting blood pressure
- chest_pain_type (type: int): chest pain type (4 values)
- num_major_vessels (type: int): number of major vessels (0-3) colored by flourosopy
- fasting_blood_sugar_gt_120_mg_per_dl (type: binary): fasting blood sugar > 120 mg/dl
- resting_ekg_results (type: int): resting electrocardiographic results (values 0,1,2)
- serum_cholesterol_mg_per_dl (type: int): serum cholestoral in mg/dl
- oldpeak_eq_st_depression (type: float): oldpeak = ST depression induced by exercise relative to rest, a measure of abnormality in electrocardiograms
- sex (type: binary): 0: female, 1: male
- age (type: int): age in years
- max_heart_rate_achieved (type: int): maximum heart rate achieved (beats per minute)
- exercise_induced_angina (type: binary): exercise-induced chest pain (0: False, 1: True)

Aim:

1. Exploratory data analysis and Data Visualisation

2. To predict if a person has a heart disease or not based on attributes blood pressure, heart beat, exang, fbs and others.

Analysis:

Import Libraries:

```
> library(tidyverse)
> library(dplyr)
```

Library dplyr is used for data manipulation

Import and Explore dataset

```
> head(data)
  ï..age sex cp trestbps chol fbs restecg thalach exang oldpeak slope ca t
hal target
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
> str(data)
'data.frame':  303 obs. of  14 variables:
 $ ï..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 ...
 $ trestbps: 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 ...
 $ thalach : int  150 187 172 178 163 148 153 173 162 174 ...
 $ exang   : 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 ...
 $ slope   : int  0 0 2 2 2 1 1 2 2 2 ...
 $ ca      : int  0 0 0 0 0 0 0 0 0 0 ...
 $ thal    : int  1 2 2 2 2 1 2 3 3 2 ...
 $ target  : int  1 1 1 1 1 1 1 1 1 1 ...
> summary(data)
     ï..age           sex               cp           trestbps          cho
l
```

```
 Min.   :29.00   Min.   :0.0000   Min.   :0.000   Min.   : 94.0   Min.   :
126.0
 1st Qu.:47.50   1st Qu.:0.0000   1st Qu.:0.000   1st Qu.:120.0   1st Qu.:
211.0
 Median :55.00   Median :1.0000   Median :1.000   Median :130.0   Median :
240.0
 Mean   :54.37   Mean   :0.6832   Mean   :0.967   Mean   :131.6   Mean   :
246.3
 3rd Qu.:61.00   3rd Qu.:1.0000   3rd Qu.:2.000   3rd Qu.:140.0   3rd Qu.:
274.5
 Max.   :77.00   Max.   :1.0000   Max.   :3.000   Max.   :200.0   Max.   :
564.0
      fbs             restecg          thalach          exang            old
peak
 Min.   :0.0000   Min.   :0.0000   Min.   : 71.0   Min.   :0.0000   Min.
:0.00
 1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:133.5   1st Qu.:0.0000   1st Qu
.:0.00
 Median :0.0000   Median :1.0000   Median :153.0   Median :0.0000   Median
:0.80
 Mean   :0.1485   Mean   :0.5281   Mean   :149.6   Mean   :0.3267   Mean
:1.04
 3rd Qu.:0.0000   3rd Qu.:1.0000   3rd Qu.:166.0   3rd Qu.:1.0000   3rd Qu
.:1.60
 Max.   :1.0000   Max.   :2.0000   Max.   :202.0   Max.   :1.0000   Max.
:6.20
     slope             ca               thal           target
 Min.   :0.000   Min.   :0.0000   Min.   :0.000   Min.   :0.0000
 1st Qu.:1.000   1st Qu.:0.0000   1st Qu.:2.000   1st Qu.:0.0000
 Median :1.000   Median :0.0000   Median :2.000   Median :1.0000
 Mean   :1.399   Mean   :0.7294   Mean   :2.314   Mean   :0.5446
 3rd Qu.:2.000   3rd Qu.:1.0000   3rd Qu.:3.000   3rd Qu.:1.0000
 Max.   :2.000   Max.   :4.0000   Max.   :3.000   Max.   :1.0000
```

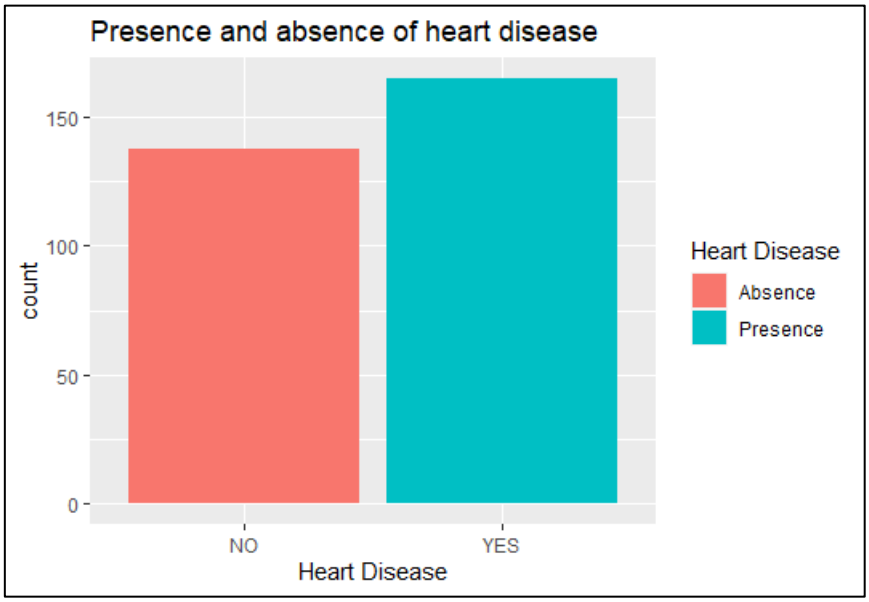
Data Transformation - mutating makes data visualization easy

```
> data2 <- data %>%
+   mutate(sex = if_else(sex==1, "MALE", "FEMALE"),
+          fbs = if_else(fbs==1, ">120", "<=120"),
+          exang = if_else(exang==1, "YES", "NO"),
+          cp = if_else(cp==1, "ATYPICAL ANGINA",
+                       if_else(cp==2, "NON-ANGINAL PAIN", "ASYMPTOMATIC")
),
+          restecg = if_else(restecg==0,"NORMAL",
+                            if_else(restecg==1, "ABNORMALITY", "PROBABLE
OR DEFINITE")),
+          slope = as.factor(slope),
+          ca = as.factor(ca),
+          thal = as.factor(thal),
+          target = if_else(target==1,"YES","NO")
+   ) %>%
+   mutate_if(is.character, as.factor) %>%
+   dplyr::select(target, sex, fbs, exang, cp, restecg, slope, ca, thal, e
verything())
> head(data2)
  target    sex    fbs exang                cp     restecg slope ca thal ï..
age trestbps chol
```

```
1    YES   MALE  >120    NO      ASYMPTOMATIC       NORMAL     0  0    1
63      145  233
2    YES   MALE <=120    NO NON-ANGINAL PAIN ABNORMALITY     0  0    2
37      130  250
3    YES FEMALE <=120    NO  ATYPICAL ANGINA       NORMAL     2  0    2
41      130  204
4    YES   MALE <=120    NO  ATYPICAL ANGINA ABNORMALITY     2  0    2
56      120  236
5    YES FEMALE <=120   YES      ASYMPTOMATIC ABNORMALITY     2  0    2
57      120  354
6    YES   MALE <=120    NO      ASYMPTOMATIC ABNORMALITY     1  0    1
57      140  192
  thalach oldpeak
1     150     2.3
2     187     3.5
3     172     1.4
4     178     0.8
5     163     0.6
6     148     0.4
```

Data Visualisation

```
> library(ggplot2)
> ggplot(data2, aes(x=target, fill=target))+
+   geom_bar() +
+   xlab("Heart Disease")+
+   ggtitle("Presence and absence of heart disease")+
+   scale_fill_discrete(name="Heart Disease", label=c("Absence","Presence"
))
```



bar plot for target (heart disease)

from bar plot we can see number of patients getting heart disease are more than number of patience with NO heart disease

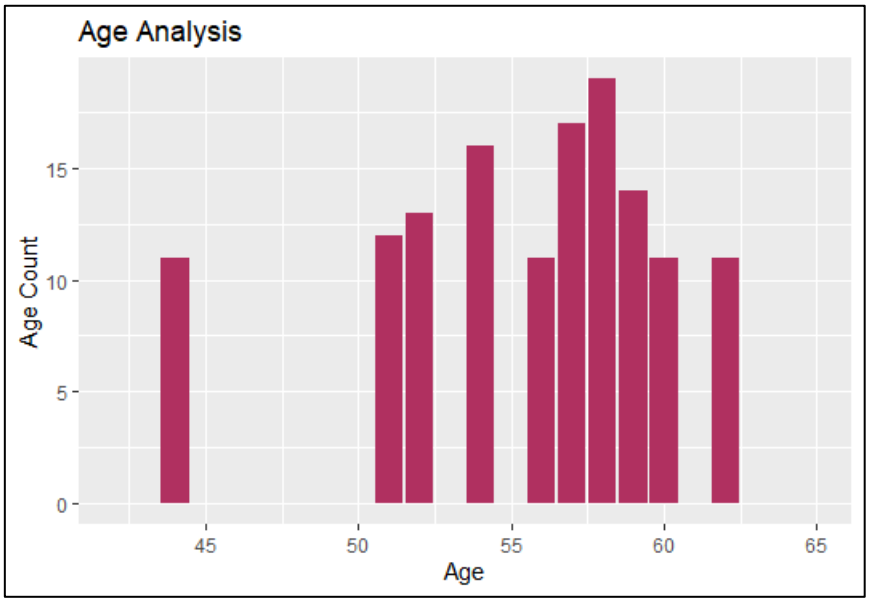
```
> prop.table(table(data2$target))

       NO        YES
0.4554455 0.5445545
```

this gives that 45% of people does not have any heart disease, however 54 % of people suffer from heart disease, so we can say data is almost balanced but more focus towards people with heart disease

lets explore age variable

count frequencies of the values of age

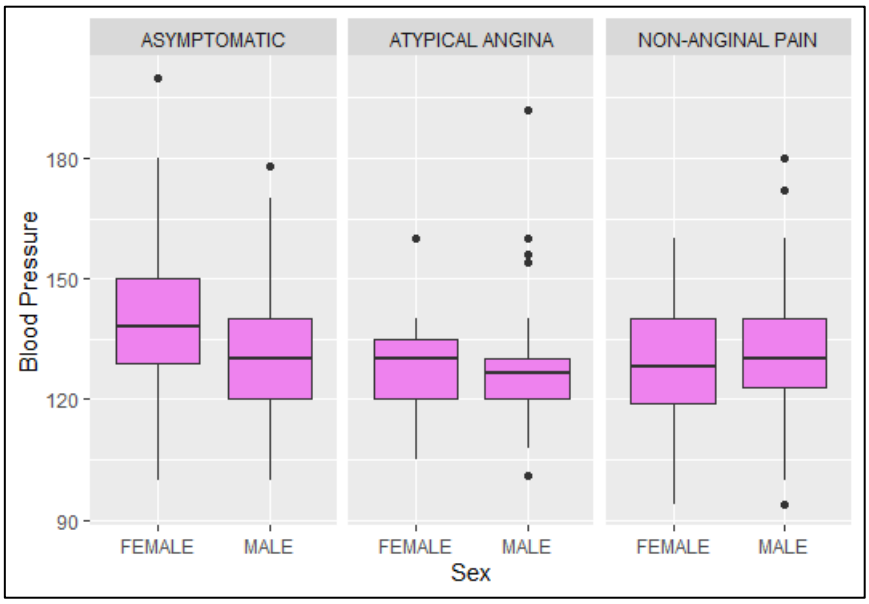
```
> data2 %>%
+   group_by(ï..age) %>%
+   count() %>%
+   filter(n>10) %>%
+   ggplot() +
+   geom_col(aes(ï..age,n), fill='maroon')+
+   ggtitle("Age Analysis")+
+   xlab("Age") +
+   ylab("Age Count")+
+   xlim(c(42,65))
```



above count plot shows dataset contains more number of people with age greater than 50,

Lets compare blood pressure across the chest pain

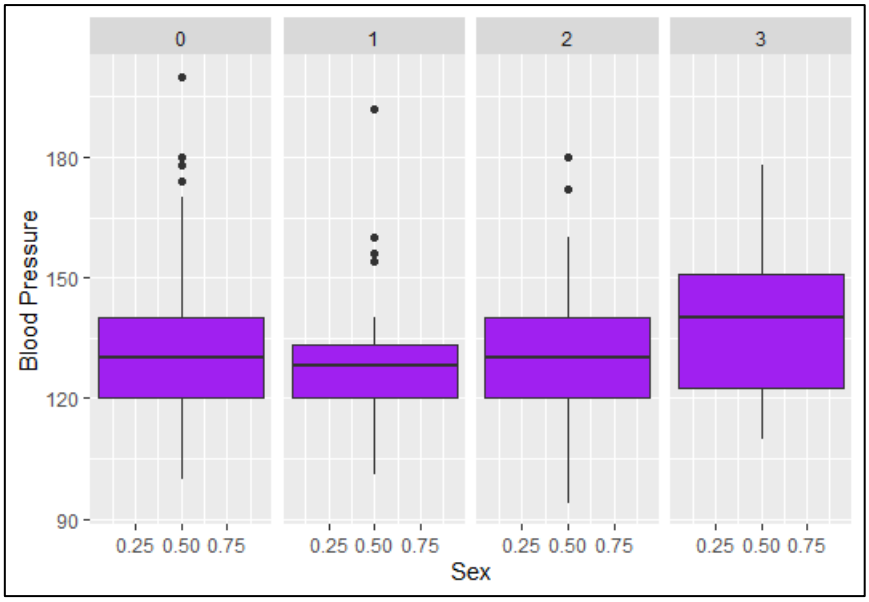
```
> data2 %>%
+   ggplot(aes(x=sex, y=trestbps))+
+   geom_boxplot(fill="violet")+
+   xlab("Sex")+
+   ylab("Blood Pressure")+
+   facet_grid(~cp)
```



above plot shows us that in case of asymptomatic chest pain - hardly any outlier for females and males, in atypical angina chest pain, hardly any outliers for females, but outliers exists in case of males, in atypical angina, males having higher blood pressure feels atypical angina chest pain

Suppose if we have not mutated the data

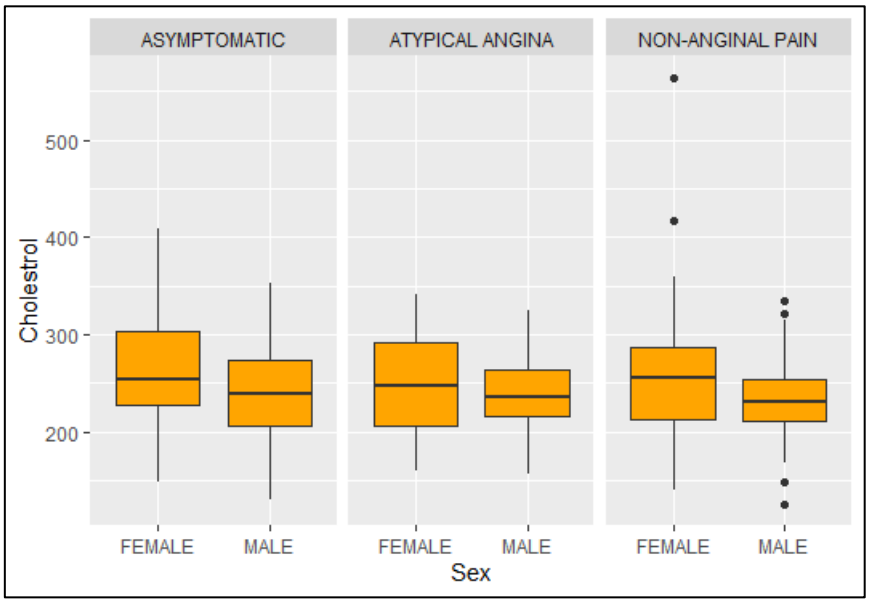
```
> data %>%
+   ggplot(aes(x=sex, y=trestbps))+
+   geom_boxplot(fill="purple")+
+   xlab("Sex")+
+   ylab("Blood Pressure")+
+   facet_grid(~cp)
```



now above graph does not give any useful information

As column names are not understandable, also male and female entries are combined and we can't say outliers belongs to male or females

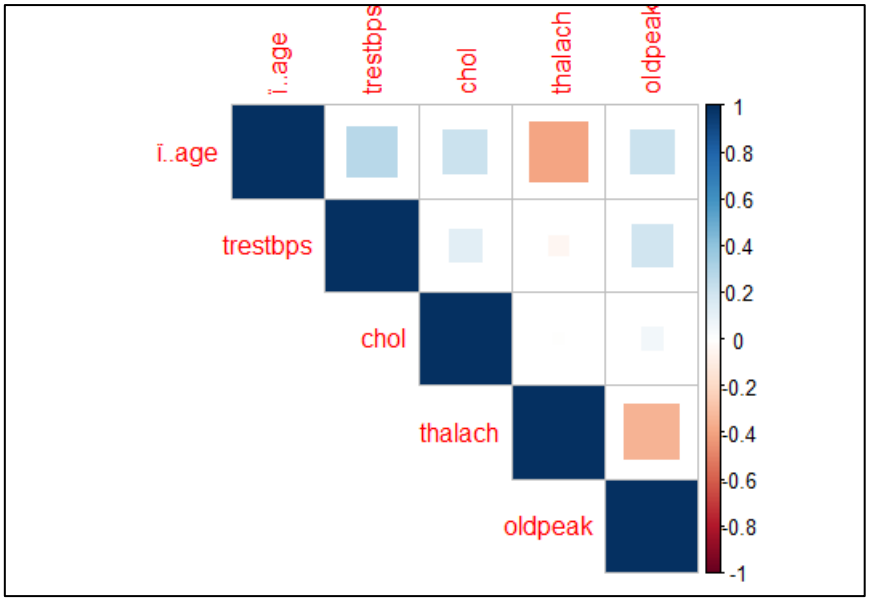
```
> data2 %>%
+   ggplot(aes(x=sex, y=chol))+
+   geom_boxplot(fill="orange")+
+   xlab("Sex")+
+   ylab("Cholestrol")+
+   facet_grid(~cp)
```



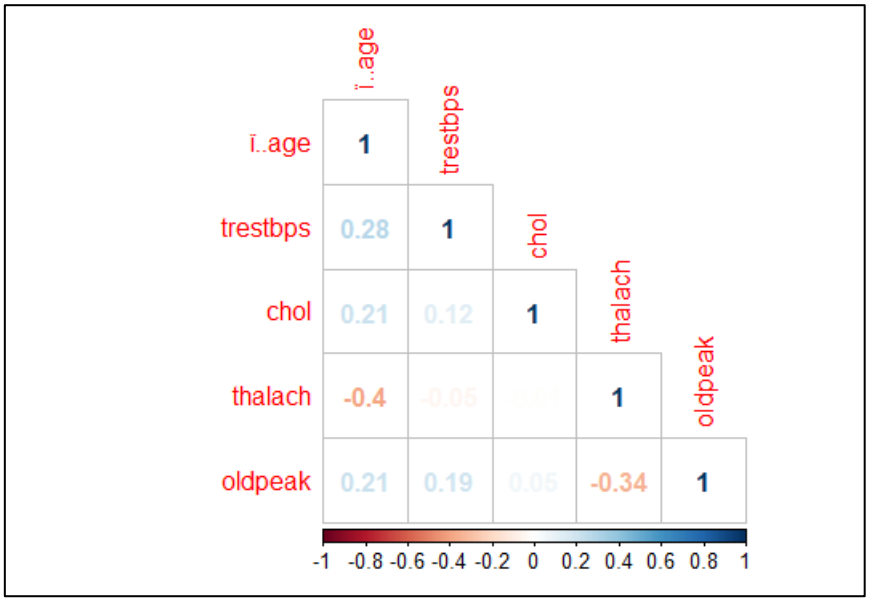
from above graph we can interpret, cholesterol levels in females are more than males in all cases, also outliers exist in non-anginal pain (Chest pain)

Correlation

```
> library(corrplot)
corrplot 0.84 loaded
> library(ggplot2)
> str(data2)
'data.frame':  303 obs. of  14 variables:
 $ target  : Factor w/ 2 levels "NO","YES": 2 2 2 2 2 2 2 2 2 2 ...
 $ sex     : Factor w/ 2 levels "FEMALE","MALE": 2 2 1 2 1 2 1 2 2 2 ...
 $ fbs     : Factor w/ 2 levels "<=120",">120": 2 1 1 1 1 1 1 1 2 1 ...
 $ exang   : Factor w/ 2 levels "NO","YES": 1 1 1 1 2 1 1 1 1 1 ...
 $ cp      : Factor w/ 3 levels "ASYMPTOMATIC",..: 1 3 2 2 1 1 2 2 3 3 ...
 $ restecg : Factor w/ 3 levels "ABNORMALITY",..: 2 1 2 1 1 1 2 1 1 1 ...
 $ slope   : Factor w/ 3 levels "0","1","2": 1 1 3 3 3 2 2 3 3 3 ...
 $ ca      : Factor w/ 5 levels "0","1","2","3",..: 1 1 1 1 1 1 1 1 1 1 ..
.
 $ thal    : Factor w/ 4 levels "0","1","2","3": 2 3 3 3 3 2 3 4 4 3 ...
 $ ï..age  : int  63 37 41 56 57 57 56 44 52 57 ...
 $ trestbps: int  145 130 130 120 120 140 140 120 172 150 ...
 $ chol    : int  233 250 204 236 354 192 294 263 199 168 ...
 $ thalach : int  150 187 172 178 163 148 153 173 162 174 ...
 $ oldpeak : num  2.3 3.5 1.4 0.8 0.6 0.4 1.3 0 0.5 1.6 ...
> cor_heart <- cor(data2[,10:14])

> cor_heart
             ï..age    trestbps         chol      thalach     oldpeak
ï..age    1.0000000  0.27935091  0.213677957 -0.398521938  0.21001257
trestbps  0.2793509  1.00000000  0.123174207 -0.046697728  0.19321647
chol      0.2136780  0.12317421  1.000000000 -0.009939839  0.05395192
thalach  -0.3985219 -0.04669773 -0.009939839  1.000000000 -0.34418695
oldpeak   0.2100126  0.19321647  0.053951920 -0.344186948  1.00000000
> corrplot(cor_heart, method='square',type='upper')
```

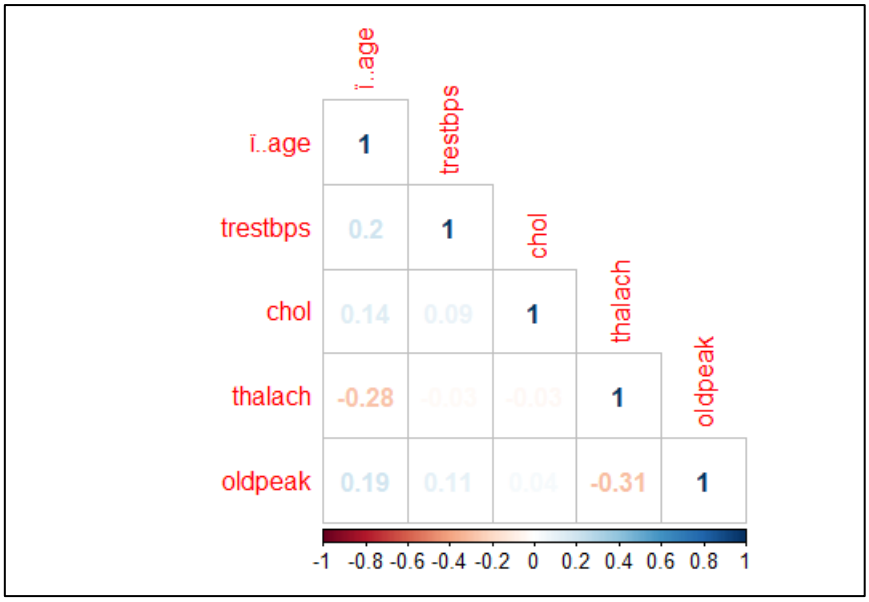



```
> corrplot(cor_heart, method='number',type='lower')
```



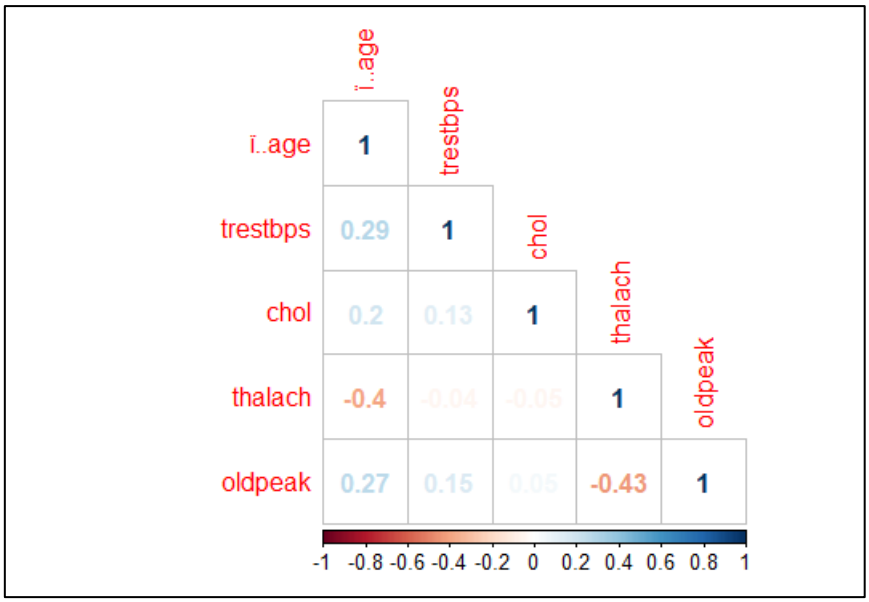
By using kendall method for correlation:

```
> cor_heart <- cor(data2[,10:14], method = "kendall")
> corrplot(cor_heart, method='number',type='lower')
```



By using spearman method for correlation

```
> cor_heart <- cor(data2[,10:14], method = "spearman")
> corrplot(cor_heart, method='number',type='lower')
```



By looking at correlation plots, we can interpret that variables are not well correlated with each other.

Prediction

```
> head(data2)
  target    sex   fbs exang               cp     restecg slope ca thal ï..
age trestbps chol
1    YES   MALE  >120    NO     ASYMPTOMATIC      NORMAL     0  0    1
63      145  233
2    YES   MALE <=120    NO NON-ANGINAL PAIN ABNORMALITY     0  0    2
37      130  250
3    YES FEMALE <=120    NO  ATYPICAL ANGINA      NORMAL     2  0    2
41      130  204
4    YES   MALE <=120    NO  ATYPICAL ANGINA ABNORMALITY     2  0    2
56      120  236
5    YES FEMALE <=120   YES     ASYMPTOMATIC ABNORMALITY     2  0    2
57      120  354
6    YES   MALE <=120    NO     ASYMPTOMATIC ABNORMALITY     1  0    1
57      140  192
  thalach oldpeak
1     150     2.3
2     187     3.5
3     172     1.4
4     178     0.8
5     163     0.6
6     148     0.4
> table(data2$target)

 NO YES
138 165
```

We have 165 entries in dataset in for which target is 1 and 138 for which target is 0

```
> library(caTools)
```

caTools library is used for ML in R

set seed (value) where value specifies the initial value of the random number seed. Syntax: set.seed(123) In the above line,123 is set as the random number value. The main point of using the seed is to be able to reproduce a particular sequence of 'random' numbers. and sed(n) reproduces random numbers results by seed.

```
> set.seed(123)
```

Splitting dataset into train and test in 75:25 ratio

```
> split=sample.split(data2$target, SplitRatio = 0.75)
> dataset=data2
> qualityTrain=subset(dataset,split == TRUE)
> qualityTest=subset(dataset,split == FALSE)
```

```
> nrow(qualityTrain)
[1] 228
> nrow(qualityTest)
[1] 75
```

qualityTrain contains 228 rows, where as qualityTest contains 75 rows.

```
> colnames(dataset)
 [1] "target"   "sex"      "fbs"      "exang"    "cp"       "restecg"  "sl
ope"      "ca"
 [9] "thal"     "ï..age"   "trestbps" "chol"     "thalach"  "oldpeak"
> names(dataset)[names(dataset) == "ï..age"] <- "age"
> colnames(dataset)
 [1] "target"   "sex"      "fbs"      "exang"    "cp"       "restecg"  "sl
ope"      "ca"
 [9] "thal"     "age"      "trestbps" "chol"     "thalach"  "oldpeak"
```

In original dataset, name of age attribute was something anonymous, so we renamed that variable.

Build the ML model

The dependent variable used is ***target****, for the independent variable* ***is age, trestbps, chol, fbs, restecg, thalach, exang, oldpeak, slope, ca,*** *and* ***thal****.*

glm is the generalized linear model we will be using. target~ *means that we want to model target using (~) every available feature (.). family = binomial() is used because we are predicting a binary outcome, 0 or 1.*

```
> datasetlog=glm(target~age+sex+cp+trestbps+chol+fbs+restecg+thalach+exang
+oldpeak+slope+ca+thal,
+              data=qualityTrain, family='binomial')
> summary(datasetlog)

Call:
glm(formula = target ~ age + sex + cp + trestbps + chol + fbs +
    restecg + thalach + exang + oldpeak + slope + ca + thal,
    family = "binomial", data = qualityTrain)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.3391  -0.3178   0.1193   0.5209   2.6425

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  5.014706   3.100650   1.617 0.105812
age         -0.013647   0.027109  -0.503 0.614670
sex         -2.351734   0.609611  -3.858 0.000114 ***
cp           0.947569   0.229796   4.124 3.73e-05 ***
trestbps    -0.018008   0.011797  -1.526 0.126887
chol        -0.006038   0.004264  -1.416 0.156791
fbs         -0.432079   0.687178  -0.629 0.529497
restecg      0.559518   0.425306   1.316 0.188320
thalach      0.020866   0.012797   1.631 0.102993
exang       -0.925549   0.506698  -1.827 0.067756 .
```

```
oldpeak     -0.509359   0.239591  -2.126 0.033508 *
slope        0.738578   0.392635   1.881 0.059961 .
ca          -0.880927   0.228038  -3.863 0.000112 ***
thal        -1.091852   0.348072  -3.137 0.001708 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 314.32  on 227  degrees of freedom
Residual deviance: 150.41  on 214  degrees of freedom
AIC: 178.41

Number of Fisher Scoring iterations: 6

> datasetlog1=glm(target~age+sex+cp+trestbps+chol+restecg+thalach+exang+ol
dpeak+slope+ca+thal,
+             data=qualityTrain, family='binomial')
> summary(datasetlog1)

Call:
glm(formula = target ~ age + sex + cp + trestbps + chol + restecg +
    thalach + exang + oldpeak + slope + ca + thal, family = "binomial",
    data = qualityTrain)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.3317  -0.3307   0.1232   0.5155   2.6678

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  5.207927   3.072849   1.695 0.090109 .
age         -0.015967   0.026605  -0.600 0.548392
sex         -2.360711   0.607239  -3.888 0.000101 ***
cp           0.917015   0.221612   4.138 3.50e-05 ***
trestbps    -0.019032   0.011676  -1.630 0.103104
chol        -0.006016   0.004269  -1.409 0.158790
restecg      0.562185   0.424127   1.326 0.185002
thalach      0.020623   0.012733   1.620 0.105319
exang       -0.943250   0.502800  -1.876 0.060656 .
oldpeak     -0.495301   0.237468  -2.086 0.037001 *
slope        0.768406   0.390913   1.966 0.049336 *
ca          -0.886937   0.226360  -3.918 8.92e-05 ***
thal        -1.073283   0.346243  -3.100 0.001937 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 314.32  on 227  degrees of freedom
Residual deviance: 150.80  on 215  degrees of freedom
AIC: 176.8

Number of Fisher Scoring iterations: 6

> datasetlog3=glm(target~age+sex+cp+trestbps+restecg+thalach+exang+oldpeak
+slope+ca+thal,
+             data=qualityTrain, family='binomial')
> summary(datasetlog3)

Call:
glm(formula = target ~ age + sex + cp + trestbps + restecg +
    thalach + exang + oldpeak + slope + ca + thal, family = "binomial",
    data = qualityTrain)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
```

```
-2.4195  -0.3653   0.1223   0.5251   2.6501

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  4.28532    2.95958   1.448 0.147631
age         -0.02253    0.02585  -0.871 0.383561
sex         -2.11883    0.55796  -3.797 0.000146 ***
cp           0.91274    0.21949   4.158  3.2e-05 ***
trestbps    -0.01862    0.01166  -1.598 0.110122
restecg      0.63833    0.41409   1.542 0.123186
thalach      0.01806    0.01239   1.458 0.144866
exang       -0.95243    0.49191  -1.936 0.052847 .
oldpeak     -0.51976    0.23603  -2.202 0.027656 *
slope        0.76778    0.39011   1.968 0.049053 *
ca          -0.85389    0.22101  -3.864 0.000112 ***
thal        -1.10337    0.34328  -3.214 0.001308 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 314.32  on 227  degrees of freedom
Residual deviance: 152.76  on 216  degrees of freedom
AIC: 176.76

Number of Fisher Scoring iterations: 6
```

A lot of variables are not significant. Now we will remove Variables based on Significance Level using the backward method

```
> datasetlog4=glm(target~age+sex+cp+trestbps+restecg+exang+oldpeak+slope+ca+thal,
+             data=qualityTrain, family='binomial')
> summary(datasetlog4)

Call:
glm(formula = target ~ age + sex + cp + trestbps + restecg +
    exang + oldpeak + slope + ca + thal, family = "binomial",
    data = qualityTrain)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.4269  -0.3482   0.1167   0.5282   2.5984

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  7.30437    2.23134   3.274  0.00106 **
age         -0.03874    0.02343  -1.653  0.09828 .
sex         -2.12459    0.55250  -3.845  0.00012 ***
cp           0.97817    0.21780   4.491 7.09e-06 ***
trestbps    -0.01614    0.01164  -1.387  0.16542
restecg      0.67557    0.41025   1.647  0.09961 .
exang       -1.11182    0.47653  -2.333  0.01964 *
oldpeak     -0.55467    0.23599  -2.350  0.01875 *
slope        0.84468    0.38138   2.215  0.02677 *
ca          -0.86649    0.21976  -3.943 8.05e-05 ***
thal        -1.02236    0.33510  -3.051  0.00228 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

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

```
    Null deviance: 314.32  on 227  degrees of freedom
Residual deviance: 154.99  on 217  degrees of freedom
AIC: 176.99

Number of Fisher Scoring iterations: 6
```

```
> datasetlog5=glm(target~age+sex+cp+restecg+exang+oldpeak+slope+ca+thal,
+              data=qualityTrain, family='binomial')
> summary(datasetlog5)

Call:
glm(formula = target ~ age + sex + cp + restecg + exang + oldpeak +
    slope + ca + thal, family = "binomial", data = qualityTrain)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.5527  -0.4072   0.1336   0.5433   2.6225

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  5.48831    1.75740   3.123 0.001790 **
age         -0.04362    0.02313  -1.886 0.059339 .
sex         -2.06532    0.53747  -3.843 0.000122 ***
cp           0.93682    0.21316   4.395 1.11e-05 ***
restecg      0.68657    0.40741   1.685 0.091954 .
exang       -1.10296    0.46780  -2.358 0.018385 *
oldpeak     -0.56282    0.23523  -2.393 0.016726 *
slope        0.79278    0.37839   2.095 0.036157 *
ca          -0.86262    0.21874  -3.944 8.03e-05 ***
thal        -1.01325    0.32996  -3.071 0.002135 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 314.32  on 227  degrees of freedom
Residual deviance: 156.96  on 218  degrees of freedom
AIC: 176.96

Number of Fisher Scoring iterations: 6
```

Applying Model after removing least significant Variables. A general rule in machine learning is that the more features you have, the more likely your model will suffer from overfitting.

Train Accuracy with logistic regression when threshold 0.5 = 0.8508

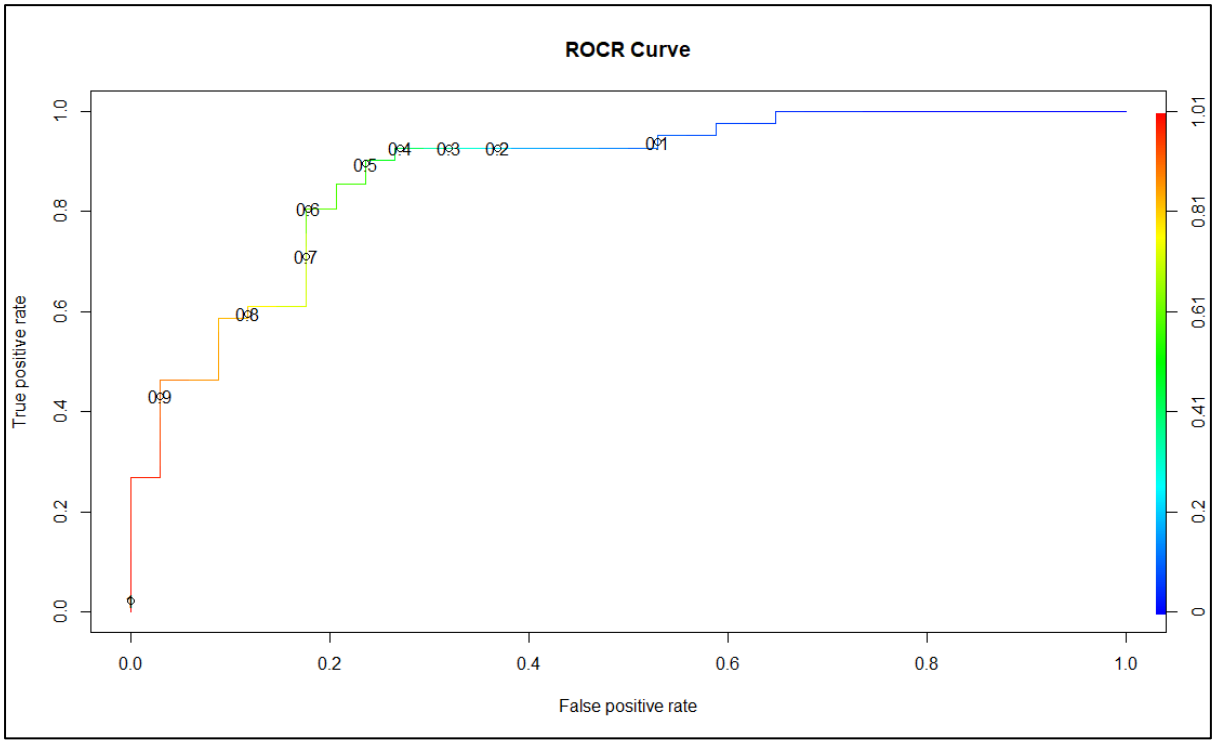
```
> predictTrain = predict(datasetlog5, type='response')
>
>
> table(qualityTrain$target, predictTrain>0.5)

    FALSE TRUE
  0    84   20
  1    14  110
>
> (84+110)/nrow(qualityTrain)
```

```
[1] 0.8508772
```

Test Accuracy with Logistic Regression when threshold 0.5 = 0.82667.

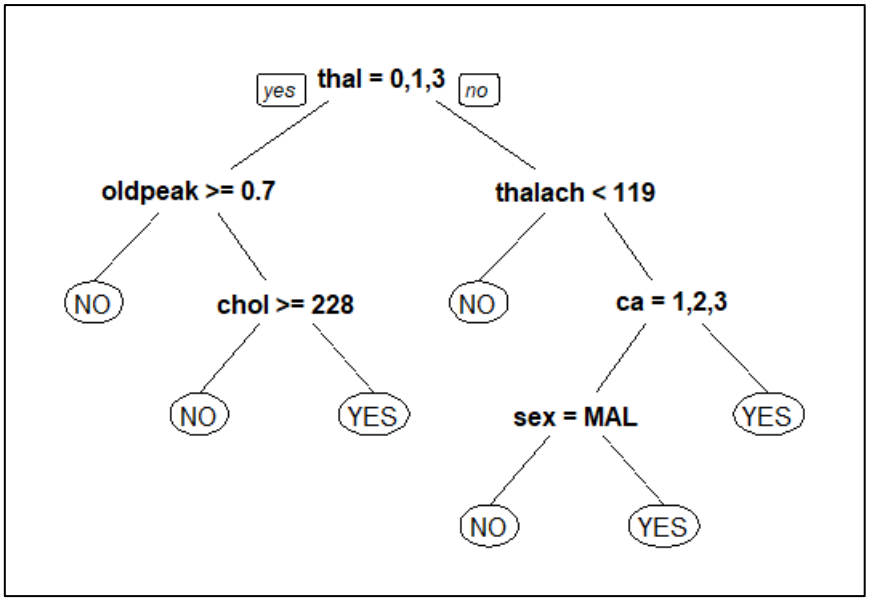
```
> predictTest = predict(datasetlog5, newdata=qualityTest, type='response')
>
> table(qualityTest$target, predictTrain>0.5)
> table(qualityTest$target, predictTest>0.5)

    FALSE TRUE
  0    26    8
  1     5   36
>
> (26+36)/nrow(qualityTest)
[1] 0.8266667
```

```
> ROCRpred = prediction(predictTest, qualityTest$target)
> ROCRperf = performance(ROCRpred, 'tpr','fpr')
plot(ROCRperf,colorize=TRUE, print.cutoffs.at=seq(0.1,by=0.1), main="ROCR
Curve")
```



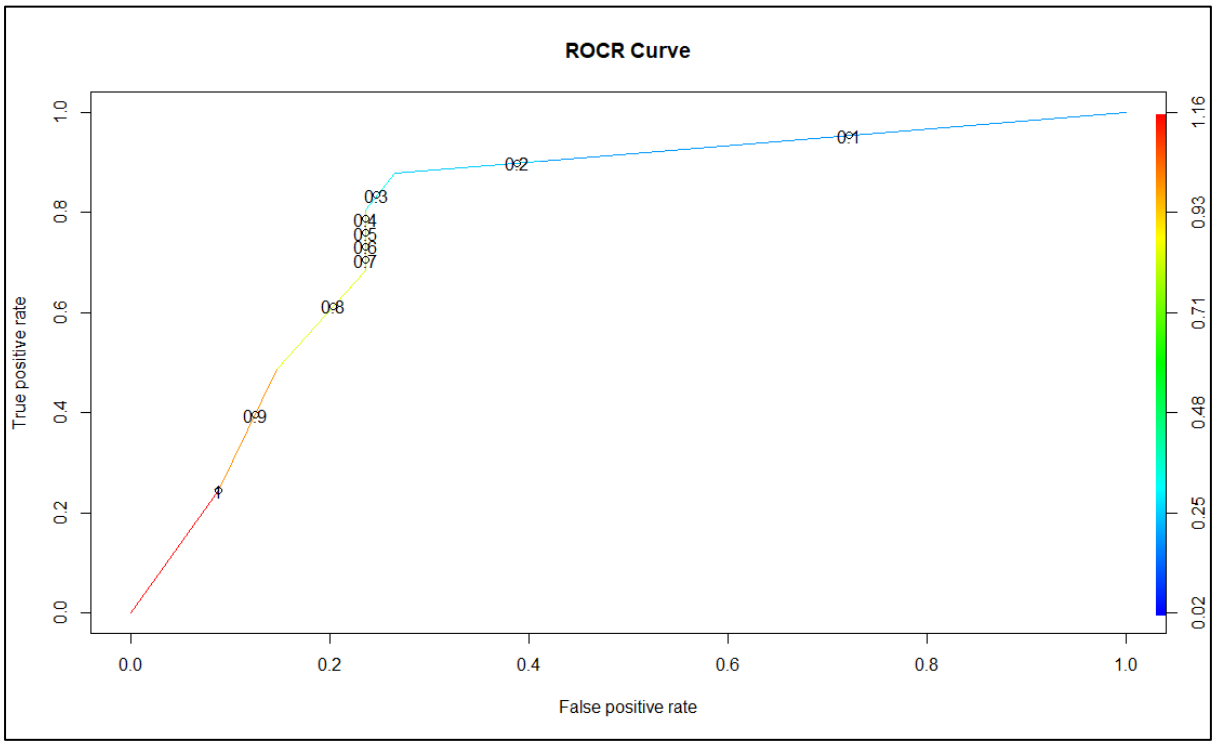
By using Decision Trees:

```
> library(rpart)
> library(rpart.plot)
> tree = rpart(target~., data=qualityTrain, method='class')
>
> prp(tree)
```




```
> predictTree = predict(tree, newdata=qualityTest, type='class')
>
```

```
> ROCRtree = prediction(predictTree[,2],qualityTest$target)
> ROCRperf = performance(ROCRtree, 'tpr','fpr')
> plot(ROCRperf,colorize=TRUE)
```



```
> table(Actual=qualityTest$target, Predicted=predictTree[,2]>0.2)
      Predicted
Actual FALSE TRUE
     0    25    9
     1     5   36
> table(Actual=qualityTest$target, Predicted=predictTree[,2]>0.3)
      Predicted
Actual FALSE TRUE
     0    26    8
     1     8   33
```

Train Accuracy = 0.8793

```
> table(Actual=qualityTrain$target, Predicted=predictTree)
      Predicted
Actual   0   1
     0  89  15
     1  13 111
>
> (89+115)/(89+115+13+15)
[1] 0.8793103
>
```

Test Accuracy = 0.8133

```
> table(Actual=qualityTest$target, Predicted=predictTree[,2]>0.2)
      Predicted
Actual FALSE TRUE
     0    25    9
     1     5   36
> (25+36)/(25+36+9+5)
[1] 0.8133333
>
```

Conclusion:

1. Number of patients getting heart disease are more than number of patience with NO heart disease
2. Dataset contains more number of people with age greater than 50
3. By looking at correlation plots, we can interpret that variables are not well correlated with each other.
4. Train Accuracy with logistic regression when threshold 0.5 = 0.8508

5. Test Accuracy with Logistic Regression when threshold 0.5 = 0.82667.
6. Train accuracy with decision tree=0.8793
7. Test accuracy with decision tree when threshold 0.2= 0.8133