R Lab Mini Project - Heart Disease Prediction

Ankita Mahadik, Shambhavi Milgir, Janvi Patel

19/05/2021

Topic Introduction

Heart disease (HD) is one of the most common diseases nowadays, and an early diagnosis of such a disease is a crucial task for many health care providers to prevent their patients from such a disease and to save lives. An early diagnosis of such disease has been sought for many years, and many data analytics tools have been applied to help health care providers to identify some of the early signs of HD. Many tests can be performed on potential patients to take the extra precautions, measures to reduce the effect of having such a disease, and reliable methods to predict early stages of HD, which can be a crucial task for saving lives. Health care industries generate enormous amounts of data, so called big data that accommodates hidden knowledge or pattern for decision making. The huge volume of data is used to make decisions which are usually more accurate than intuition.

Purpose of selecting this Topic

Analytics is an essential technique for any profession as it forecasts the future and hidden pattern. Data analytics is considered as a cost effective technology in the recent past and it plays an essential role in healthcare which includes new research findings, emergency situations and outbreaks of disease. The use of analytics in healthcare improves care by facilitating preventive care. Exploratory Data Analysis (EDA) is a pre-processing step to understand the data. There are numerous methods and steps in performing EDA, however, most of them are specific, focusing on either visualization or distribution, and are incomplete. Therefore, in this project, we try to perform some visualizations, correlation analysis and regression to predict heart disease risk.

Dataset

Heart Disease is a data set available in UCI repository as well as can be downloaded from Kaggle.

There are 14 features(Columns) including the target as explained below:

1. Age : It is a continuous data type which describes the age of the person in years.
2. Sex: It is a discrete data type that describes the gender of the person. Here 0 = Female and 1 = Male
3. CP(Chest Pain type): It is a discrete data type that describes the chest pain type with following parameters- 1 = Typical angina; 2 = Atypical angina; 3 = Non-anginal pain ; 4 = Asymptotic

4. Trestbps : It is a continuous data type which describes resting blood pressure in mm Hg
5. Cholesterol: It is a continuous data type that describes the serum cholesterol in mg/dl
6. FBS: It is a discrete data type that compares the fasting blood sugar of the person with 120 mg/dl. If FBS >120 then 1 = true else 0 = false
7. RestECG: It is a discrete data type that shows the resting ECG results where 0 = normal; 1 = ST-T wave abnormality; 2 = left ventricular hypertrophy
8. Thalach: It is a continuous data type that describes the max heart rate achieved.
9. Exang: It is a discrete data type where exercise induced angina is shown by 1 = Yes and 0 = No
10. Oldpeak: It is a continuous data type that shows the depression induced by exercise relative to weight
11. Slope: It is a discrete data type that shows us the slope of the peak exercise segment where 1= up-sloping; 2 = flat; 3 = down-sloping
12. ca: It is a continuous data type that shows us the number of major vessels colored by fluoroscopy that ranges from 0 to 3.
13. Thal: It is a discrete data type that shows us Thalassemia where 3 = normal ; 6 = fixed defect ; 7 = reversible defect.
14. Class: It is a discrete data type where diagnose class 0 = No Presence and 1 -4 is range for the person to have the heart disease from least likely to most likely, 1 being least likely.

Algorithm

1. First, all the necessary libraries (such as tidyverse, dplyr, ggplot2, caTools) are loaded.
2. Then, the current working directory is identified.
3. The dataset is then loaded from a csv file, and the glimpse() function is used to get a rough idea about the dataset.
4. The next step is data transformation. In this step, data is mutated, ie, string names are being assigned to binary values. Also, dplyr library is used for data manipulation.
5. The modified data is then visualised with the help of ggplot2 library. Various functions such as barplot, boxplot, etc. and comparisons have been used to visualise the dataset.
6. Once the data has been visualised, correlation analysis is performed on numeric attributes and a correlation graph is plotted.
7. Logistic regression is used to create a model where thalach, age, sex are used to predict target attribute and its summary is then analyzed.
8. The probability of predictions is obtained, and a decision rule is set such that beyond a certain value, the person is at a high risk of developing a heart disease, and below that value, the person has a relatively lower risk of developing a heart disease.
9. Sample data is used to test the model and predictions are made for the data.
10. Various predictions are made with respect to factors such as Thalmium test and Age, and the model is evaluated.
11. Accuracy and classification error of the model is specified, and a confusion matrix is also generated.

Steps performed/Procedure

LOADING LIBRARIES AND DATASET

1. Loading necessary Libraries

```
library(tidyverse)
library(dplyr)
library(ggplot2)
library(caTools)
```

2. Printing the working directory

```
print(getwd())
```

```
## [1] "C:/Users/HP/Documents/R PROG LAB EXP"
```

3. Load the dataset from the working directory and view top and bottom entries

```
data_heart <- read.csv("C:/Users/HP/Documents/heart.csv")

print("Top 5 rows -")
```

```
## [1] "Top 5 rows -"
```

```
head(data_heart, n=5)
```

```
##   ï..age sex cp trestbps chol fbs restecg thalach exang oldpeak slope ca
thal
## 1     63   1  3      145  233   1       0     150     0     2.3     0  0
1
## 2     37   1  2      130  250   0       1     187     0     3.5     0  0
2
## 3     41   0  1      130  204   0       0     172     0     1.4     2  0
2
## 4     56   1  1      120  236   0       1     178     0     0.8     2  0
2
## 5     57   0  0      120  354   0       1     163     1     0.6     2  0
2
##   target
## 1      1
## 2      1
## 3      1
## 4      1
## 5      1
```

```
print("\nBottom 5 rows -")
```

```
## [1] "\nBottom 5 rows -"
```

```
tail(data_heart, n=5)
```

```
##     ï..age sex cp trestbps chol fbs restecg thalach exang oldpeak slope ca
thal
```

```
## 299     57   0  0      140  241   0       1     123     1     0.2     1  0
3
## 300     45   1  3      110  264   0       1     132     0     1.2     1  0
3
## 301     68   1  0      144  193   1       1     141     0     3.4     1  2
3
## 302     57   1  0      130  131   0       1     115     1     1.2     1  1
3
## 303     57   0  1      130  236   0       0     174     0     0.0     1  1
2
##     target
## 299      0
## 300      0
## 301      0
## 302      0
## 303      0
```

4. To get a rough idea about the dataset, we can use glimpse function

```
glimpse(data_heart)
```

```
## Rows: 303
## Columns: 14
## $ ï..age   <int> 63, 37, 41, 56, 57, 57, 56, 44, 52, 57, 54, 48, 49, 64,
58, 5~
## $ sex      <int> 1, 1, 0, 1, 0, 1, 0, 1, 1, 1, 1, 0, 1, 1, 0, 0, 0, 0, 1,
0, 1~
## $ cp       <int> 3, 2, 1, 1, 0, 0, 1, 1, 2, 2, 0, 2, 1, 3, 3, 2, 2, 3, 0,
3, 0~
## $ trestbps <int> 145, 130, 130, 120, 120, 140, 140, 120, 172, 150, 140,
130, 1~
## $ chol     <int> 233, 250, 204, 236, 354, 192, 294, 263, 199, 168, 239,
275, 2~
## $ fbs      <int> 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0,
0, 0~
## $ restecg  <int> 0, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1,
1, 1~
## $ thalach  <int> 150, 187, 172, 178, 163, 148, 153, 173, 162, 174, 160,
139, 1~
## $ exang    <int> 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0,
0, 0~
## $ oldpeak  <dbl> 2.3, 3.5, 1.4, 0.8, 0.6, 0.4, 1.3, 0.0, 0.5, 1.6, 1.2,
0.2, 0~
## $ slope    <int> 0, 0, 2, 2, 2, 1, 1, 2, 2, 2, 2, 2, 2, 1, 2, 1, 2, 0, 2,
2, 1~
## $ ca       <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
2, 0~
## $ thal     <int> 1, 2, 2, 2, 2, 1, 2, 3, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
2, 3~
## $ target   <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1~
```

5. Looking closely at the number of rows and columns

```
cat("No. of Columns = " , ncol(data_heart))
```

```
## No. of Columns =  14
```

```
cat("No. of Rows = " , nrow(data_heart))
```

```
## No. of Rows =  303
```

6. Printing the column names

```
print(colnames(data_heart))
```

```
##  [1] "ï..age"   "sex"      "cp"       "trestbps" "chol"     "fbs"
##  [7] "restecg"  "thalach"  "exang"    "oldpeak"  "slope"    "ca"
## [13] "thal"     "target"
```

7. We can obtain the summary of dataset and get basic information columnwise.

```
summary(data_heart)
```

```
##      ï..age           sex               cp           trestbps
##  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
##       chol            fbs            restecg          thalach
##  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
##      exang           oldpeak         slope             ca
##  Min.   :0.0000   Min.   :0.00   Min.   :0.000   Min.   :0.0000
##  1st Qu.:0.0000   1st Qu.:0.00   1st Qu.:1.000   1st Qu.:0.0000
##  Median :0.0000   Median :0.80   Median :1.000   Median :0.0000
##  Mean   :0.3267   Mean   :1.04   Mean   :1.399   Mean   :0.7294
##  3rd Qu.:1.0000   3rd Qu.:1.60   3rd Qu.:2.000   3rd Qu.:1.0000
##  Max.   :1.0000   Max.   :6.20   Max.   :2.000   Max.   :4.0000
##       thal           target
##  Min.   :0.000   Min.   :0.0000
##  1st Qu.:2.000   1st Qu.:0.0000
##  Median :2.000   Median :1.0000
##  Mean   :2.314   Mean   :0.5446
##  3rd Qu.:3.000   3rd Qu.:1.0000
##  Max.   :3.000   Max.   :1.0000
```

DATA TRANSFORMATION

1. As seen in the dataset, column age has some symbols in it. Let's change name of age column and make it easy to handle.

```
names(data_heart)[1] <- "age"
colnames(data_heart)
```

```
##  [1] "age"      "sex"      "cp"       "trestbps" "chol"     "fbs"
##  [7] "restecg"  "thalach"  "exang"    "oldpeak"  "slope"    "ca"
## [13] "thal"     "target"
```

2. Modify the dataset for better visualization. For this purpose, we use mutate function to modify the column entries for better understanding. Eg - in sex column, 1 will be replaced with 'MALE' and 0 with 'FEMALE'.

```
df <- data_heart %>%
  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, "ABNORMAL", "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,
everything())
```

3. Check the modified dataset

```
head(df)
```

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

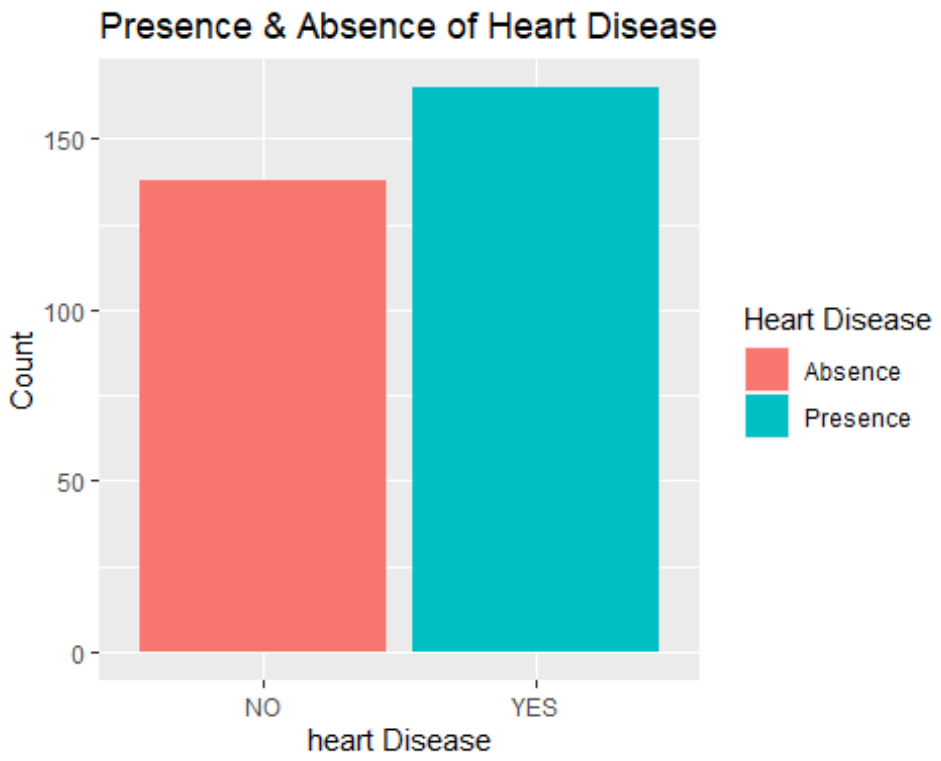
DATA VISUALIZATION

1. Our main focus is on target attribute as it will help in prediction. So to get frequency table for this attribute we can use table function.

```
table(df$target)

##
##  NO YES
## 138 165
```

2. For better understanding, we can visualize the column using ggplot to plot a bar graph.

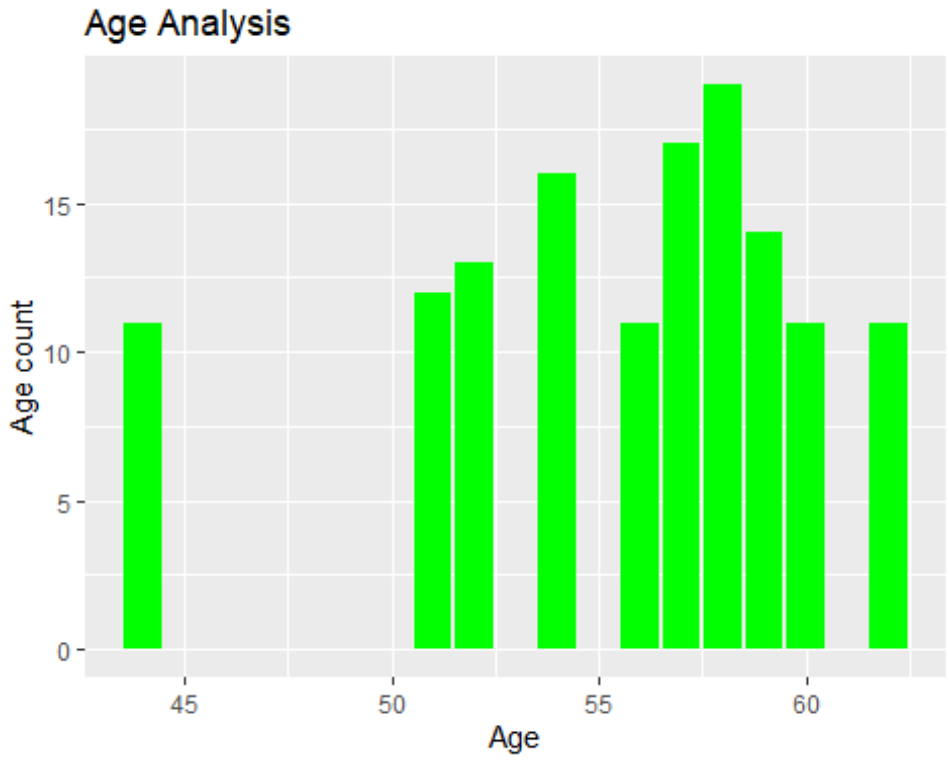
```
library(ggplot2)

ggplot(df, aes(x=df$target, fill=df$target))+
  geom_bar()+
  xlab("heart Disease")+
  ylab("Count")+
  ggtitle("Presence & Absence of Heart Disease")+
  scale_fill_discrete(name="Heart Disease", labels=c("Absence", "Presence"))
```



3. We can use ggplot to plot the count of a particular column vs the target column thereby performing age analysisas shown below -

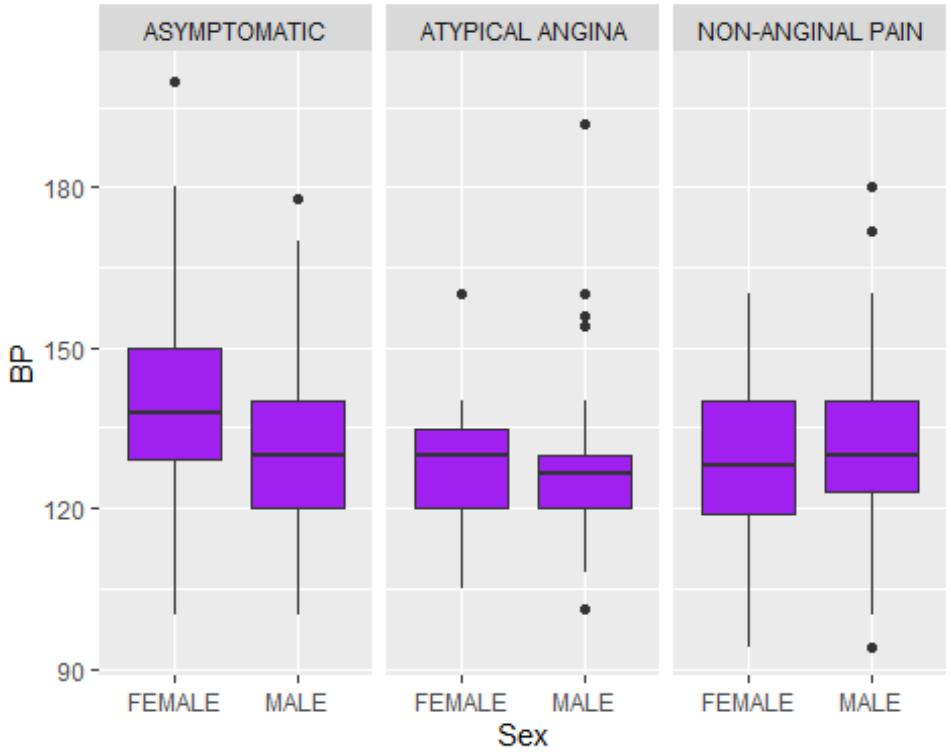
```
df %>%
  group_by(age) %>%
  count() %>%
  filter(n>10) %>% #these people are at higher risk
  ggplot()+
```

```
  geom_col(aes(age, n), fill="green")+
  ggtitle("Age Analysis")+
  xlab("Age")+
  ylab("Age count")
```



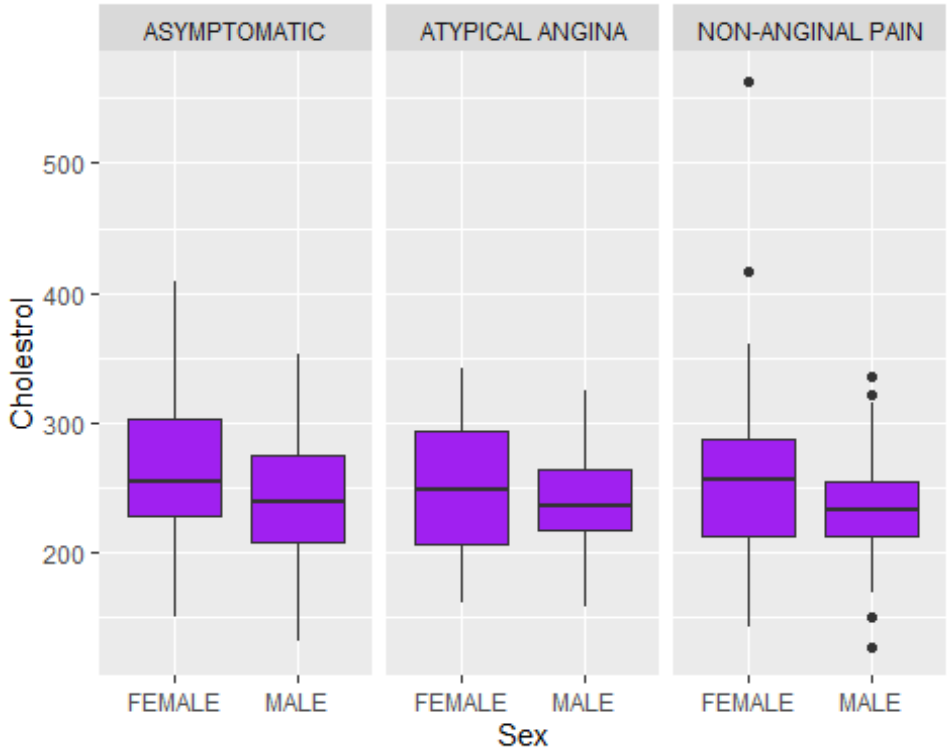
4. Next, we compare blood pressure with chest pain wrt sex attribute using ggplot and facet grid.

```
ggplot(df, aes(x=sex, y=trestbps))+
  geom_boxplot(fill="purple")+
  xlab("Sex")+
  ylab("BP")+
  facet_grid(~cp)
```



5. Using similar approach as above, we can also compare cholestrol with chest pain wrt sex attribute.

```
ggplot(df, aes(x=sex, y=chol))+
  geom_boxplot(fill="purple")+
  xlab("Sex")+
  ylab("Cholestrol")+
  facet_grid(~cp)
```



Similarly, we can perform multiple visualizations using ggplot to gain insights about which attributes are related with the target attribute. For more accurate results about relationship between attributes, we perform 'Correlation Analysis'.

CORRELATION ANALYSIS

1. The very first step will be loading the corrplot library which helps to visualize the correlation.

```
library(corrplot)
```

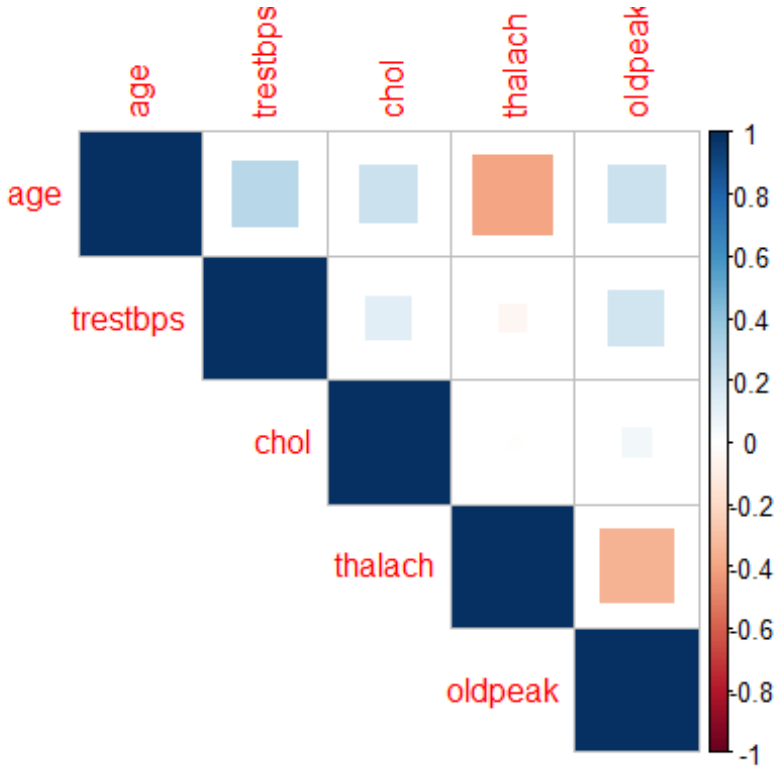
2. We have considered only numeric columns for performing correlation analysis i.e. age, tresbps, chol, thalach and oldpeak columns.

```
cor1 <- cor(df[,10:14])
print(cor1)
```

```
##                 age    trestbps         chol      thalach     oldpeak
## 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
```

3. To visualize the correlation we can we the command given below -

```
corrplot(cor1, method = "square", type = "upper")
```



The method and type can be changed to circle and lower as per liking. From the visualization we can see that age is related to tresbps, chol and oldpeak. Similarly, we can consider other attributes as well. Further, we use Logistic Regression for creating a model and obtaining predictions.

LOGISTIC REGRESSION

1. To create LR model, use the command given below.

```
Model = glm(target ~ age + sex + thalach, data = df,
            family = "binomial")
```

We have fitted a Logistic Regression model here since there are two predicting variables and one binary outcome variable. This model will help us determine the effect that a max heart rate (thalach), age and sex can have on the likelihood that an individual will have a heart disease.

2. Extracting the summary of the model

```
summary(Model)
```

```
##
## Call:
## glm(formula = target ~ age + sex + thalach, family = "binomial",
##     data = df)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.1488  -0.9009   0.4462   0.8358   2.2386
```

```
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -3.187349   1.616480  -1.972   0.0486 *
## age         -0.031774   0.016472  -1.929   0.0537 .
## sexMALE     -1.545892   0.311924  -4.956 7.20e-07 ***
## thalach      0.041393   0.007135   5.802 6.57e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 417.64  on 302  degrees of freedom
## Residual deviance: 329.91  on 299  degrees of freedom
## AIC: 337.91
##
## Number of Fisher Scoring iterations: 4
```

This logistic regression model can be used to predict the probability of a person having heart disease given his/her age, sex and max heart rate.

3. We can translate the predicted probability into a decision rule for clinical use by defining cutoff value on the probability scale. For instance- if a 45 year old female patient with a max heart rate = 150 walks in, we can find out the predicted probability of the heart disease by creating a new data frame called newdata.

```
#get probability of predictions
probability = predict(Model, df, type = "response")

#decision rule definition
df$pred = if_else(probability>=0.5, "Higher risk of HD",
                  "Lower risk of HD")

#sample data to test Model
sample = data.frame(age=55, sex="MALE", thalach=150)

#prediction for sample data
p_new = predict(Model, sample, type = "response")
p_new
```

```
##         1
## 0.4324471
```

We can see that the model generated a heart disease probability of 0.4324 for a 55 year old male with a max heart rate of 150 which indicates a low risk of heart disease.

4. The dataset will also be modified as per decision rule defined above.

```
head(df)
```

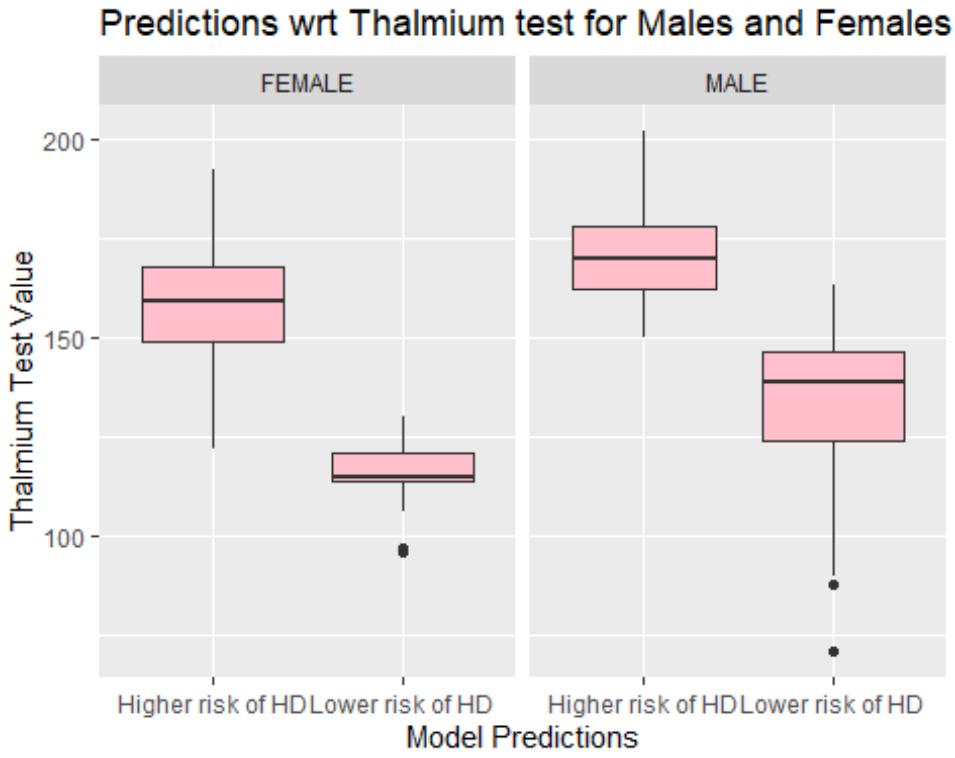
```
##   target    sex   fbs exang               cp  restecg slope ca thal age
## 1    YES   MALE  >120    NO     ASYMPTOMATIC   NORMAL     0  0    1  63
```

```
## 2    YES   MALE <=120    NO NON-ANGINAL PAIN ABNORMAL     0  0    2  37
## 3    YES FEMALE <=120    NO  ATYPICAL ANGINA   NORMAL     2  0    2  41
## 4    YES   MALE <=120    NO  ATYPICAL ANGINA ABNORMAL     2  0    2  56
## 5    YES FEMALE <=120   YES     ASYMPTOMATIC ABNORMAL     2  0    2  57
## 6    YES   MALE <=120    NO     ASYMPTOMATIC ABNORMAL     1  0    1  57
##   trestbps chol thalach oldpeak              pred
## 1      145  233     150     2.3  Lower risk of HD
## 2      130  250     187     3.5 Higher risk of HD
## 3      130  204     172     1.4 Higher risk of HD
## 4      120  236     178     0.8 Higher risk of HD
## 5      120  354     163     0.6 Higher risk of HD
## 6      140  192     148     0.4  Lower risk of HD
```

PREDICTION VISUALIZATIONS

1. Plot - 1 : Predictions wrt Thalmium test for Males and Females using ggplot

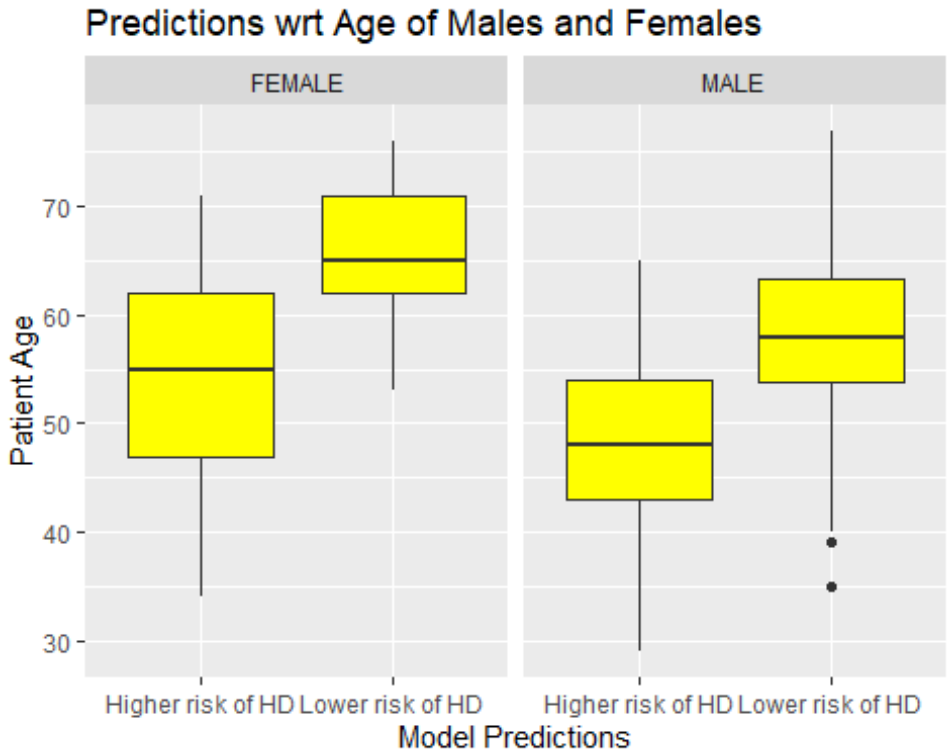
```
ggplot(df, aes(pred, thalach)) +
  geom_boxplot(fill="pink") +
  ggtitle("Predictions wrt Thalmium test for Males and Females")+
  xlab("Model Predictions")+
  ylab("Thalmium Test Value")+
  facet_grid(~sex)
```



2. Plot - 2 : Predictions wrt Age of Males and Females usinh ggplot

```
#we can replace thalach with age

ggplot(df, aes(pred, age)) +
```

```
  geom_boxplot(fill="yellow") +
  ggtitle("Predictions wrt Age of Males and Females")+
  xlab("Model Predictions")+
  ylab("Patient Age")+
  facet_grid(~sex)
```



EVALUATING THE MODEL

While these predictive models can be used to predict the probability of an event occurring, it is vital to check the accuracy of any model before computing the predicted values. Some of the core metrics that can be used to evaluate this model are as described below-

- Accuracy : It is one of the most straightforward metric which tells us the proportion of total number of predictions being correct
- Classification Error Rate : This can be calculated using 1-Accuracy
- Area under the ROC curve (AUC): This is one of the most sought after metrics used for evaluation. It is popular since it is independent of the change in proportion of responders. It ranges from 0–1. The closer it gets to 1, the better is the model performance
- Confusion Matrix: It is a N*N matrix where N is the level of outcome.This metric reports the the number of false positives, false negatives, true positives, and true negatives.

We require Metrics library for evaluating the model.

```
library(Metrics)
```

```
## Warning: package 'Metrics' was built under R version 4.0.5
```

Before evaluating the model, we need to change target atribute entries as per pred values since both must have similar type of entries.

```
#modifying target variable as per pred values
df$target = if_else(df$target=="YES", "Higher risk of HD", "Lower risk of HD")
```

Model Evaluation is done using the following commands -

```
#calculating Accuracy, Classification error
acc = accuracy(df$target, df$pred)
class_err = ce(df$target, df$pred)

print(paste("Accuracy = ",acc))

## [1] "Accuracy =  0.71947194719472"

print(paste("Classification error = ",class_err))

## [1] "Classification error =  0.280528052805281"

#Confusion Matrix
table(df$target, df$pred, dnn = c("True Status", "Predicted Status"))

##                     Predicted Status
## True Status          Higher risk of HD Lower risk of HD
##   Higher risk of HD               125               40
##   Lower risk of HD                 45               93
```

Results and conclusions

From the above output, we can see that the model has an overall accuracy of 0.71. Also, there are cases that were mis-classified as shown in the confusion matrix. We can improve the existing model by including other relevant predictors from the dataset into our model. Also, we can conclude that age, heart rate and sex are important factors to be considered while predicting heart disease risk.

References

[1] Predicting Heart Disease Using Regression Analysis, Sailee Mene, Accessed at : 10th May 2021, Available at: https://medium.com/swlh/predicting-heart-disease-using-regression-analysis-486401cd0a47

[2] Heart Disease Prediction Dataset Kaggle, Accessed at : 6th May 2021, Available at: https://www.kaggle.com/priyanka841/heart-disease-prediction-uci