# HarvardX: PH125.9x Data Science
# Heart Disease Prediction

Battula Damodhar

July 28, 2021

## Contents

# 1 Overview

This project is related to the Heart disease prediction of the HervardX: PH125.9x Data Science: Capstone course. The present report start with a general idea of the project and by representing its objective.

Then the given dataset will be prepared and setup. Data cleaning, Visualization and an exploratory data analysis is carried out in order to develop a machine learning model that could predict dataset. Results will be explained. Finally the report ends with some concluding remarks and future work.

## 1.1 Introduction

We are using processed Cleveland Heart Disease dataset which is freely available at https://archive.ics.uci.edu/ml/index.php In this project we are going to predict the heart disease by using this data set. It is a multivariate dataset with 303 instance and 14 attributes with Categorical, Real and integer characteristics. The 14th attribute in the data set is the predicted attribute that we have to use to predict. The predicted attribute field refers to the presence of heart disease in the patient. It is integer valued from 0 (no presence) to 4. Experiments with the Cleveland database have concentrated on simply attempting to distinguish of presence (values 1,2,3,4) from absence (value 0).

- The attribute information of the dataset can be available at :

https://archive.ics.uci.edu/ml/datasets/Heart+Disease

There are 14 variables provided in the data set and the last one is the dependent variable that we want to be able to predict. Here is a summary of what the other variables mean:

| Attribute name | Short description |
|---|---|
| age | Age of patient in Years |
| sex | 1 = male; 0 = female |
| cp | chest pain type (1 = typical angina, 2 = atypical angina, 3 = non-anginal pain, 4 = asymptomatic) |
| trestbps | resting blood pressure (in mm Hg on admission to the hospital) |
| chol | serum cholestoral in mg/dl |
| fbs | fasting blood sugar > 120mg/dl (1 = true; 0 = false) |
| restecg | resting electrocardiographic results (0 = normal; 1 = having ST-T wave abnormality; 2 = showing probable or definite left ventricular hypertrophy by Estes' criteria) |
| thalach | maximum heart rate achieved |
| exang | exercise induced angina (1 = yes; 0 = no) |
| oldpeak | ST depression induced by exercise relative to rest |
| slope | the slope of the peak exercise ST segment (1 = upsloping; 2 = flat; 3 = downsloping) |
| ca | number of major vessel (0-3) colored by flourosopy |
| thal | 3 = normal; 6 = fixed defect; 7 = reversible defect |
| num | diagnosis of heart disease (angiographic disease status) |

One file has been "processed", that one containing the Cleveland data. In this project we are going to use this file as a dataset to predict heart disease.

### 1.2 Aim of the Project

In this project we are dealing with different machine learning algorithms to predict heart disease (angiographic disease status) are compared. For some algorithms, model parameters are tuned and the best model selected. We are going to predict heart disease with different machine learning models to find the best model out off those.

## 2 Dataset downloading and preperation

### 2.1 Used Libraries

```r
#Installing required packages:
if(!require(tidyverse)) install.packages("tidyverse", repos = "http://cran.us.r-project.org")
if(!require(caret)) install.packages("caret", repos = "http://cran.us.r-project.org")
if(!require(data.table)) install.packages("data.table", repos = "http://cran.us.r-project.org")
if(!require(pROC)) install.packages("pROC", repos = "http://cran.us.r-project.org")
if(!require(randomForest)) install.packages("randomForest", repos = "http://cran.us.r-project.org")
if(!require(e1071)) install.packages("e1071", repos = "http://cran.us.r-project.org")
```

### 2.2 Used Dataset

- [Preprocessed Heart disease dataset] https://archive.ics.uci.edu/ml/machine-learning-databases/heart-disease/processed.cleveland.data
- [The description of the dataset can be found] https://archive.ics.uci.edu/ml/datasets/Heart+Disease

```r
#Reading data in CSV format as dataframe
url = 'https://archive.ics.uci.edu/ml/machine-learning-databases/heart-disease/processed.cleveland.data
heart_disease_dataset <- read.csv(file = url, sep = ',', na = '?', header = F)
```

Here, I am using full attribute names instead of abbreviation to avoid confusion.

```r
#Prepare column names
column_names <- c("Age",
                  "Sex",
                  "Chest_Pain_Type",
                  "Resting_Blood_Pressure",
                  "Serum_Cholesterol",
                  "Fasting_Blood_Sugar",
                  "Resting_ECG",
                  "Max_Heart_Rate_Achieved",
                  "Exercise_Induced_Angina",
                  "ST_Depression_Exercise",
                  "Peak_Exercise_ST_Segment",
                  "Num_Major_Vessels_Flouro",
                  "Thalassemia",
                  "Diagnosis_Heart_Disease")

#Apply column names to the dataframe
colnames(heart_disease_dataset) <- column_names
```

# 3 Methods and Analysis

## 3.1 Data visualization

To get familiar with data, let's have a look on general overview of dataset.

```
# Glimpse data
heart_disease_dataset %>% glimpse()
```

```
## Rows: 303
## Columns: 14
## $ Age                      <dbl> 63, 67, 67, 37, 41, 56, 62, 57, 63, 53, 57, 5~
## $ Sex                      <dbl> 1, 1, 1, 1, 0, 1, 0, 0, 1, 1, 1, 0, 1, 1, 1, ~
## $ Chest_Pain_Type          <dbl> 1, 4, 4, 3, 2, 2, 4, 4, 4, 4, 4, 2, 3, 2, 3, ~
## $ Resting_Blood_Pressure   <dbl> 145, 160, 120, 130, 130, 120, 140, 120, 130, ~
## $ Serum_Cholesterol        <dbl> 233, 286, 229, 250, 204, 236, 268, 354, 254, ~
## $ Fasting_Blood_Sugar      <dbl> 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1, ~
## $ Resting_ECG              <dbl> 2, 2, 2, 0, 2, 0, 2, 0, 2, 2, 0, 2, 2, 0, 0, ~
## $ Max_Heart_Rate_Achieved  <dbl> 150, 108, 129, 187, 172, 178, 160, 163, 147, ~
## $ Exercise_Induced_Angina  <dbl> 0, 1, 1, 0, 0, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, ~
## $ ST_Depression_Exercise   <dbl> 2.3, 1.5, 2.6, 3.5, 1.4, 0.8, 3.6, 0.6, 1.4, ~
## $ Peak_Exercise_ST_Segment <dbl> 3, 2, 2, 3, 1, 1, 3, 1, 2, 3, 2, 2, 2, 1, 1, ~
## $ Num_Major_Vessels_Flouro <dbl> 0, 3, 2, 0, 0, 0, 2, 0, 1, 0, 0, 0, 1, 0, 0, ~
## $ Thalassemia              <dbl> 6, 3, 7, 3, 3, 3, 3, 3, 7, 7, 6, 3, 6, 7, 7, ~
## $ Diagnosis_Heart_Disease  <int> 0, 2, 1, 0, 0, 0, 3, 0, 2, 1, 0, 0, 2, 0, 0, ~
```

```
# heading few records of data
heart_disease_dataset %>% head()
```

```
##   Age Sex Chest_Pain_Type Resting_Blood_Pressure Serum_Cholesterol
## 1  63   1               1                    145               233
## 2  67   1               4                    160               286
## 3  67   1               4                    120               229
## 4  37   1               3                    130               250
## 5  41   0               2                    130               204
## 6  56   1               2                    120               236
##   Fasting_Blood_Sugar Resting_ECG Max_Heart_Rate_Achieved
## 1                   1           2                     150
## 2                   0           2                     108
## 3                   0           2                     129
## 4                   0           0                     187
## 5                   0           2                     172
## 6                   0           0                     178
##   Exercise_Induced_Angina ST_Depression_Exercise Peak_Exercise_ST_Segment
## 1                       0                    2.3                        3
## 2                       1                    1.5                        2
## 3                       1                    2.6                        2
## 4                       0                    3.5                        3
## 5                       0                    1.4                        1
## 6                       0                    0.8                        1
##   Num_Major_Vessels_Flouro Thalassemia Diagnosis_Heart_Disease
## 1                        0           6                       0
## 2                            3           3                       2
## 3                            2           7                       1
## 4                            0           3                       0
## 5                            0           3                       0
## 6                            0           3                       0
```

```r
# Summary of data
heart_disease_dataset %>% summary()
```

```
##       Age             Sex          Chest_Pain_Type Resting_Blood_Pressure
##  Min.   :29.00   Min.   :0.0000   Min.   :1.000   Min.   : 94.0
##  1st Qu.:48.00   1st Qu.:0.0000   1st Qu.:3.000   1st Qu.:120.0
##  Median :56.00   Median :1.0000   Median :3.000   Median :130.0
##  Mean   :54.44   Mean   :0.6799   Mean   :3.158   Mean   :131.7
##  3rd Qu.:61.00   3rd Qu.:1.0000   3rd Qu.:4.000   3rd Qu.:140.0
##  Max.   :77.00   Max.   :1.0000   Max.   :4.000   Max.   :200.0
##
##  Serum_Cholesterol Fasting_Blood_Sugar  Resting_ECG     Max_Heart_Rate_Achieved
##  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 :241.0     Median :0.0000      Median :1.0000   Median :153.0
##  Mean   :246.7     Mean   :0.1485      Mean   :0.9901   Mean   :149.6
##  3rd Qu.:275.0     3rd Qu.:0.0000      3rd Qu.:2.0000   3rd Qu.:166.0
##  Max.   :564.0     Max.   :1.0000      Max.   :2.0000   Max.   :202.0
##
##  Exercise_Induced_Angina ST_Depression_Exercise Peak_Exercise_ST_Segment
##  Min.   :0.0000          Min.   :0.00           Min.   :1.000
##  1st Qu.:0.0000          1st Qu.:0.00           1st Qu.:1.000
##  Median :0.0000          Median :0.80           Median :2.000
##  Mean   :0.3267          Mean   :1.04           Mean   :1.601
##  3rd Qu.:1.0000          3rd Qu.:1.60           3rd Qu.:2.000
##  Max.   :1.0000          Max.   :6.20           Max.   :3.000
##
##  Num_Major_Vessels_Flouro  Thalassemia    Diagnosis_Heart_Disease
##  Min.   :0.0000           Min.   :3.000   Min.   :0.0000
##  1st Qu.:0.0000           1st Qu.:3.000   1st Qu.:0.0000
##  Median :0.0000           Median :3.000   Median :0.0000
##  Mean   :0.6722           Mean   :4.734   Mean   :0.9373
##  3rd Qu.:1.0000           3rd Qu.:7.000   3rd Qu.:2.0000
##  Max.   :3.0000           Max.   :7.000   Max.   :4.0000
##  NA's   :4                NA's   :2
```

```r
# Missing values
sum(is.na(heart_disease_dataset))
```

```
## [1] 6
```

There are 6 missing values (NA's) present in the data in which 4 from Num_Major_Vessels_Flouro and 2 from Thalassemia.

Here are some visualizations that provide some insights into the data.

```r
####################################################
# Distinct values
####################################################

heart_disease_dataset %>% summarise(n_age = n_distinct(Age),
                                    n_sex = n_distinct(Sex),
                                    n_cp = n_distinct(Chest_Pain_Type),
                                    n_trestbps = n_distinct(Resting_Blood_Pressure),
                                    n_chol = n_distinct(Serum_Cholesterol),
                                    n_fbs = n_distinct(Fasting_Blood_Sugar),
                                    n_restecg = n_distinct(Resting_ECG),
                                    n_thalach = n_distinct(Max_Heart_Rate_Achieved),
                                    n_exang = n_distinct(Exercise_Induced_Angina),
                                    n_oldpeak = n_distinct(ST_Depression_Exercise),
                                    n_slope = n_distinct(Peak_Exercise_ST_Segment),
                                    n_ca = n_distinct(Num_Major_Vessels_Flouro),
                                    n_thal = n_distinct(Thalassemia),
                                    n_condition = n_distinct(Diagnosis_Heart_Disease))
```

```
##   n_age n_sex n_cp n_trestbps n_chol n_fbs n_restecg n_thalach n_exang
## 1    41     2    4         50    152     2         3        91       2
##   n_oldpeak n_slope n_ca n_thal n_condition
## 1        40       3    5      4           5
```

A closer look at the data identifies some NA values that will need to be addressed in the cleaning step. We also want to know the number of observations in the dependent variable column to understand if the dataset is relatively balanced.

Identify the count of Sex:

```r
heart_disease_dataset %>%
  group_by(Sex) %>%
  count() %>%
  knitr::kable()
```

| Sex | n |
|---:|---:|
| 0 | 97 |
| 1 | 206 |

Identify the different levels of Heart Disease:

```r
#Determine the number of values in each level of dependent variable
heart_disease_dataset %>%
  drop_na() %>%
  group_by(Diagnosis_Heart_Disease) %>%
  count() %>%
  knitr::kable()
```

| Diagnosis_Heart_Disease | n |
|---:|---:|
| 0 | 160 |
| 1 | 54 |
| 2 | 35 |
| 3 | 35 |
| 4 | 13 |

Identify the different levels of Thalassemia:

```r
heart_disease_dataset %>%
  drop_na() %>%
  group_by(Thalassemia) %>%
  count() %>%
  knitr::kable()
```

| Thalassemia | n |
|---:|---:|
| 3 | 164 |
| 6 | 18 |
| 7 | 115 |

## 3.2 Data Pre-processing

Since any value above 0 in 'Diagnosis_Heart_Disease' (column 14) indicates the presence of heart disease, we can make all levels > 0 together as 1, so the classification predictions are binary – Yes or No (1 or 0). The total count of positive heart disease results is less than the number of negative results so the fct_lump() call with default arguments will convert that variable from 4 levels to 2.

The data cleaning pipeline below deals with NA values, converts some variables to factors, lumps the dependent variable into two buckets, removes the rows that had "NA" for observations, and reorders the variables within the dataframe:

```r
#Drop NA's, convert to factors, lump target variable to 2 levels, remove "?", reorder variables
heart_dataset_clean_tbl <- heart_disease_dataset %>%
  drop_na() %>%
  mutate_at(c("Resting_ECG",
              "Fasting_Blood_Sugar",
              "Sex",
              "Diagnosis_Heart_Disease",
              "Exercise_Induced_Angina",
              "Peak_Exercise_ST_Segment",
              "Chest_Pain_Type",
              "Thalassemia"), as_factor) %>%
  mutate(Num_Major_Vessels_Flouro = as.numeric(Num_Major_Vessels_Flouro)) %>%
  mutate(Diagnosis_Heart_Disease = fct_lump(Diagnosis_Heart_Disease, other_level = "1")) %>%
  filter(Thalassemia != "na") %>%
  select(Age,
         Resting_Blood_Pressure,
         Serum_Cholesterol,
         Max_Heart_Rate_Achieved,
         ST_Depression_Exercise,
         Num_Major_Vessels_Flouro,
         everything())

#Glimpse data
heart_dataset_clean_tbl %>% glimpse()
```

```
## Rows: 297
## Columns: 14
## $ Age                      <dbl> 63, 67, 67, 37, 41, 56, 62, 57, 63, 53, 57, 5~
## $ Resting_Blood_Pressure   <dbl> 145, 160, 120, 130, 130, 120, 140, 120, 130, ~
## $ Serum_Cholesterol        <dbl> 233, 286, 229, 250, 204, 236, 268, 354, 254, ~
## $ Max_Heart_Rate_Achieved  <dbl> 150, 108, 129, 187, 172, 178, 160, 163, 147, ~
## $ ST_Depression_Exercise   <dbl> 2.3, 1.5, 2.6, 3.5, 1.4, 0.8, 3.6, 0.6, 1.4, ~
## $ Num_Major_Vessels_Flouro <dbl> 0, 3, 2, 0, 0, 0, 2, 0, 1, 0, 0, 0, 1, 0, 0, ~
## $ Sex                      <fct> 1, 1, 1, 1, 0, 1, 0, 0, 1, 1, 1, 0, 1, 1, 1, ~
## $ Chest_Pain_Type          <fct> 1, 4, 4, 3, 2, 2, 4, 4, 4, 4, 4, 2, 3, 2, 3, ~
## $ Fasting_Blood_Sugar      <fct> 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1, ~
## $ Resting_ECG              <fct> 2, 2, 2, 0, 2, 0, 2, 0, 2, 2, 0, 2, 2, 0, 0, ~
## $ Exercise_Induced_Angina  <fct> 0, 1, 1, 0, 0, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, ~
## $ Peak_Exercise_ST_Segment <fct> 3, 2, 2, 3, 1, 1, 3, 1, 2, 3, 2, 2, 2, 1, 1, ~
## $ Thalassemia              <fct> 6, 3, 7, 3, 3, 3, 3, 3, 7, 7, 6, 3, 6, 7, 7, ~
## $ Diagnosis_Heart_Disease  <fct> 0, 1, 1, 0, 0, 0, 1, 0, 1, 1, 0, 0, 1, 0, 0, ~
```

```r
#Summary
heart_dataset_clean_tbl %>% summary()
```

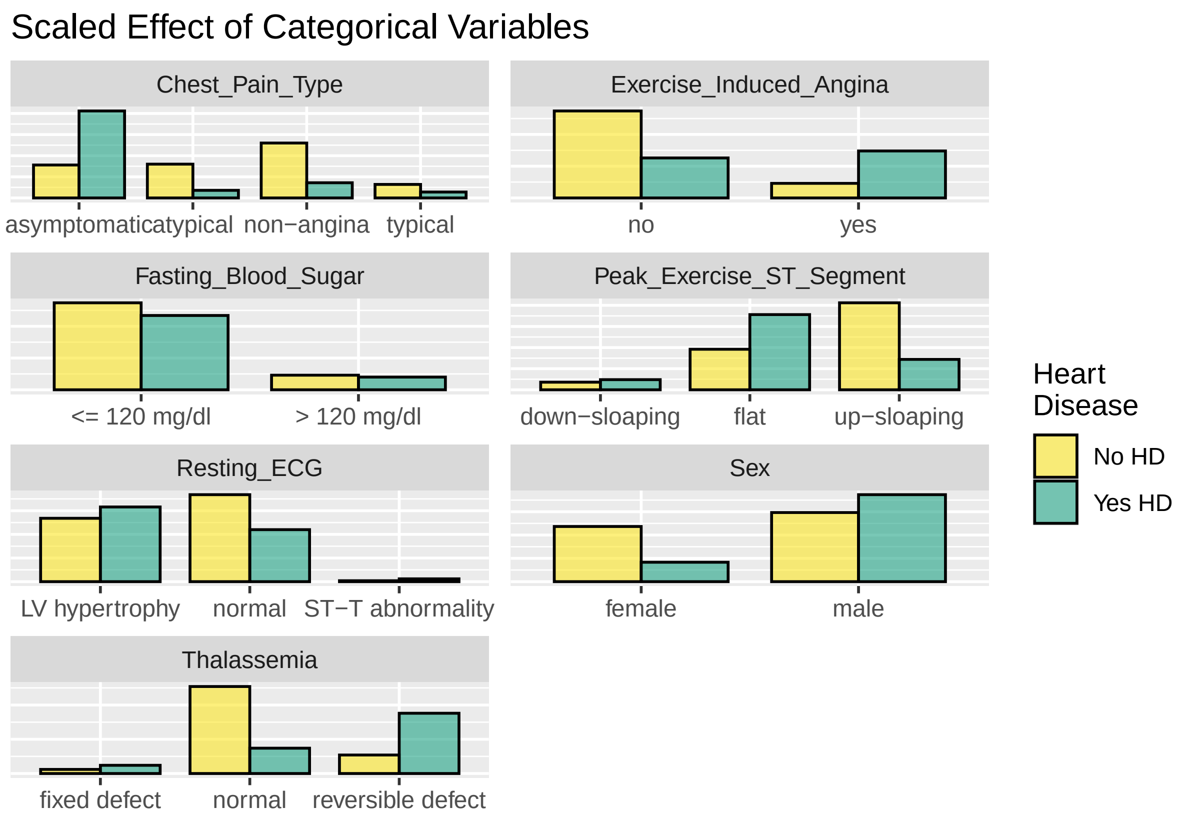
```
##       Age        Resting_Blood_Pressure Serum_Cholesterol
##  Min.   :29.00   Min.   : 94.0          Min.   :126.0    
##  1st Qu.:48.00   1st Qu.:120.0          1st Qu.:211.0    
##  Median :56.00   Median :130.0          Median :243.0    
##  Mean   :54.54   Mean   :131.7          Mean   :247.4    
##  3rd Qu.:61.00   3rd Qu.:140.0          3rd Qu.:276.0    
##  Max.   :77.00   Max.   :200.0          Max.   :564.0    
##  Max_Heart_Rate_Achieved ST_Depression_Exercise Num_Major_Vessels_Flouro
##  Min.   : 71.0           Min.   :0.000          Min.   :0.0000          
##  1st Qu.:133.0           1st Qu.:0.000          1st Qu.:0.0000          
##  Median :153.0           Median :0.800          Median :0.0000          
##  Mean   :149.6           Mean   :1.056          Mean   :0.6768          
##  3rd Qu.:166.0           3rd Qu.:1.600          3rd Qu.:1.0000          
##  Max.   :202.0           Max.   :6.200          Max.   :3.0000          
##  Sex     Chest_Pain_Type Fasting_Blood_Sugar Resting_ECG
##  0: 96   1: 23           0:254               0:147      
##  1:201   2: 49           1: 43               1:  4      
##          3: 83                               2:146      
##          4:142                                          
##                                                         
##                                                         
##  Exercise_Induced_Angina Peak_Exercise_ST_Segment Thalassemia
##  0:200                   1:139                    3:164      
##  1: 97                   2:137                    6: 18      
##                          3: 21                    7:115      
##                                                              
##                                                              
##                                                              
##  Diagnosis_Heart_Disease
##  0:160                  
##  1:137                  
##                         
##                         
##                         
##                         
```

You can see after cleaning data by removing NAs' from dataset there are 96 female and 201 male, available which means 1 record cleansed from female and 5 from male data.

## 3.3 Data Analysis and Exploration

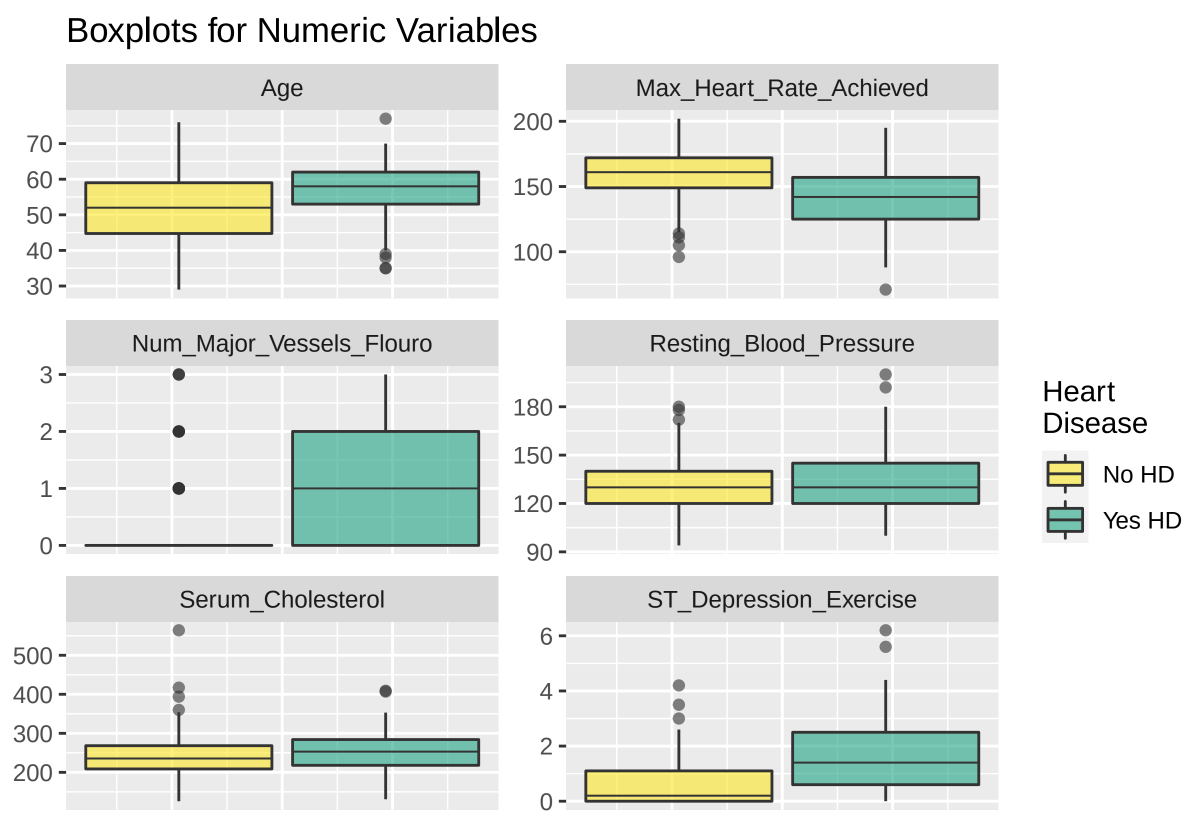
Now, time for some basic exploratory data analysis. The workflow below breaks out the categorical variables and visualizes them on a faceted bar plot. I'm recoding the factors levels from numeric back to text-based so the labels are easy to interpret on the plots and stripping the y-axis labels since the relative differences are what matters.

```r
#Select categorical vars, recode them to their character values, convert to long format
hd_long_fact_tbl <- heart_dataset_clean_tbl  %>%
  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, `1` = "typical",
                                         `2` = "atypical",
                                         `3` = "non-angina",
                                         `4` = "asymptomatic"),
         Fasting_Blood_Sugar = recode_factor(Fasting_Blood_Sugar, `0` = "<= 120 mg/dl",
                                             `1` = "> 120 mg/dl"),
         Resting_ECG = recode_factor(Resting_ECG, `0` = "normal",
                                     `1` = "ST-T abnormality",
                                     `2` = "LV hypertrophy"),
         Exercise_Induced_Angina = recode_factor(Exercise_Induced_Angina, `0` = "no",
                                                 `1` = "yes"),
         Peak_Exercise_ST_Segment = recode_factor(Peak_Exercise_ST_Segment, `1` = "up-sloaping",
                                                  `2` = "flat",
                                                  `3` = "down-sloaping"),
         Thalassemia = recode_factor(Thalassemia, `3` = "normal",
                                     `6` = "fixed defect",
                                     `7` = "reversible defect")) %>%
  gather(key = "key", value = "value", -Diagnosis_Heart_Disease)


#Visualize with bar plot
hd_long_fact_tbl %>%
  ggplot(aes(value)) +
  geom_bar(aes(x        = value,
               fill     = Diagnosis_Heart_Disease),
           alpha    = .6,
           position = "dodge",
           color    = "black",
           width    = .8
  ) +
  labs(x = "",
       y = "",
       title = "Scaled Effect of Categorical Variables") +
  theme(
    axis.text.y  = element_blank(),
    axis.ticks.y = element_blank()) +
  facet_wrap(~ key, scales = "free", nrow = 4) +
  scale_fill_manual(
    values = c("#fde725ff", "#20a486ff"),
    name   = "Heart\nDisease",
    labels = c("No HD", "Yes HD"))
```



Now, evaluating the numeric variables using boxplot.

```r
#Must gather() data first in order to facet wrap by key
#(default gather call puts all var names into new key col)
hd_long_cont_tbl <- heart_dataset_clean_tbl  %>%
  select(Age,
         Resting_Blood_Pressure,
         Serum_Cholesterol,
         Max_Heart_Rate_Achieved,
         ST_Depression_Exercise,
         Num_Major_Vessels_Flouro,
         Diagnosis_Heart_Disease) %>%
  gather(key   = "key",
         value = "value",
         -Diagnosis_Heart_Disease)

#Visualize numeric variables as boxplots
hd_long_cont_tbl %>%
  ggplot(aes(y = value)) +
  geom_boxplot(aes(fill = Diagnosis_Heart_Disease),
               alpha  = .6,
               fatten = .7) +
  labs(x = "",
       y = "",
       title = "Boxplots for Numeric Variables") +
  scale_fill_manual(
    values = c("#fde725ff", "#20a486ff"),
    name   = "Heart\nDisease",
    labels = c("No HD", "Yes HD")) +
  theme(
    axis.text.x  = element_blank(),
    axis.ticks.x = element_blank()) +
  facet_wrap(~ key,
             scales = "free",
             ncol   = 2)
```



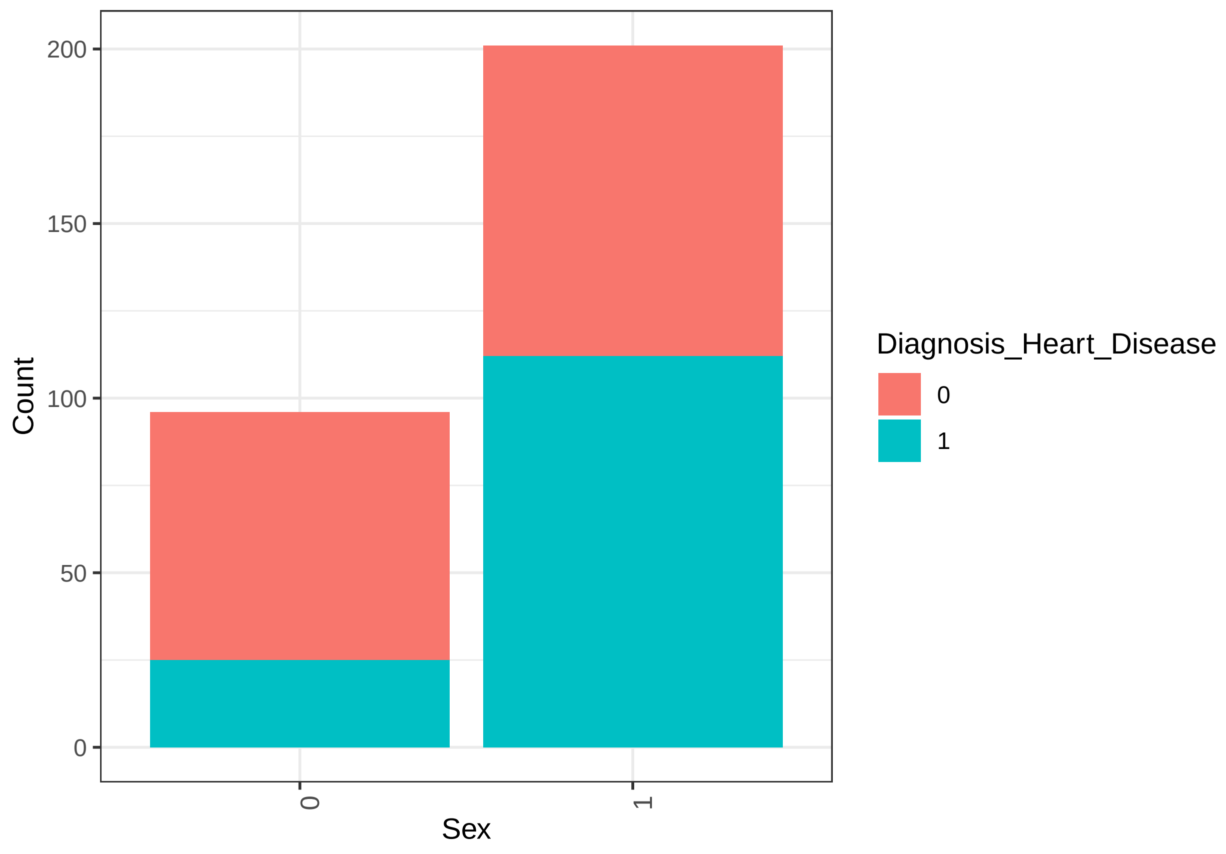
The plots for categorical and numeric variables suggest the following conditions are highly associated with increased prevalence of heart disease.

- Asymptomatic chest pain (relative to typical angina chest pain, atypical angina pain, or non-angina pain)
- Presence of exercise induced angina
- Lower fasting blood sugar
- Flat and down-sloaping peak exercise ST segment
- Presence of left ventricle hypertrophy
- Male
- Higher thelassemia score in reversible defect
- Higher age
- Lower max heart rate achieved
- Higher resting blood pressure
- Higher Serum cholesterol
- Higher ST depression induced by exercise relative to rest

You can see that the Age, blood pressure, cholesterol, and Sex all point in the right direction based on what we generally know about the world around us. This provides a nice phase gate to let us proceed with the analysis.

Heart disease distribution by Sex:

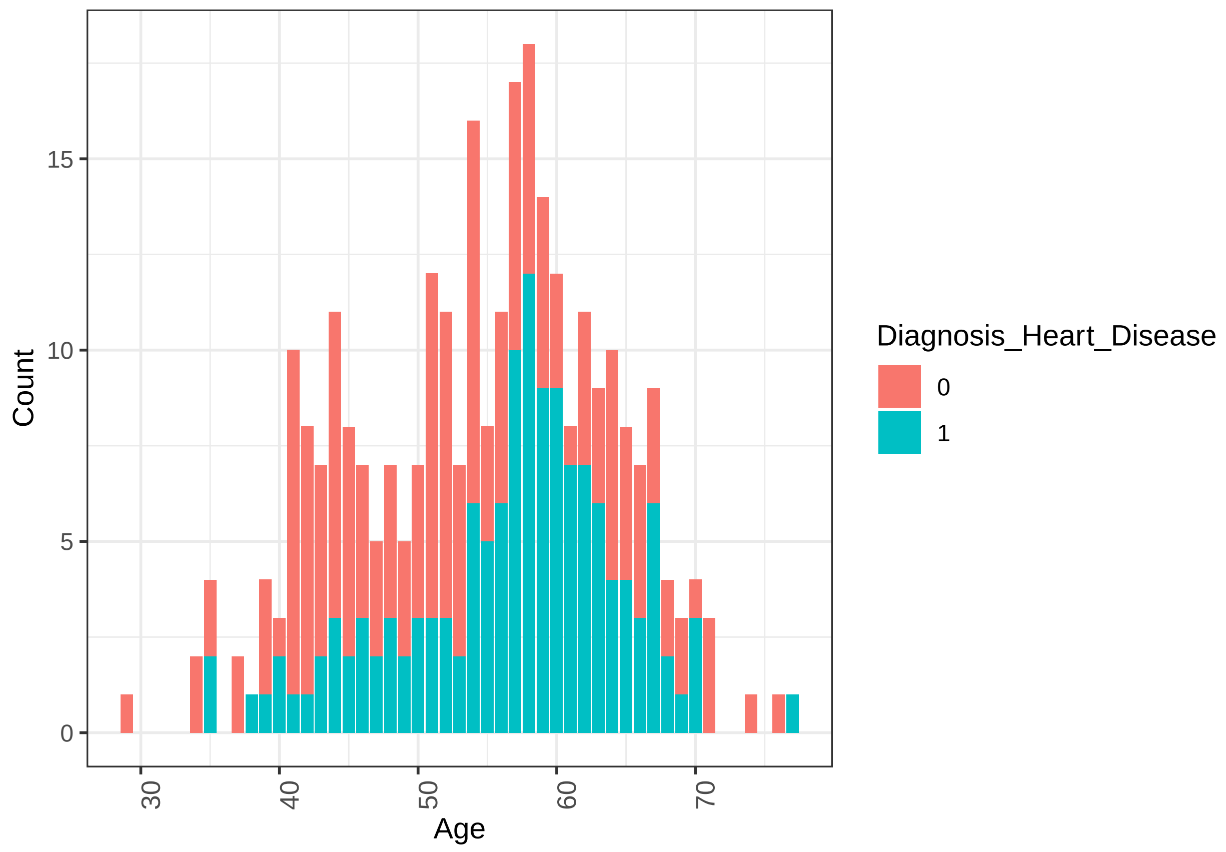
```r
####################################################
# Disease distribution by Sex.
# 0 = Female
# 1 = Male
####################################################
heart_dataset_clean_tbl %>% group_by(Sex, Diagnosis_Heart_Disease) %>% summarise(count = n()) %>%
  ggplot() + geom_bar(aes(Sex, count, fill = Diagnosis_Heart_Disease), stat = "Identity") +
  theme_bw() +
  theme(axis.text.x = element_text(angle = 90, size = 10)) +
  ylab("Count") + xlab("Sex") + labs(fill = "Diagnosis_Heart_Disease")
```



The number of Heart disease patients is higher in male comparing to female.

Heart disease distribution by Age:

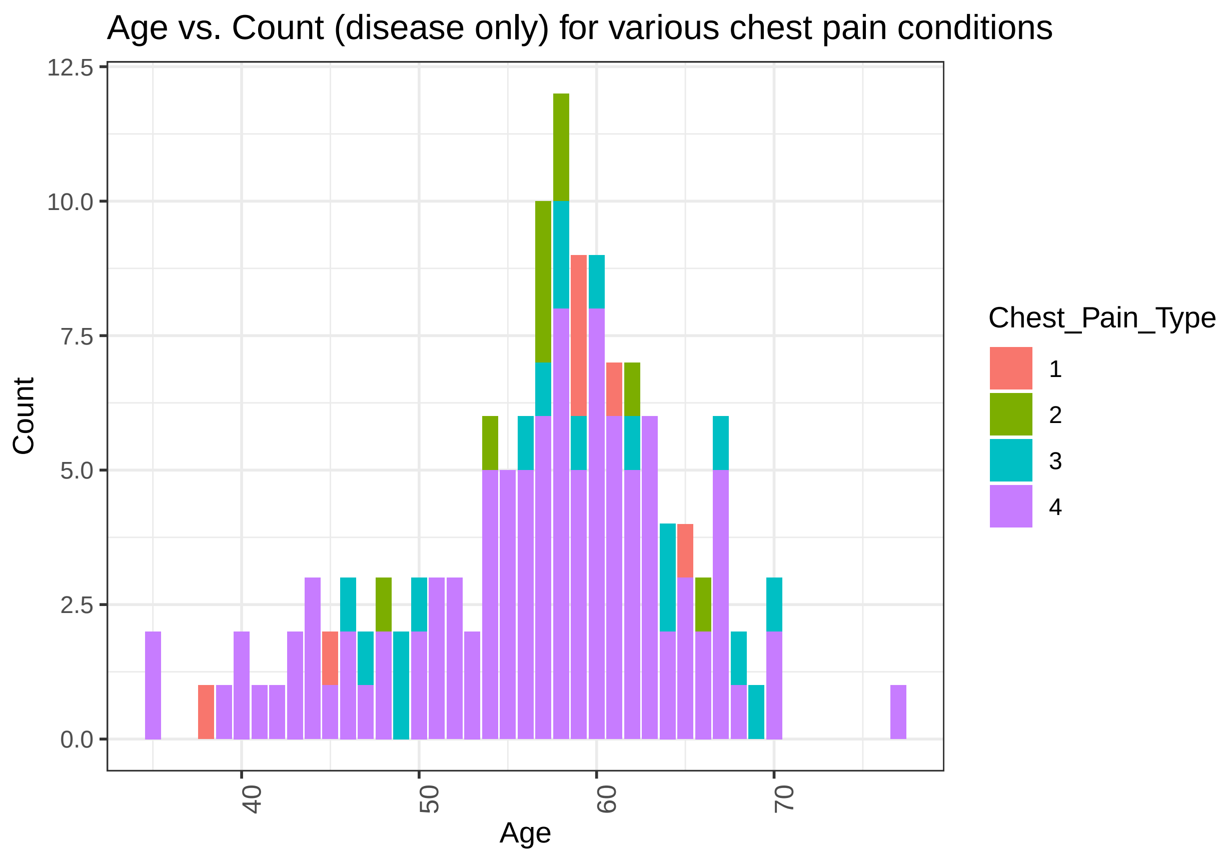
```
########################################################
# Disease distribution by Age.
########################################################

heart_dataset_clean_tbl %>% group_by(Age, Diagnosis_Heart_Disease) %>% summarise(count = n()) %>%
  ggplot() + geom_bar(aes(Age, count, fill = Diagnosis_Heart_Disease), stat = "Identity") +
  theme_bw() +
  theme(axis.text.x = element_text(angle = 90, size = 10)) +
  ylab("Count") + xlab("Age") + labs(fill = "Diagnosis_Heart_Disease")
```



The number of Heart disease higher in patients of age between 50-70 years.

Heart disease distribution by Chest pain type:

```
####################################################
# Chest pain type for diseased people
# You can see - Majority as chest pain type 4
# 1: typical angina 2: atypical angina  Value 3: non-anginal pain Value 4: asymptomatic
####################################################

heart_dataset_clean_tbl %>% filter(Diagnosis_Heart_Disease == '1') %>% group_by(Age, Chest_Pain_Type) %
  ggplot() + geom_bar(aes(Age, count, fill = Chest_Pain_Type),stat = "Identity") +
  theme_bw() +
  theme(axis.text.x = element_text(angle = 90, size = 10)) +
  ylab("Count") + xlab("Age") + labs(fill = "Chest_Pain_Type") +
  ggtitle("Age vs. Count (disease only) for various chest pain conditions")
```



Age between 50-70 age chest pain type patients are mostly effecting with heart disease. Out of which 57 and 58 Aged patients are highly effecting typical angina, atypical angina and Asymptotic chest pain.

## 3.4 Modelling Approach

Data partition into train and test set for model development:

```
#set seed for repeatability
set.seed(2021, sample.kind = "Rounding")

# Divide data into train and test set
train_index <- createDataPartition(heart_dataset_clean_tbl$Diagnosis_Heart_Disease, p = 0.8, list= FALS
train_set <- heart_dataset_clean_tbl[train_index, ]
test_set <- heart_dataset_clean_tbl[-train_index, ]

#checking whether really 80%
nrow(train_set)/(nrow(test_set)+nrow(train_set))
```

```
## [1] 0.8013468
```

Results are going to be stored in variable AUC. AUC is the area under the ROC which represents the proportion of positive data points that are correctly considered as positive and the proportion of negative data points that are mistakenly considered as positive. We also store Accuracy which is true positive and true negative divided by all results.

```
AUC = list()
Accuracy = list()
```

### 3.4.1 Linear Discriminant Analysis (LDA)

```
################################
# LDA Analysis
################################
#set seed for repeat ability
set.seed(2021, sample.kind = "Rounding")

lda_fit <- train(Diagnosis_Heart_Disease ~ ., method = "lda", data = train_set)
ldaPred <- predict(lda_fit, test_set)

ldaPredProb <- predict(lda_fit, test_set, type='prob')[2]
ldaConfMat <- confusionMatrix(ldaPred, test_set$Diagnosis_Heart_Disease)

ldaConfMat
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 31  8
##          1  1 19
##
##                Accuracy : 0.8475
##                  95% CI : (0.7301, 0.9278)
##     No Information Rate : 0.5424
##     P-Value [Acc > NIR] : 7.195e-07
##
##                   Kappa : 0.6864
##
##  Mcnemar's Test P-Value : 0.0455
##
##             Sensitivity : 0.9688
##             Specificity : 0.7037
##          Pos Pred Value : 0.7949
##          Neg Pred Value : 0.9500
##              Prevalence : 0.5424
##          Detection Rate : 0.5254
##    Detection Prevalence : 0.6610
##       Balanced Accuracy : 0.8362
##
##        'Positive' Class : 0
##
```

```
#ROC Curve
AUC$lda <- roc(as.numeric(test_set$Diagnosis_Heart_Disease),
                  as.numeric(as.matrix((ldaPredProb))))$auc
Accuracy$LDA <- ldaConfMat$overall['Accuracy']

row.names <- names(Accuracy)
col.names <- c("AUC", "Accuracy")
Model_result <- cbind(as.data.frame(matrix(c(AUC,Accuracy),nrow = 1, ncol = 2,
                             dimnames = list(row.names, col.names))))
Model_result
```

```
##          AUC  Accuracy
## LDA 0.900463 0.8474576
```

### 3.4.2 Quadrant Discriminant Analysis (QDA)

```
###############################
# QDA Analysis
###############################
#set seed for repeat ability
set.seed(2021, sample.kind = "Rounding")

qda_fit <- train(Diagnosis_Heart_Disease ~ ., method = "qda", data = train_set)

qdaPred <- predict(qda_fit, test_set)

qdaPredProb <- predict(qda_fit, test_set, type='prob')[2]

qdaConfMat <- confusionMatrix(qdaPred, test_set$Diagnosis_Heart_Disease)

qdaConfMat
```

```
## Confusion Matrix and Statistics
```

```
##
##           Reference
## Prediction  0  1
##          0 26  8
##          1  6 19
##
##                Accuracy : 0.7627
##                  95% CI : (0.6341, 0.8638)
##     No Information Rate : 0.5424
##     P-Value [Acc > NIR] : 0.0003962
##
##                   Kappa : 0.5192
##
##  Mcnemar's Test P-Value : 0.7892680
##
##             Sensitivity : 0.8125
##             Specificity : 0.7037
##          Pos Pred Value : 0.7647
##          Neg Pred Value : 0.7600
##              Prevalence : 0.5424
##          Detection Rate : 0.4407
##    Detection Prevalence : 0.5763
##       Balanced Accuracy : 0.7581
##
##        'Positive' Class : 0
##
```

```
#ROC Curve
AUC$qda <- roc(as.numeric(test_set$Diagnosis_Heart_Disease),
                  as.numeric(as.matrix((qdaPredProb))))$auc
Accuracy$QDA <- qdaConfMat$overall['Accuracy']

row.names <- names(Accuracy)
col.names <- c("AUC", "Accuracy")
Model_result <- cbind(as.data.frame(matrix(c(AUC,Accuracy),nrow = 2, ncol = 2,
                           dimnames = list(row.names, col.names))))
Model_result
```

```
##           AUC  Accuracy
## LDA  0.900463 0.8474576
## QDA 0.8368056 0.7627119
```

### 3.4.3 Logistic Regression

```
###############################
# Logistic Regression
###############################
#set seed for repeat ability
set.seed(2021, sample.kind = "Rounding")

logReg_fit <- train(Diagnosis_Heart_Disease ~ .,
                     data=train_set, method = 'glm', family = 'binomial')
```

```r
logRegPred <- predict(logReg_fit, test_set)
logRegPredProb <- predict(logReg_fit, test_set, type='prob')[2]
logRegConfMat <- confusionMatrix(logRegPred, test_set$Diagnosis_Heart_Disease)

logRegConfMat
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 30  7
##          1  2 20
##
##                Accuracy : 0.8475
##                  95% CI : (0.7301, 0.9278)
##     No Information Rate : 0.5424
##     P-Value [Acc > NIR] : 7.195e-07
##
##                   Kappa : 0.6882
##
##  Mcnemar's Test P-Value : 0.1824
##
##             Sensitivity : 0.9375
##             Specificity : 0.7407
##          Pos Pred Value : 0.8108
##          Neg Pred Value : 0.9091
##              Prevalence : 0.5424
##          Detection Rate : 0.5085
##    Detection Prevalence : 0.6271
##       Balanced Accuracy : 0.8391
##
##        'Positive' Class : 0
##
```

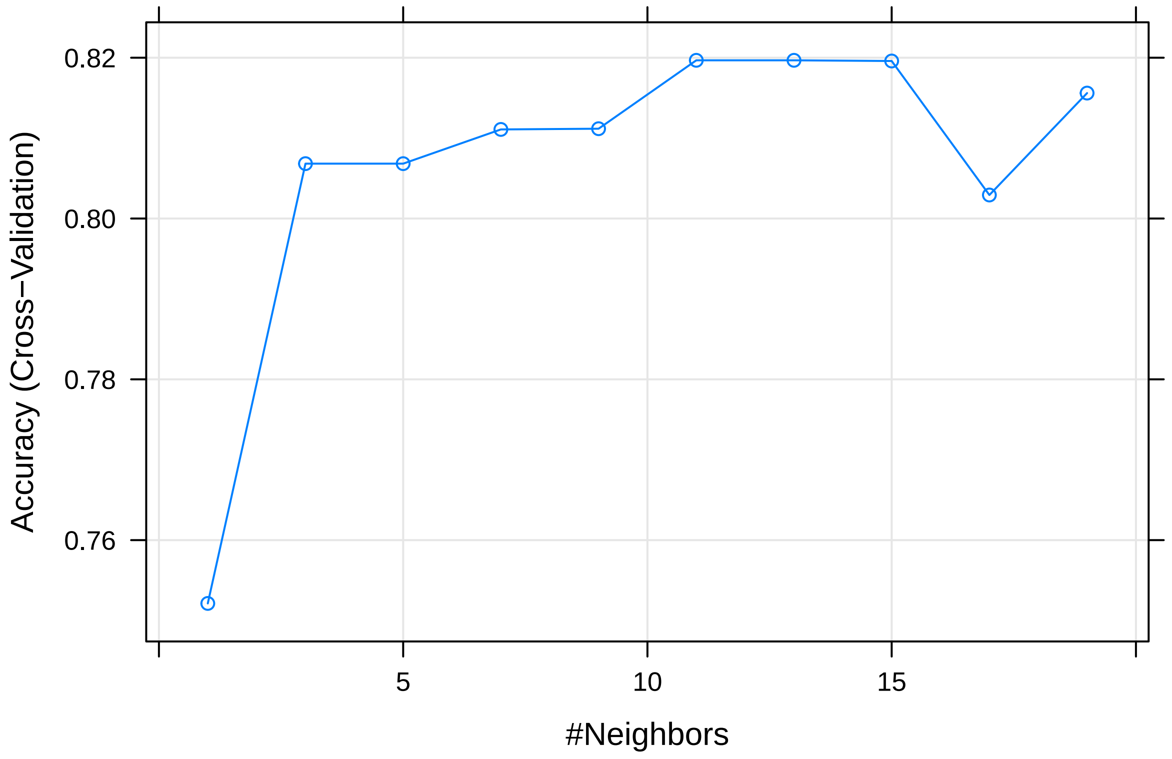
```r
#ROC Curve
AUC$logReg <- roc(as.numeric(test_set$Diagnosis_Heart_Disease),
                  as.numeric(as.matrix((logRegPredProb))))$auc
Accuracy$logReg <- logRegConfMat$overall['Accuracy']

row.names <- names(Accuracy)
col.names <- c("AUC", "Accuracy")
model_result <- cbind(as.data.frame(matrix(c(AUC,Accuracy),nrow = 3, ncol = 2,
                           dimnames = list(row.names, col.names))))
model_result
```

```
##              AUC  Accuracy
## LDA     0.900463 0.8474576
## QDA    0.8368056 0.7627119
## logReg 0.9143519 0.8474576
```

### 3.4.4 KNN Classifier

5-fold cross validation was used, and tuning was done on all the next algorithms discussed here to avoid over-training the algorithms.

```
################################
# Knn Classifier
################################
#set seed for repeat ability
set.seed(2021, sample.kind = "Rounding")

ctrl <- trainControl(method = "cv", verboseIter = FALSE, number = 5)
knn_fit <- train(Diagnosis_Heart_Disease ~ .,
                data = train_set, method = "knn", preProcess = c("center","scale"),
                trControl = ctrl , tuneGrid = expand.grid(k = seq(1, 20, 2)))

plot(knn_fit)
```



```
knnPred <- predict(knn_fit,newdata = test_set)
knnPredProb <- predict(knn_fit, test_set, type='prob')[2]
knnConfMat <- confusionMatrix(knnPred, test_set$Diagnosis_Heart_Disease )

knnConfMat
```

```
## Confusion Matrix and Statistics
```

```
##
##           Reference
## Prediction  0  1
##          0 28  6
##          1  4 21
##
##                Accuracy : 0.8305
##                  95% CI : (0.7103, 0.9156)
##     No Information Rate : 0.5424
##     P-Value [Acc > NIR] : 3.14e-06
##
##                   Kappa : 0.6566
##
##  Mcnemar's Test P-Value : 0.7518
##
##             Sensitivity : 0.8750
##             Specificity : 0.7778
##          Pos Pred Value : 0.8235
##          Neg Pred Value : 0.8400
##              Prevalence : 0.5424
##          Detection Rate : 0.4746
##    Detection Prevalence : 0.5763
##       Balanced Accuracy : 0.8264
##
##        'Positive' Class : 0
##
```

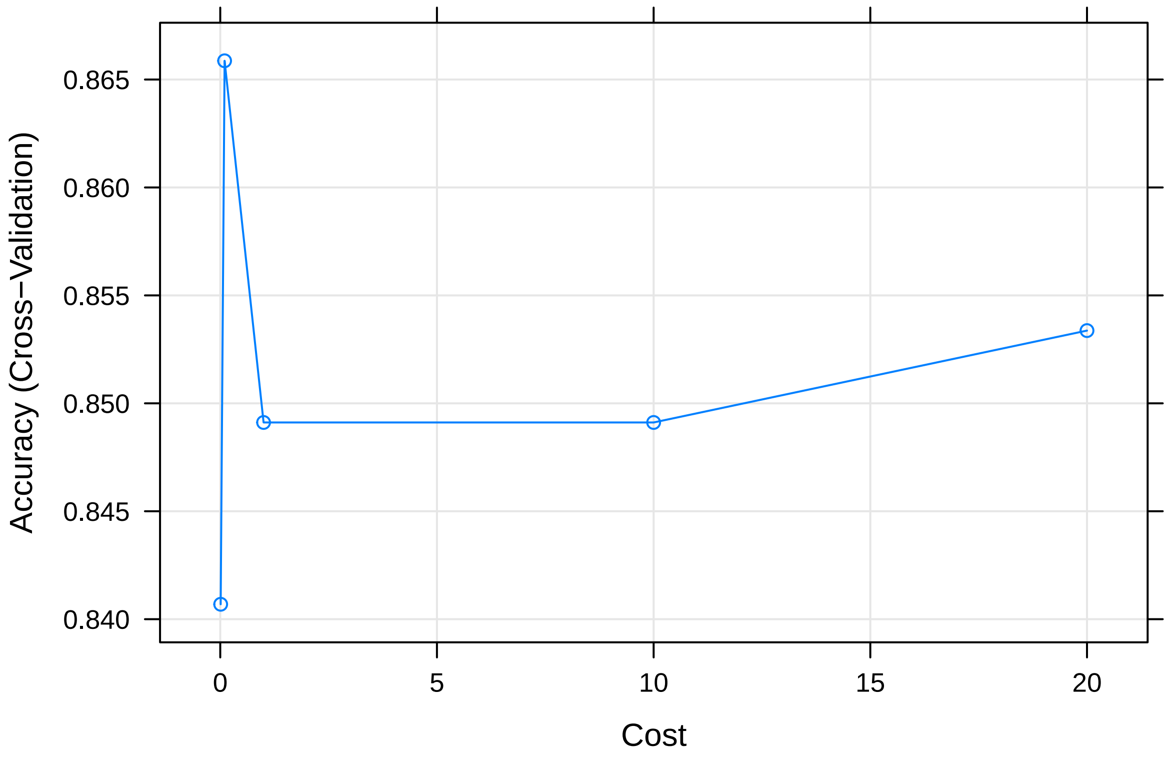
```
#ROC Curve
AUC$Knn <- roc(as.numeric(test_set$Diagnosis_Heart_Disease),
               as.numeric(as.matrix((knnPredProb))))$auc
Accuracy$Knn <- knnConfMat$overall['Accuracy']

row.names <- names(Accuracy)
col.names <- c("AUC", "Accuracy")
model_result <- cbind(as.data.frame(matrix(c(AUC,Accuracy),nrow = 4, ncol = 2,
                           dimnames = list(row.names, col.names))))
model_result
```

```
##              AUC  Accuracy
## LDA     0.900463 0.8474576
## QDA    0.8368056 0.7627119
## logReg 0.9143519 0.8474576
## Knn    0.8755787 0.8305085
```

### 3.4.5 Support Vector Machine (SVM)

```
############################
# SVM
############################
#set seed for repeat ability
set.seed(2021, sample.kind = "Rounding")
```

```r
ctrl <- trainControl(method = "cv", verboseIter = FALSE, number = 5)

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

svm_fit <- train(Diagnosis_Heart_Disease ~ .,data = train_set,
                 method = "svmLinear", preProcess = c("center","scale"),
                 tuneGrid = grid_svm, trControl = ctrl)

plot(svm_fit)
```



```r
svmPred <- predict(svm_fit, newdata = test_set)
svmPredProb <- predict(svm_fit, test_set, type='prob')[2]
svmConfMat <- confusionMatrix(svmPred, test_set$Diagnosis_Heart_Disease)

svmConfMat
```

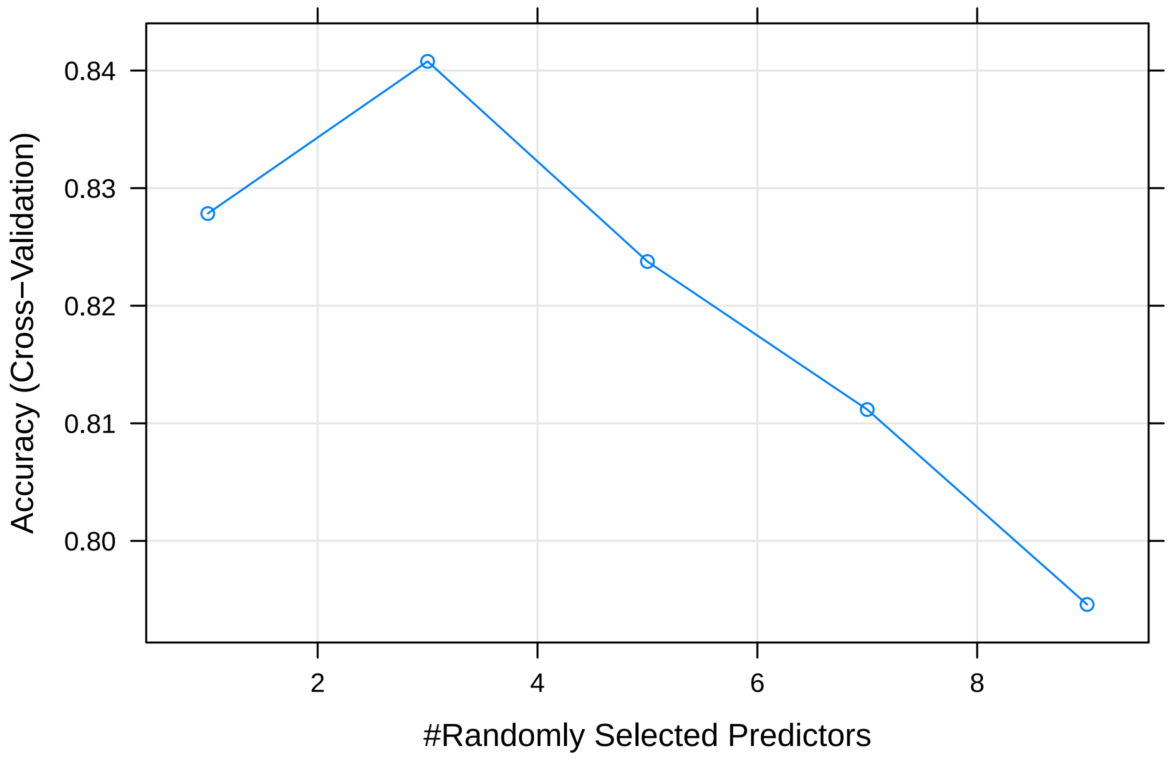
```
## Confusion Matrix and Statistics
##
##             Reference
## Prediction  0  1
##            0 30  8
##            1  2 19
##
##                Accuracy : 0.8305
##                  95% CI : (0.7103, 0.9156)
##     No Information Rate : 0.5424
##     P-Value [Acc > NIR] : 3.14e-06
##
##                   Kappa : 0.6525
##
##  Mcnemar's Test P-Value : 0.1138
##
##             Sensitivity : 0.9375
##             Specificity : 0.7037
##          Pos Pred Value : 0.7895
##          Neg Pred Value : 0.9048
##              Prevalence : 0.5424
##          Detection Rate : 0.5085
##    Detection Prevalence : 0.6441
##       Balanced Accuracy : 0.8206
##
##        'Positive' Class : 0
##
```

```
#ROC Curve
AUC$svm <- roc(as.numeric(test_set$Diagnosis_Heart_Disease),
               as.numeric(as.matrix((svmPred))))$auc
Accuracy$SVM <- svmConfMat$overall['Accuracy']

row.names <- names(Accuracy)
col.names <- c("AUC", "Accuracy")
model_result <- cbind(as.data.frame(matrix(c(AUC,Accuracy),nrow = 5, ncol = 2,
                           dimnames = list(row.names, col.names))))
model_result
```

```
##              AUC  Accuracy
## LDA     0.900463 0.8474576
## QDA    0.8368056 0.7627119
## logReg 0.9143519 0.8474576
## Knn    0.8755787 0.8305085
## SVM    0.8206019 0.8305085
```

#### 3.4.6 Random Forest (RF)

```
############################
# RF
############################
#set seed for repeat ability
set.seed(2021, sample.kind = "Rounding")

ctrl<- trainControl(method = "cv", number = 5, verboseIter = FALSE)
grid <- data.frame(mtry = seq(1, 10, 2))

rf_fit <- train(Diagnosis_Heart_Disease ~ ., method = "rf", data = train_set,
                 ntree = 200, trControl = ctrl, tuneGrid = grid)

plot(rf_fit)
```



```r
rfPred <- predict(rf_fit, newdata = test_set)
rfPredProb <- predict(rf_fit, test_set, type='prob')[2]
rfConfMat <- confusionMatrix(rfPred, test_set$Diagnosis_Heart_Disease)

rfConfMat
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 28  9
##          1  4 18
##
##                Accuracy : 0.7797
##                  95% CI : (0.6527, 0.8771)
##     No Information Rate : 0.5424
##     P-Value [Acc > NIR] : 0.0001366
##
##                   Kappa : 0.5496
##
##  Mcnemar's Test P-Value : 0.2672575
##
##             Sensitivity : 0.8750
##             Specificity : 0.6667
##          Pos Pred Value : 0.7568
##          Neg Pred Value : 0.8182
##               Prevalence : 0.5424
##           Detection Rate : 0.4746
##     Detection Prevalence : 0.6271
##        Balanced Accuracy : 0.7708
##
##         'Positive' Class : 0
##
```

```
#ROC Curve
AUC$rf <- roc(as.numeric(test_set$Diagnosis_Heart_Disease),
              as.numeric(as.matrix((rfPredProb))))$auc
Accuracy$RF <- rfConfMat$overall['Accuracy']

row.names <- names(Accuracy)
col.names <- c("AUC", "Accuracy")
model_result <- cbind(as.data.frame(matrix(c(AUC,Accuracy),nrow = 6, ncol = 2,
                             dimnames = list(row.names, col.names))))
model_result
```

```
##               AUC  Accuracy
## LDA      0.900463 0.8474576
## QDA     0.8368056 0.7627119
## logReg 0.9143519 0.8474576
## Knn     0.8755787 0.8305085
## SVM     0.8206019 0.8305085
## RF      0.9021991  0.779661
```

## 4 Results

### 4.1 Comparison of AUC and Accuracy between models

```
model_result
```

```
##              AUC  Accuracy
## LDA     0.900463 0.8474576
## QDA    0.8368056 0.7627119
## logReg 0.9143519 0.8474576
## Knn    0.8755787 0.8305085
## SVM    0.8206019 0.8305085
## RF     0.9021991  0.779661
```

The best model is the relative simple Linear Discriminant Analysis(LDA) and logistic regression model with an Area under the ROC of 0.9 and 0.91. We can predict heart disease with an accuracy of 0.847 in both. The Sensitivity and Specificity is 0.96, 0.70 for LDA and 0.93, 0.74 for Logistic Regression.

## 5 Conclusion

The short analysis shows the predictive capability of machine learning algorithms for Cleveland heart disease. There are 14 variables from the Processed Cleveland heart disease dataset used to predict the diagnosis of heart disease (angiographic disease status). The performances of different machine learning algorithms like LDA, QDA, logistic regression, Knn, random forest and support vector machines - are compared . 80% of the data is hold out as a training set that is not seen during the testing stage of the data. Remaining 20% of the data is hold out as a testing set that is not seen during the training stage of the data. A comparison of the area under the ROC and the accuracy of the model predictions shows that LDA and logistic regression performs best with similar accuracy of 0.847 but different ROC, sensitivity and specificity. Tree-based methods with different tuning parameters performed slightly worse.

We can use Gradient Boosting Machine (GBM) which is unique compared to other decision tree algorithms because it builds models sequentially with higher weights give to those cases that were poorly predicted in previous models, thus improving accuracy incrementally instead of simply taking the average of all models like a Random Forest algorithm would. By reducing the error iteratively to produce best final model. GBM is an efficient powerful algorithm for both classification and Regression problems.