Heart Disease Prediction

Sweta Swarupa

21/01/22

Contents

1 Introduction

Cardiovascular diseases (CVDs) are the number 1 cause of death globally, taking an estimated 17.9 million lives each year, which accounts for 31% of all deaths worldwide. Four out of 5CVD deaths are due to heart attacks and strokes, and one-third of these deaths occur prematurely in people under 70 years of age. Heart failure is a common event caused by CVDs and this dataset contains 11 features that can be used to predict a possible heart disease.

People with cardiovascular disease or who are at high cardiovascular risk (due to the presence of one or more risk factors such as hypertension, diabetes, hyperlipidaemia or already established disease) need early detection and management wherein a machine learning model can be of great help.

1.1 Source:

The dataset is taken from Kaggle. https://www.kaggle.com/fedesoriano/heart-failure-prediction

This dataset was created by combining different datasets already available independently but not combined before. In this dataset, 5 heart datasets are combined over 11 common features which makes it the largest heart disease dataset available so far for research purposes. The five datasets used for its curation are:

- Cleveland: 303 observations
- Hungarian: 294 observations
- Switzerland: 123 observations
- Long Beach VA: 200 observations
- Stalog (Heart) Data Set: 270 observations

Total: 1190 observations, Duplicated: 272 observations, Final dataset: 918 observations

1.2 Attribute Information:

1. Age: age of the patient [years]
2. Sex: sex of the patient [M: Male, F: Female]
3. ChestPainType: chest pain type [TA: Typical Angina, ATA: Atypical Angina, NAP: Non-Anginal Pain, ASY: Asymptomatic]

4. RestingBP: resting blood pressure [mm Hg]
5. Cholesterol: serum cholesterol [mm/dl]
6. FastingBS: fasting blood sugar [1: if FastingBS > 120 mg/dl, 0: otherwise]
7. RestingECG: resting electrocardiogram results [Normal: Normal, ST: having ST-T wave abnormality (T wave inversions and/or ST elevation or depression of > 0.05 mV), LVH: showing probable or definite left ventricular hypertrophy by Estes' criteria]
8. MaxHR: maximum heart rate achieved [Numeric value between 60 and 202]
9. ExerciseAngina: exercise-induced angina [Y: Yes, N: No]
10. Oldpeak: oldpeak = ST [Numeric value measured in depression]
11. ST_Slope: the slope of the peak exercise ST segment [Up: upsloping, Flat: flat, Down: downsloping]
12. HeartDisease: output class [1: heart disease, 0: Normal]

1.3 Reading the data

```
heart <- read.csv('C:/Heart Failure Prediction/heart.csv', header = TRUE)
str(heart)
## 'data.frame':    918 obs. of  12 variables:
##  $ Age           : int  40 49 37 48 54 39 45 54 37 48 ...
##  $ Sex           : chr  "M" "F" "M" "F" ...
##  $ ChestPainType : chr  "ATA" "NAP" "ATA" "ASY" ...
##  $ RestingBP     : int  140 160 130 138 150 120 130 110 140 120 ...
##  $ Cholesterol   : int  289 180 283 214 195 339 237 208 207 284 ...
##  $ FastingBS     : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ RestingECG    : chr  "Normal" "Normal" "ST" "Normal" ...
##  $ MaxHR         : int  172 156 98 108 122 170 170 142 130 120 ...
##  $ ExerciseAngina: chr  "N" "N" "N" "Y" ...
##  $ Oldpeak       : num  0 1 0 1.5 0 0 0 0 1.5 0 ...
##  $ ST_Slope      : chr  "Up" "Flat" "Up" "Flat" ...
##  $ HeartDisease  : int  0 1 0 1 0 0 0 0 1 0 ...
```

There are 918 observations and 12 variables. There are 5 character variables, 6 integer variables and 1 numeric variable.

1.4 Checking the number of unique values for each variable.

```
sapply(heart, n_distinct)
##           Age           Sex  ChestPainType      RestingBP    Cholesterol
##            50             2             4             67            222
##     FastingBS     RestingECG         MaxHR ExerciseAngina       Oldpeak
##             2             3           119             2            53
##      ST_Slope   HeartDisease
##             3             2
```

A portion of the heart data set is shown below:

Table 1: Heart Data

Age	Sex	ChestPainType	RestingBP	Cholesterol	FastingBS	RestingECG	MaxHR	ExerciseAngina
40	M	ATA	140	289	0	Normal	172	N
49	F	NAP	160	180	0	Normal	156	N
37	M	ATA	130	283	0	ST	98	N
48	F	ASY	138	214	0	Normal	108	Y
54	M	NAP	150	195	0	Normal	122	N

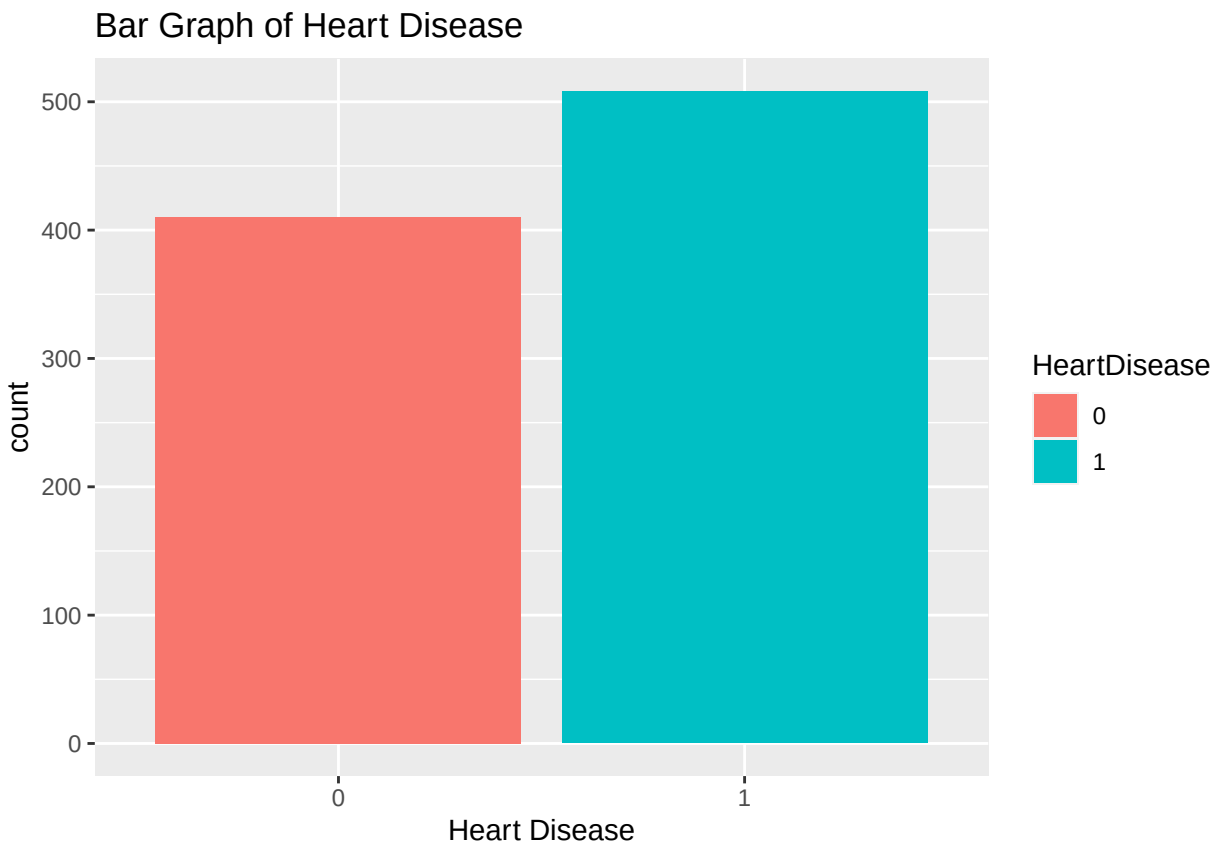
1.5 Checking for any Missing Values.

```
sapply(heart, function(x) sum(is.na(x)))
##            Age            Sex  ChestPainType      RestingBP    Cholesterol
##              0              0              0              0              0
##      FastingBS     RestingECG          MaxHR ExerciseAngina        Oldpeak
##              0              0              0              0              0
##       ST_Slope   HeartDisease
##              0              0
```

There are no missing values.

1.6 Checking for Target Imbalance

```
heart$HeartDisease <- as.character(heart$HeartDisease)
ggplot(data = heart, aes(x=HeartDisease, fill = HeartDisease)) +
  geom_bar() + labs(x='Heart Disease') + labs(title = "Bar Graph of Heart Disease")
```



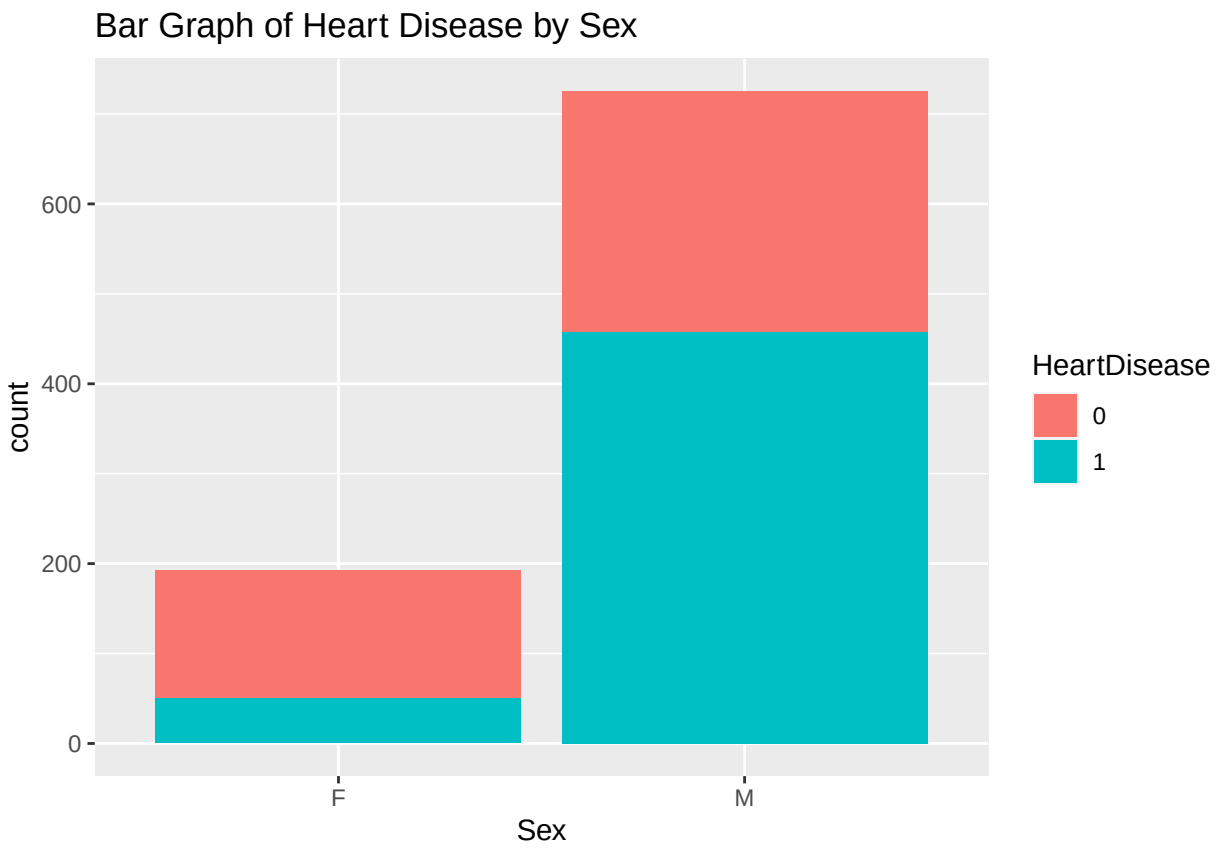

```
table(heart$HeartDisease)
##
##   0   1
## 410 508
prop.table(table(heart$HeartDisease))
##
##         0         1
## 0.4466231 0.5533769
```

The target looks balanced. 44.6% of the observations do not have heart diseases whereas 55.3% of the observations have the heart disease.

2 Data Analysis

2.1 Plotting Sex Variable

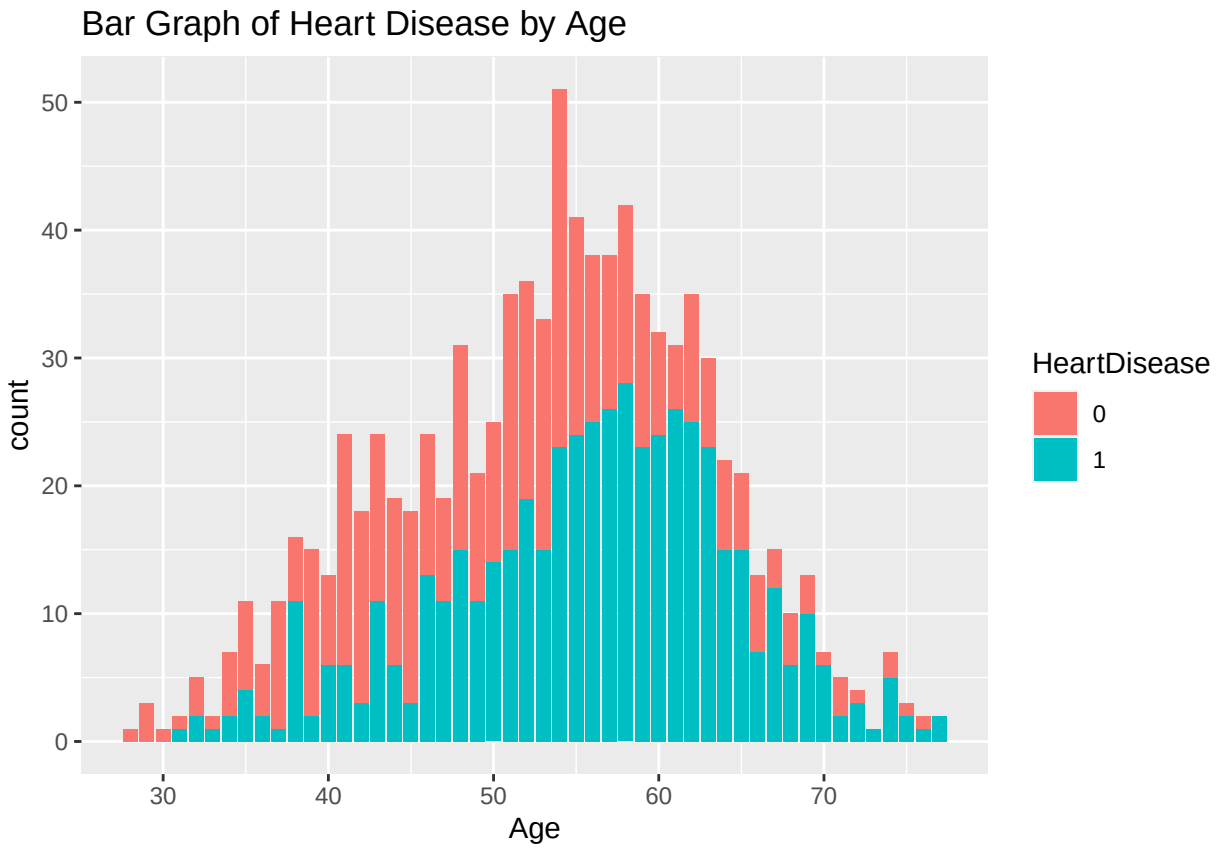
```
ggplot(data = heart, aes(x=Sex, fill = HeartDisease, position = 'dodge')) +
  geom_bar() + labs(x='Sex') + labs(title = "Bar Graph of Heart Disease by Sex")
```



Looks like males are affected more by heart disease than females.

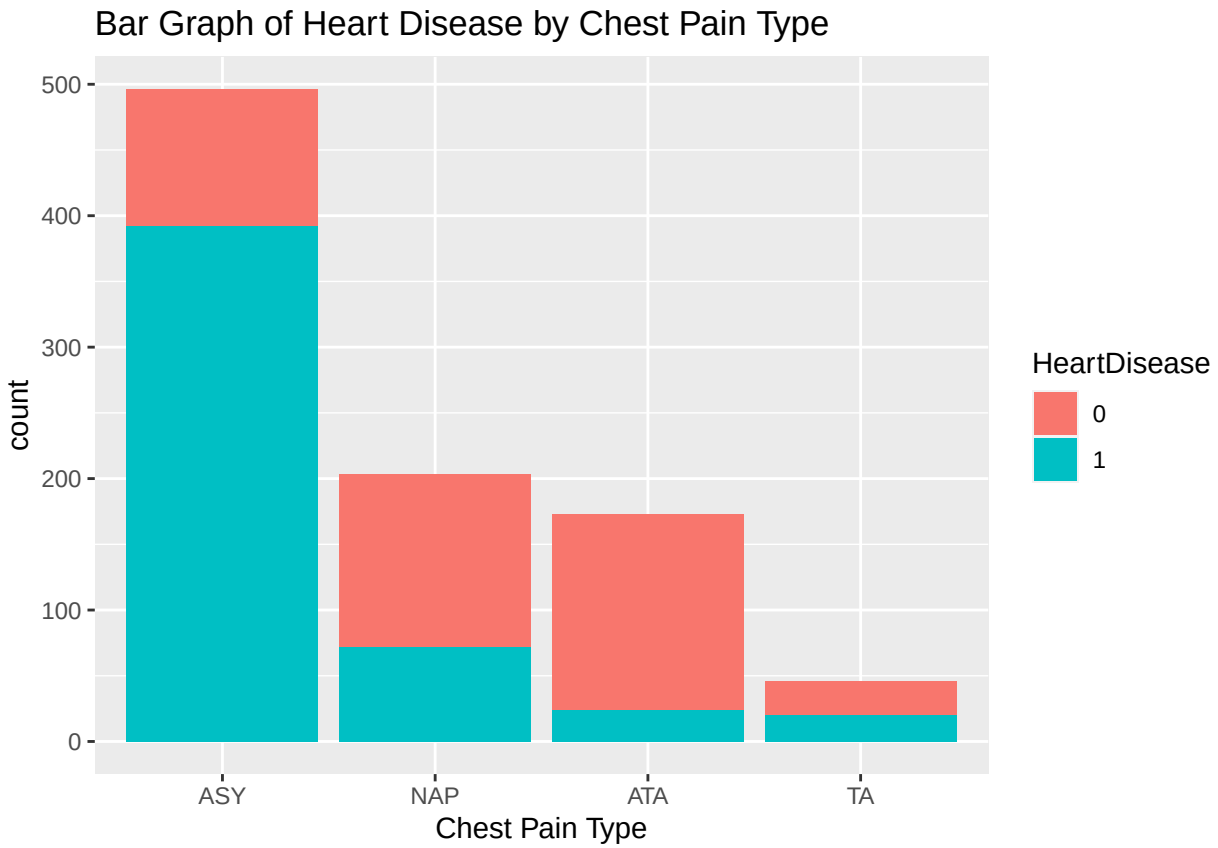
2.2 Plotting Age Variable

```
ggplot(data = heart, aes(x=Age, fill = HeartDisease, position = 'dodge')) +
  geom_bar() + labs(x='Age') + labs(title = "Bar Graph of Heart Disease by Age")
```



2.3 Plotting ChestPainType Variable

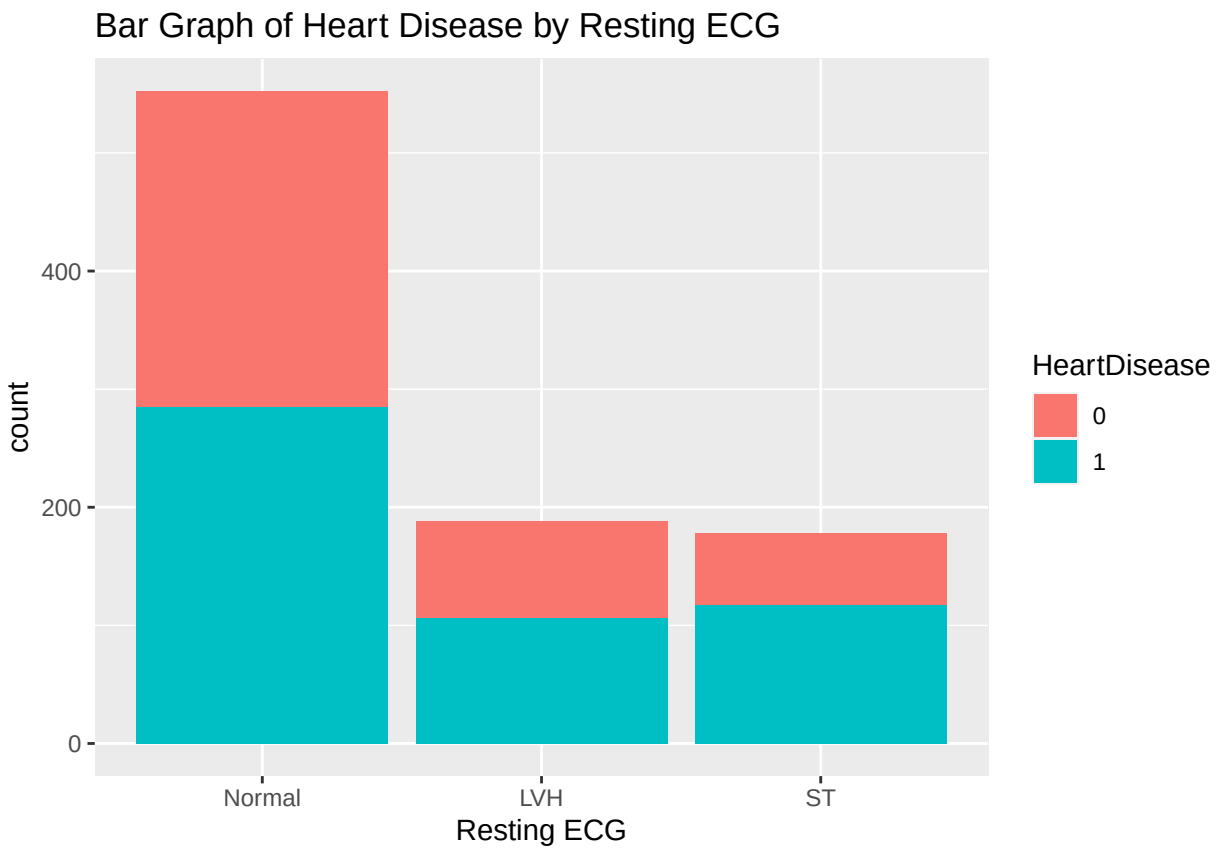
```
ggplot(data = heart, aes(x=reorder(ChestPainType, ChestPainType, function(x)-length(x)), fill = HeartDisea
  geom_bar() + labs(x='Chest Pain Type') + labs(title = "Bar Graph of Heart Disease by Chest Pain Type")
```



Proportion of heart patients is higher when the chest pain type is ASY.

2.4 Plotting RestingECG Variable

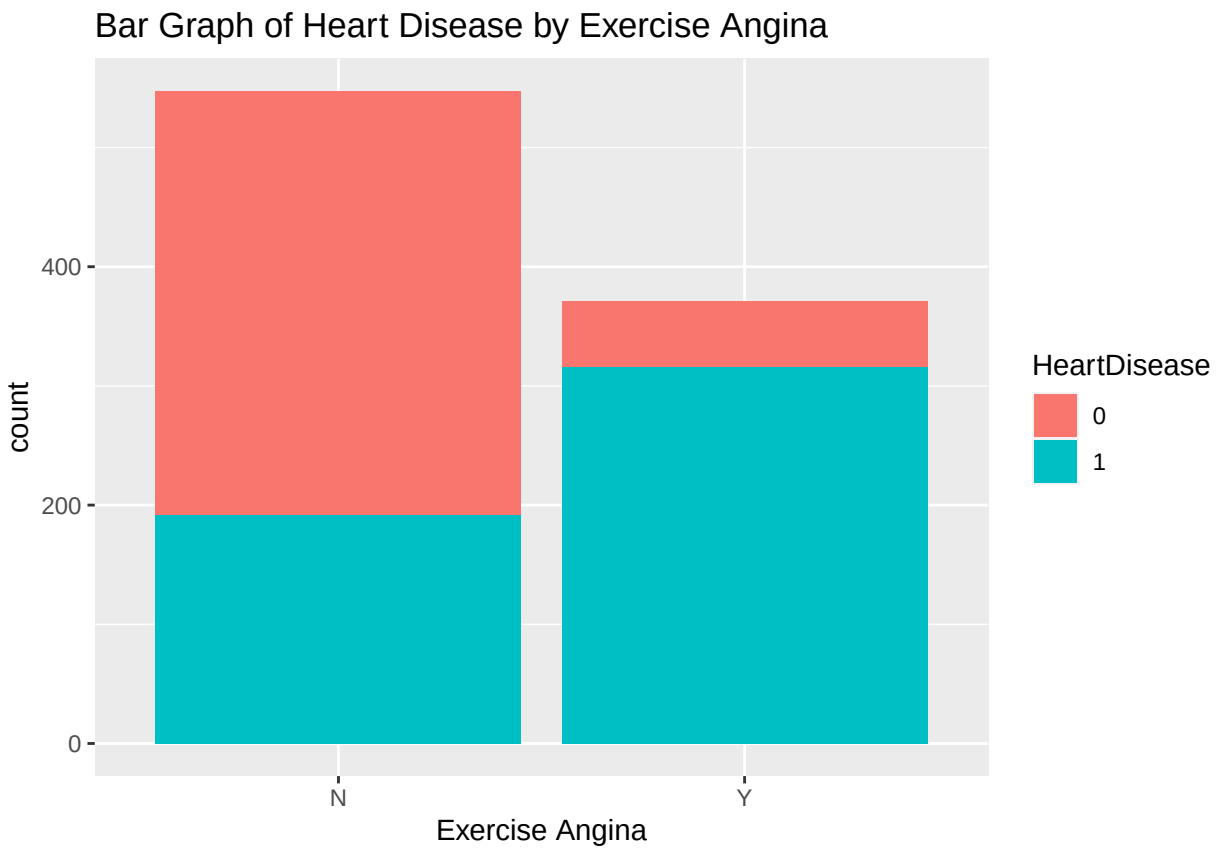
```
ggplot(data = heart, aes(x=reorder(RestingECG, RestingECG, function(x)-length(x)), fill = HeartDisease, po
  geom_bar() + labs(x='Resting ECG') + labs(title = "Bar Graph of Heart Disease by Resting ECG")
```



Proportion of heart patients is higher when the Resting ECG type is LVH and ST.

2.5 Plotting ExerciseAngina Variable

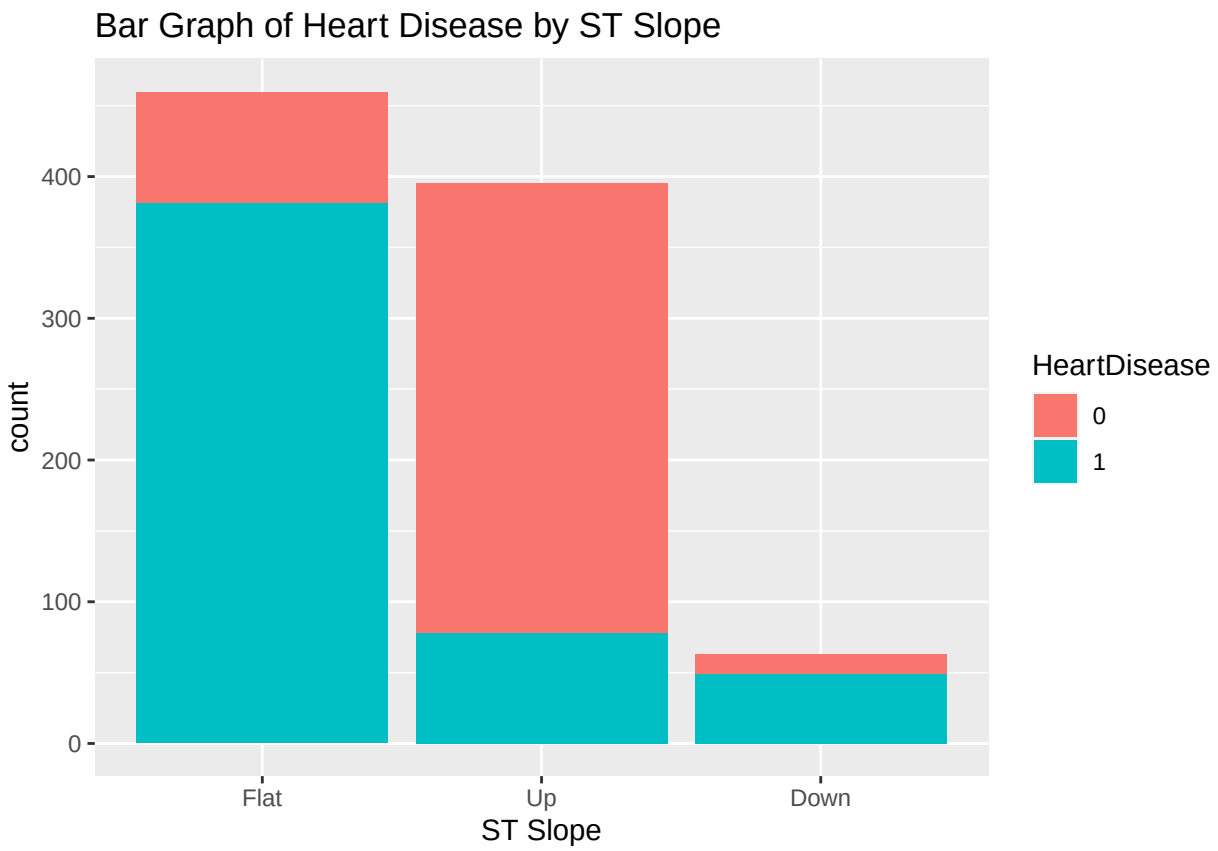
```
ggplot(data = heart, aes(x=ExerciseAngina, fill = HeartDisease, position = 'dodge')) +
  geom_bar() + labs(x='Exercise Angina') + labs(title = "Bar Graph of Heart Disease by Exercise Angina")
```



Proportion of heart patients is significantly higher when the ExerciseAngina is Y.

2.6 Plotting ST_Slope Variable

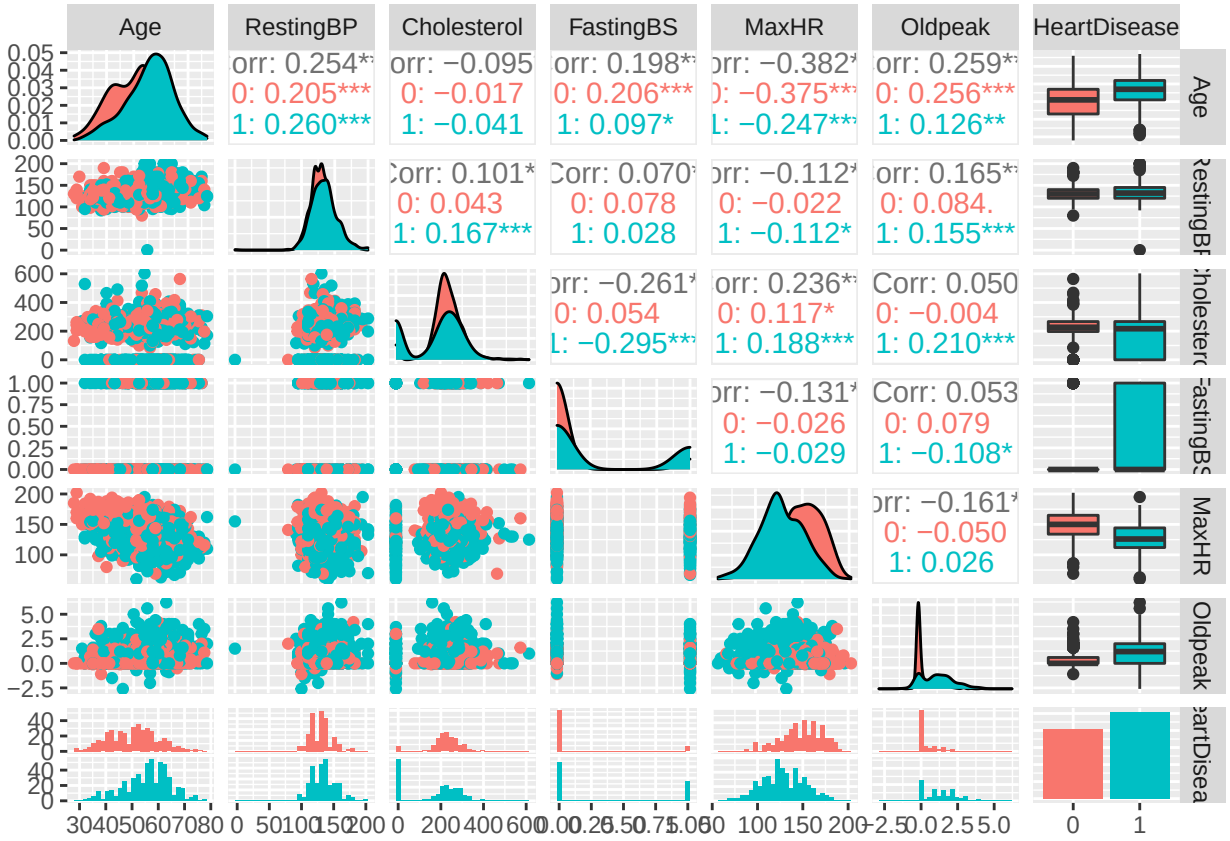
```
ggplot(data = heart, aes(x=reorder(ST_Slope, ST_Slope, function(x)-length(x)), fill = HeartDisease, positi
  geom_bar() + labs(x='ST Slope') + labs(title = "Bar Graph of Heart Disease by ST Slope")
```



Proportion of heart patients is significantly higher when the ST_Slope is Flat and Down.

2.7 Pair Plot

```
heart1 <- heart %>% select(-c(Sex,ChestPainType,RestingECG,ExerciseAngina,ST_Slope))
heart1$HeartDisease <- as.character(heart1$HeartDisease)
ggpairs(heart1, ggplot2::aes(colour=HeartDisease))
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

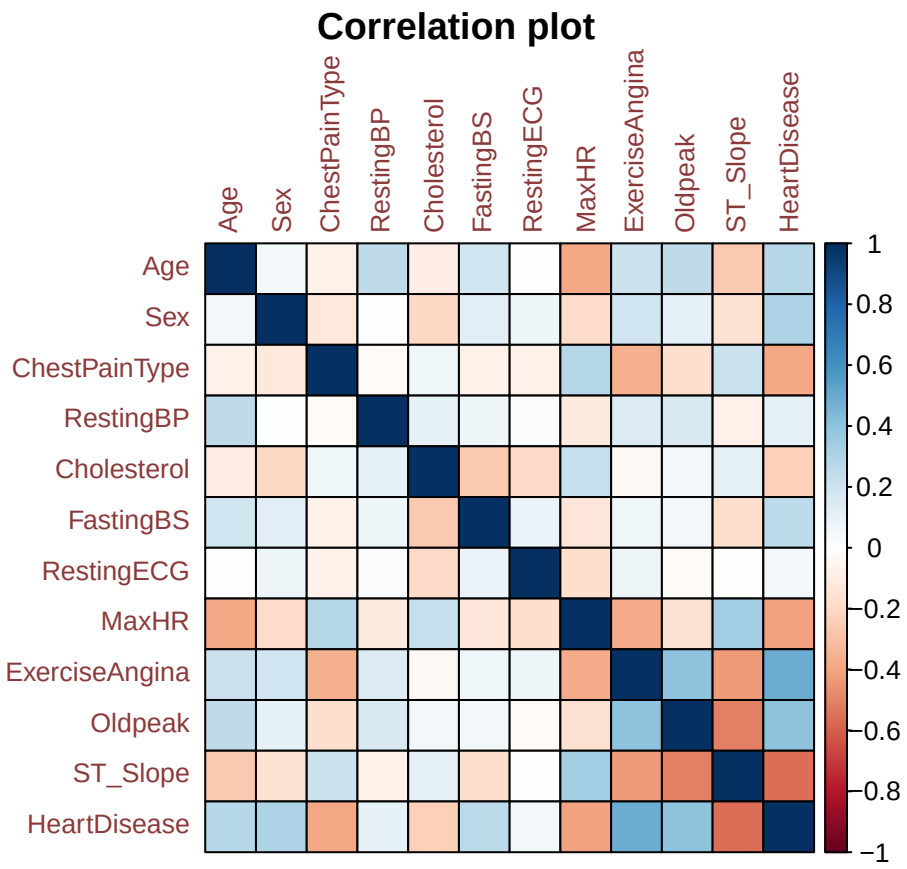


The above graph shows the distribution of observations with heart disease and without heart disease.

3 Checking Correlation between Variables

```
library(corrplot)
#Converting Categorical Variables into Numerical Variable
heart2 <- heart %>% mutate_if(is.character, as.factor)
heart2 <- heart2 %>%  mutate_if(is.factor, as.numeric)

corrplot(cor(heart2), type="full",
         method ="color", title = "Correlation plot",
         mar=c(0,0,1,0), tl.cex= 0.8, outline= T, tl.col="indianred4")
```




```
round(cor(heart2),2)
##                 Age   Sex ChestPainType RestingBP Cholesterol FastingBS
## Age            1.00  0.06         -0.08      0.25       -0.10      0.20
## Sex            0.06  1.00         -0.13      0.01       -0.20      0.12
## ChestPainType -0.08 -0.13          1.00     -0.02        0.07     -0.07
## RestingBP      0.25  0.01         -0.02      1.00        0.10      0.07
## Cholesterol   -0.10 -0.20          0.07      0.10        1.00     -0.26
## FastingBS      0.20  0.12         -0.07      0.07       -0.26      1.00
## RestingECG    -0.01  0.07         -0.07      0.02       -0.20      0.09
## MaxHR         -0.38 -0.19          0.29     -0.11        0.24     -0.13
## ExerciseAngina 0.22  0.19         -0.35      0.16       -0.03      0.06
## Oldpeak        0.26  0.11         -0.18      0.16        0.05      0.05
## ST_Slope      -0.27 -0.15          0.21     -0.08        0.11     -0.18
## HeartDisease   0.28  0.31         -0.39      0.11       -0.23      0.27
##                RestingECG MaxHR ExerciseAngina Oldpeak ST_Slope HeartDisease
## Age                 -0.01 -0.38           0.22    0.26    -0.27         0.28
## Sex                  0.07 -0.19           0.19    0.11    -0.15         0.31
## ChestPainType       -0.07  0.29          -0.35   -0.18     0.21        -0.39
## RestingBP            0.02 -0.11           0.16    0.16    -0.08         0.11
## Cholesterol         -0.20  0.24          -0.03    0.05     0.11        -0.23
## FastingBS            0.09 -0.13           0.06    0.05    -0.18         0.27
## RestingECG           1.00 -0.18           0.08   -0.02    -0.01         0.06
## MaxHR               -0.18  1.00          -0.37   -0.16     0.34        -0.40
## ExerciseAngina       0.08 -0.37           1.00    0.41    -0.43         0.49
## Oldpeak             -0.02 -0.16           0.41    1.00    -0.50         0.40
## ST_Slope            -0.01  0.34          -0.43   -0.50     1.00        -0.56
## HeartDisease         0.06 -0.40           0.49    0.40    -0.56         1.00
```

Heart Disease has high negative correlation with ST_Slope followed by MaxHR and high positive correlation with

ExerciseAngina followed by Oldpeak.

4 Splitting the Data into Training and Test datasets

```
library(caret)
heart <- heart %>% mutate_if(is.character, as.factor)
heart$HeartDisease <- as.factor(heart$HeartDisease)

set.seed(5)
trainIndex <- createDataPartition(heart$HeartDisease, p = .7,
                                  list = FALSE,
                                  times = 1)
Train <- heart[ trainIndex,]
Test <- heart[-trainIndex,]

prop.table(table(Train$HeartDisease))
##
##         0         1
## 0.4463453 0.5536547
prop.table(table(Test$HeartDisease))
##
##         0         1
## 0.4472727 0.5527273
```

Splitting data into 70% Training and 30% Test. We can see the proportion of heart patients and normal in both training and test dataset is similar to heart dataset.

5 Model 1 Logistic Regression Model

5.1 Building Model

```
lm <- glm(HeartDisease ~.,family=binomial(link='logit'),data=Train)
anova(lm, test="Chisq")
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: HeartDisease
##
## Terms added sequentially (first to last)
##
##
##                Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
## NULL                           642     883.97
## Age             1   60.455       641     823.51 7.528e-15 ***
## Sex             1   69.200       640     754.31 < 2.2e-16 ***
## ChestPainType   3  159.658       637     594.66 < 2.2e-16 ***
## RestingBP       1    0.162       636     594.49 0.6877076
## Cholesterol     1   14.700       635     579.79 0.0001260 ***
## FastingBS       1   19.218       634     560.58 1.166e-05 ***
## RestingECG      2    0.307       632     560.27 0.8575041
```

```
## MaxHR           1   12.802       631     547.47 0.0003462 ***
## ExerciseAngina  1   39.088       630     508.38 4.052e-10 ***
## Oldpeak         1   26.500       629     481.88 2.635e-07 ***
## ST_Slope        2   74.318       627     407.56 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

5.2 Using McFadden R2 index to assess the model fit

```
library(pscl)
## Classes and Methods for R developed in the
## Political Science Computational Laboratory
## Department of Political Science
## Stanford University
## Simon Jackman
## hurdle and zeroinfl functions by Achim Zeileis
pR2(lm)
## fitting null model for pseudo-r2
##          llh      llhNull           G2     McFadden         r2ML         r2CU
## -203.7801563 -441.9843216  476.4083305    0.5389426    0.5233224    0.7004728
```

5.3 Using the model to predict heart disease in the test dataset

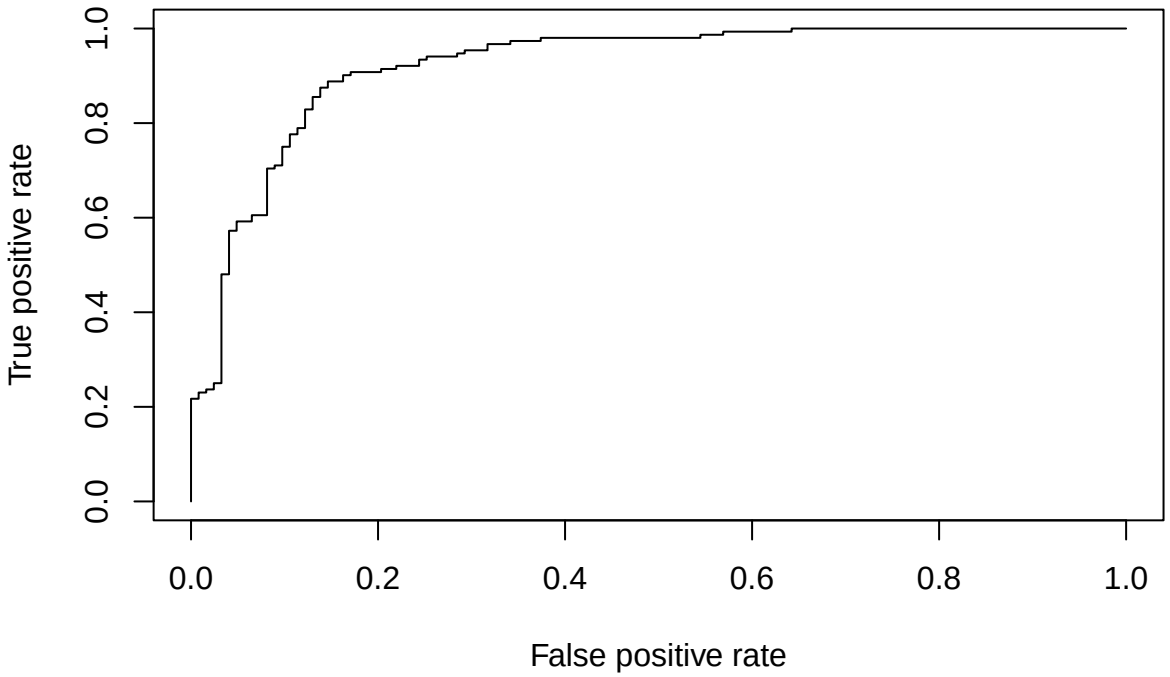
```
pred_lm <- predict(lm, newdata = Test, type = 'response')
```

5.4 Checking accuracy of the model on the test dataset

```
pred_lm <- ifelse(pred_lm > 0.5,1,0)
misClasificError <- mean(pred_lm != Test$HeartDisease)
print(paste('Accuracy',1-misClasificError))
## [1] "Accuracy 0.869090909090909"
```

5.5 ROC Plot

```
library(ROCR)
p <- predict(lm, newdata=Test, type="response")
pr <- prediction(p, Test$HeartDisease)
prf <- performance(pr, measure = "tpr", x.measure = "fpr")
plot(prf)
```



5.6 AUC

```
auc <- performance(pr, measure = "auc")
auc <- auc@y.values[[1]]
auc
## [1] 0.921641
```

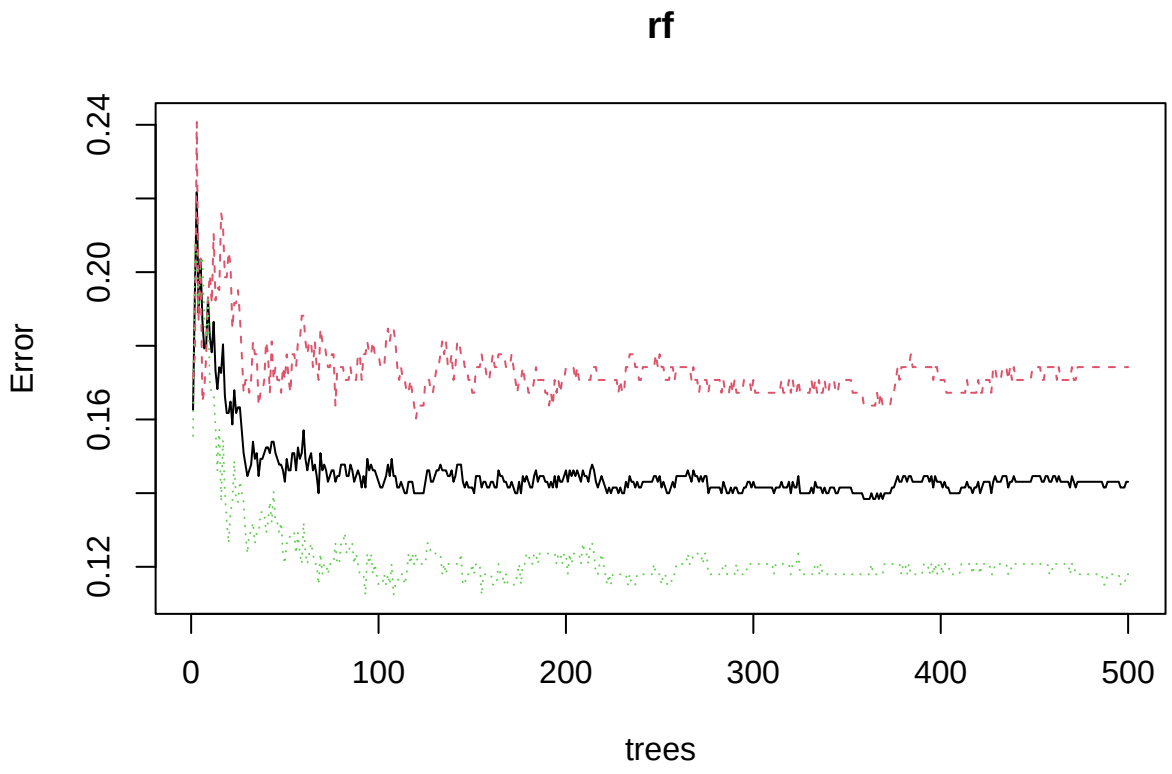
The accuracy of the model is 86.9%

6 Model 2 Random Forest Model

6.1 Building Model

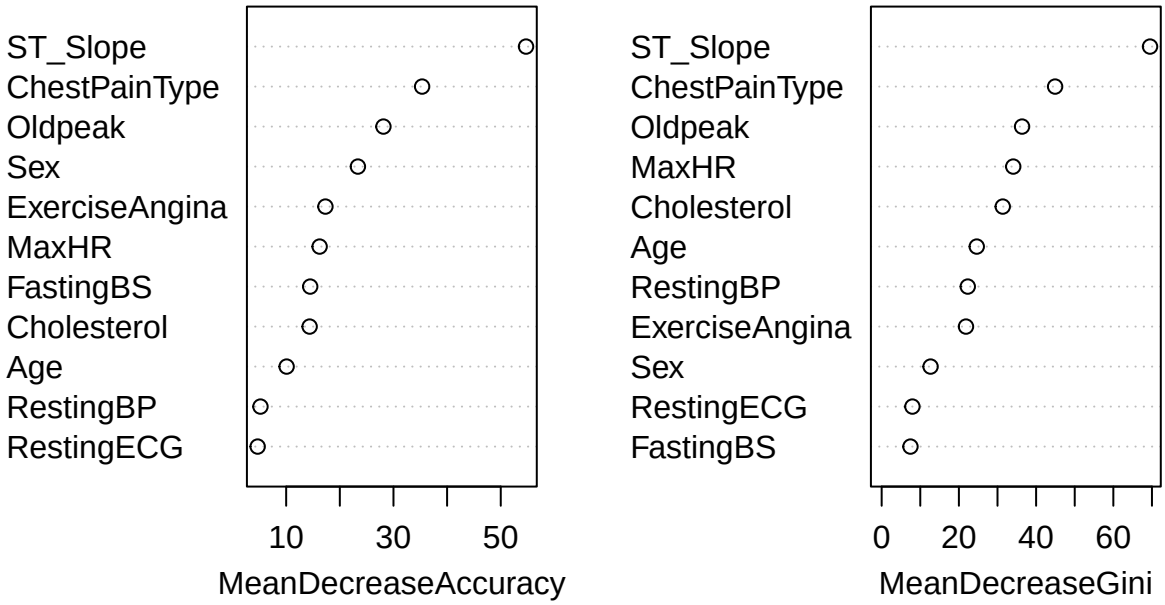
```
library(randomForest)
set.seed(5)
rf <- randomForest(HeartDisease~.,data = Train, type = "class", importance=TRUE, ntree= 500, mtry = 3)
print(rf)
##
## Call:
##  randomForest(formula = HeartDisease ~ ., data = Train, type = "class",      importance = TRUE, ntree =
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 3
```

```
##
##         OOB estimate of  error rate: 14.31%
## Confusion matrix:
##     0   1 class.error
## 0 237  50   0.1742160
## 1  42 314   0.1179775
plot(rf)
```



6.2 Feature Importance Plot

```
varImpPlot(rf, main ='Feature Importance')
```



6.3 Using the model to predict heart diseases in the test dataset

```
pred_rf <- predict(rf, Test, type = "class")
```

6.4 Confusion Matrix

```
confusionMatrix(as.factor(pred_rf),as.factor(Test$HeartDisease))
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 102  12
##          1  21 140
##
##                Accuracy : 0.88
##                  95% CI : (0.8356, 0.9159)
##     No Information Rate : 0.5527
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.7556
##
##  Mcnemar's Test P-Value : 0.1637
##
##             Sensitivity : 0.8293
```

```
##             Specificity : 0.9211
##          Pos Pred Value : 0.8947
##          Neg Pred Value : 0.8696
##              Prevalence : 0.4473
##          Detection Rate : 0.3709
##    Detection Prevalence : 0.4145
##       Balanced Accuracy : 0.8752
##
##        'Positive' Class : 0
##
```

The accuracy of the model is 88%.

7 Model 3 Naive Bayes Model

7.1 Set up 10-fold cross validation procedure

```
library(caret)
train_control <- trainControl(
  method = "cv",
  number = 10
)
```

7.2 Building Model

```
set.seed(123)
nb <- train(
  x = Train,
  y = Train$HeartDisease,
  method = "nb",
  trControl = train_control
)
nb
## Naive Bayes
##
## 643 samples
##  12 predictor
##   2 classes: '0', '1'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 578, 579, 578, 579, 579, 579, ...
## Resampling results across tuning parameters:
##
##   usekernel  Accuracy   Kappa
##   FALSE      0.9937500  0.9873078
##    TRUE      0.9906731  0.9811214
##
## Tuning parameter 'fL' was held constant at a value of 0
## Tuning
##  parameter 'adjust' was held constant at a value of 1
```

```
## Accuracy was used to select the optimal model using the largest value.
## The final values used for the model were fL = 0, usekernel = FALSE and adjust
##  = 1.
```

7.3 Using the model to predict heart diseases in the test dataset

```
pred_nb <- predict(nb, newdata = Test)
```

7.4 Confusion Matrix

```
confusionMatrix(pred_nb, Test$HeartDisease)
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 118   0
##          1   5 152
##
##                Accuracy : 0.9818
##                  95% CI : (0.9581, 0.9941)
##     No Information Rate : 0.5527
##     P-Value [Acc > NIR] : < 2e-16
##
##                   Kappa : 0.9631
##
##  Mcnemar's Test P-Value : 0.07364
##
##             Sensitivity : 0.9593
##             Specificity : 1.0000
##          Pos Pred Value : 1.0000
##          Neg Pred Value : 0.9682
##              Prevalence : 0.4473
##          Detection Rate : 0.4291
##    Detection Prevalence : 0.4291
##       Balanced Accuracy : 0.9797
##
##        'Positive' Class : 0
##
```

The accuracy of the model is 98.1%. The accuracy of this model could be high due to overfitting.

8 Conclusion and Model Comparison

We used logistic regression, random forest and naive bayes models to predict heart disease and we see that naive bayes gives us a better accuracy among the three models. The accuracy comparison for three different models is shown below.

Model	Accuracy
Logistic Regression	0.869
Random Forest	0.880
Naive Bayes	0.981