# Heart Disease Prediction

Tarek Elkady

2/22/2022

## Introduction/Overview

AS per Centers for Disease Control and Prevention (CDC), Coronary artery disease (CAD) is the most common type of heart disease in the United States. It is one of the main leading causes of death in US and the world.
In this project, we will try to find the best machine learning model that can predict the presence of CAD using available patients data.
We will use (Heart Disease UCI) data set from kaggle downloaded from this link

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

The data is provided by Cleveland Clinic and contains 14 features related to 303 patients.

### Description of the data set features

- age: age in years
- sex: sex (1 = male; 0 = female)
- cp: chest pain type
  Value 1: typical angina
  Value 2: atypical angina
  Value 3: non-anginal pain
- trestbps: resting blood pressure (in mm Hg on admission to the hospital)
- chol: serum cholesterol in mg/dl
- fbs: (fasting blood sugar > 120 mg/dl) (1 = true; 0 = false)
- restecg: resting electrocardiographic results
  Value 0: normal
  Value 1: having ST-T wave abnormality (T wave inversions and/or ST elevation or depression of > 0.05 mV)
  Value 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
  Value 1: upsloping

Value 2: flat
Value 3: downsloping

- ca: number of major vessels (0-3) colored by flourosopy
- thal: 1 = normal; 1 = fixed heart defect; 3 = reversable heart defect
- target: diagnosis of heart disease (angiographic disease status)
  Value 0: $< 50\%$ diameter narrowing
  Value 1: $> 50\%$ diameter narrowing

The target feature is the feature that we will try to predict in this project.

### Summary of the project

The aim of the project is to find the best machine learning model that predicts the presence of CAD (target = 1) with overall accuracy $> 85\%$ and returns the highest F1 score.
F1 score is the harmonic average of specificity (Precision) and sensitivity (recall) and it is calculated using the following equation

$$2 \times \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

The following models will be used:
* Logistic Regression
* Linear Discriminant Analysis
* Quadratic Discriminant Analysis
* K-Nearest Neighbors
* Random Forest

## Methods/Analysis

We will start by loading the heart data and the needed packages

```
library(tidyverse)
library(caret)
#Loading data
heart_data <- read.csv('heart.csv')
```

### Data Exploration

```
dim(heart_data)
```

```
## [1] 303  14
```

The data set contains 303 observations and 14 variables.

The following code will return the data structure.

```
str(heart_data)
```

```
## 'data.frame':    303 obs. of  14 variables:
##  $ age     : int  63 37 41 56 57 57 56 44 52 57 ...
##  $ sex     : int  1 1 0 1 0 1 0 1 1 1 ...
##  $ cp      : int  3 2 1 1 0 0 1 1 2 2 ...
##  $ 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 ...
```

Before continuing our analysis, we want to make sure that our data set does not contain missing values.

```
sum(is.na(heart_data) == TRUE)
```

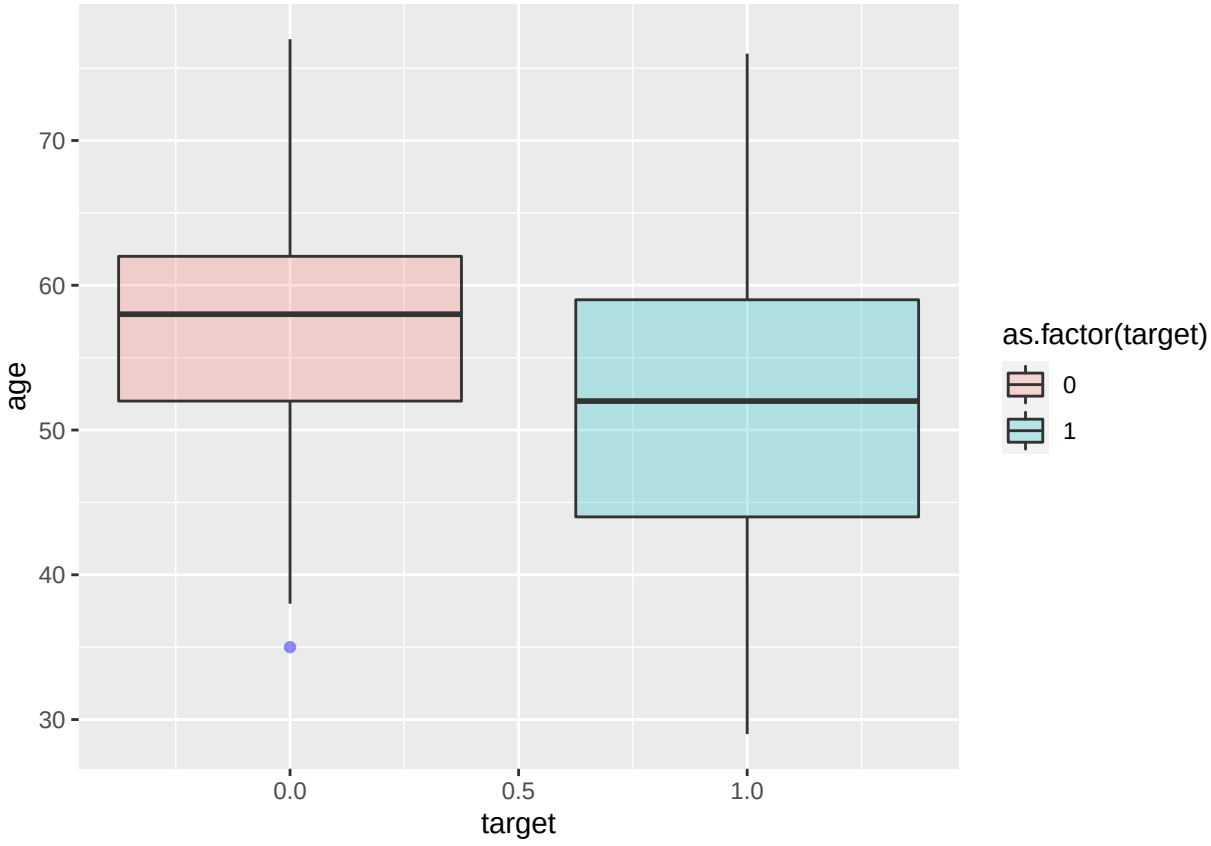
```
## [1] 0
```

There are no missing values in the data set.

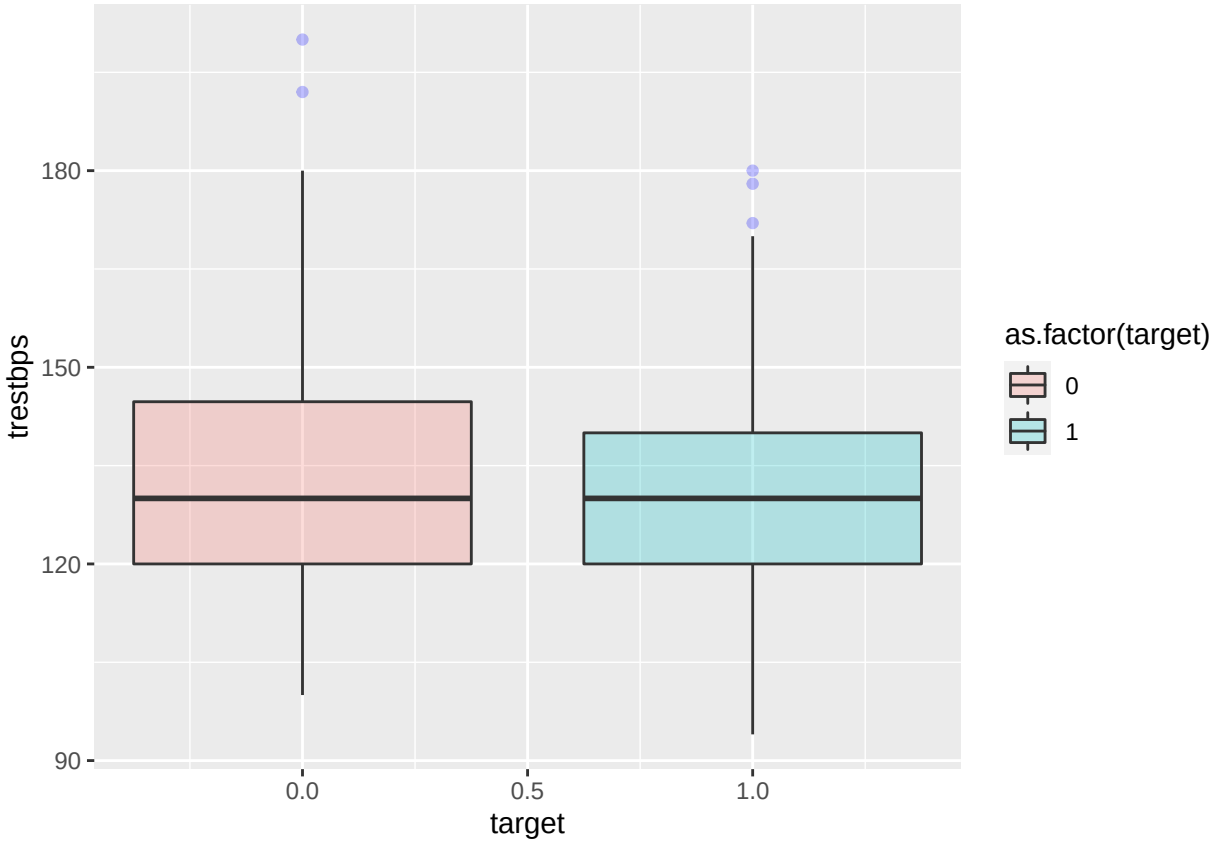
## Data Visualization

Now we will explore our data visually to examine the relationship between the different variables and the target variable (Patient has CAD = 1 or Patient does not have CAD = 0)

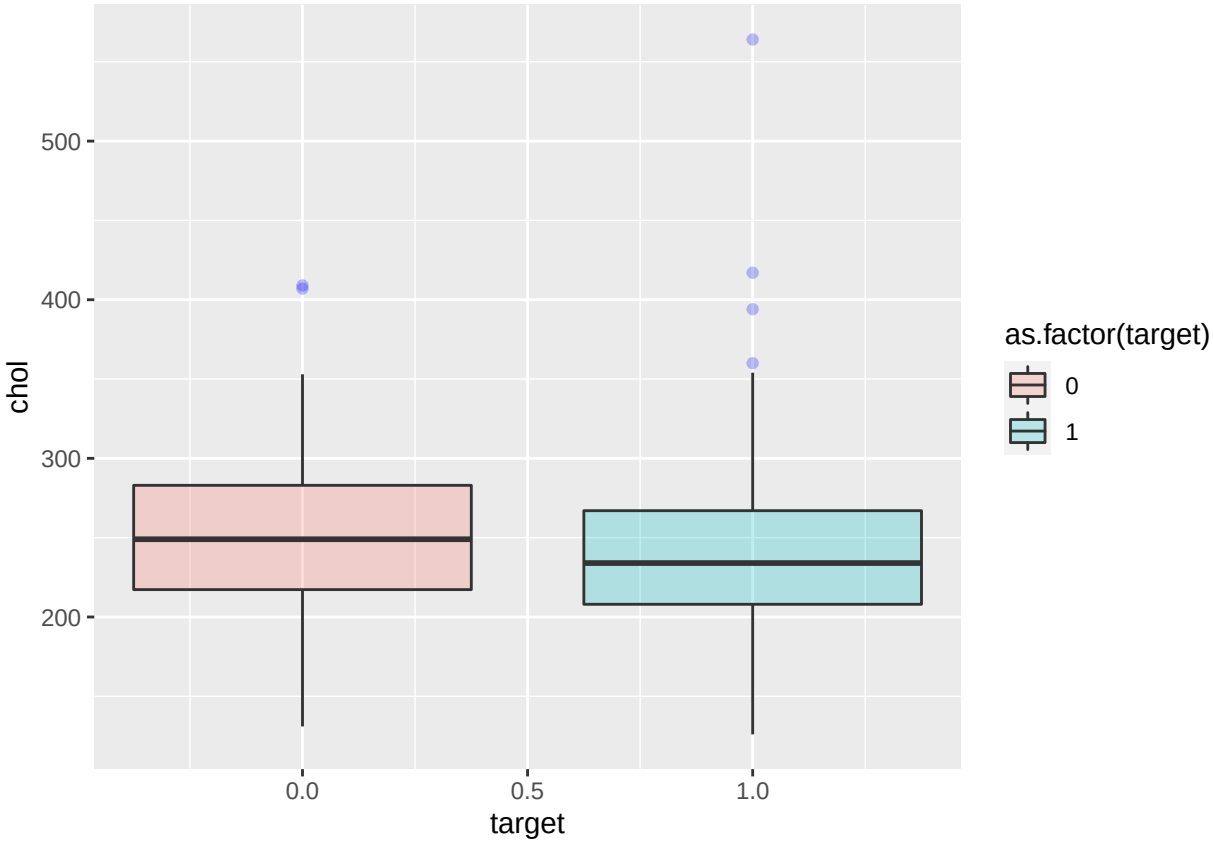
**The relationship between age and presence of CAD**



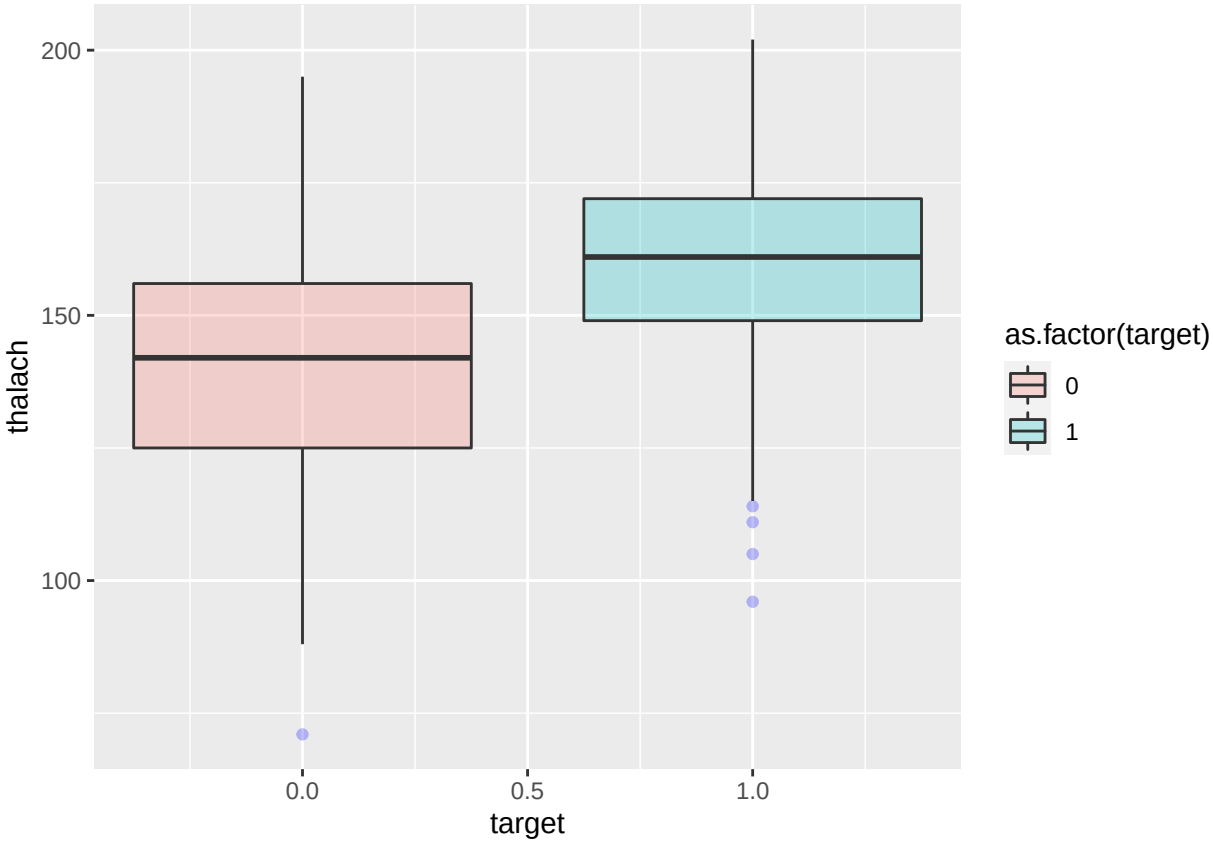
**The relationship between resting blood pressure and presence of CAD**



**The relationship between serum cholesterol and presence of CAD**



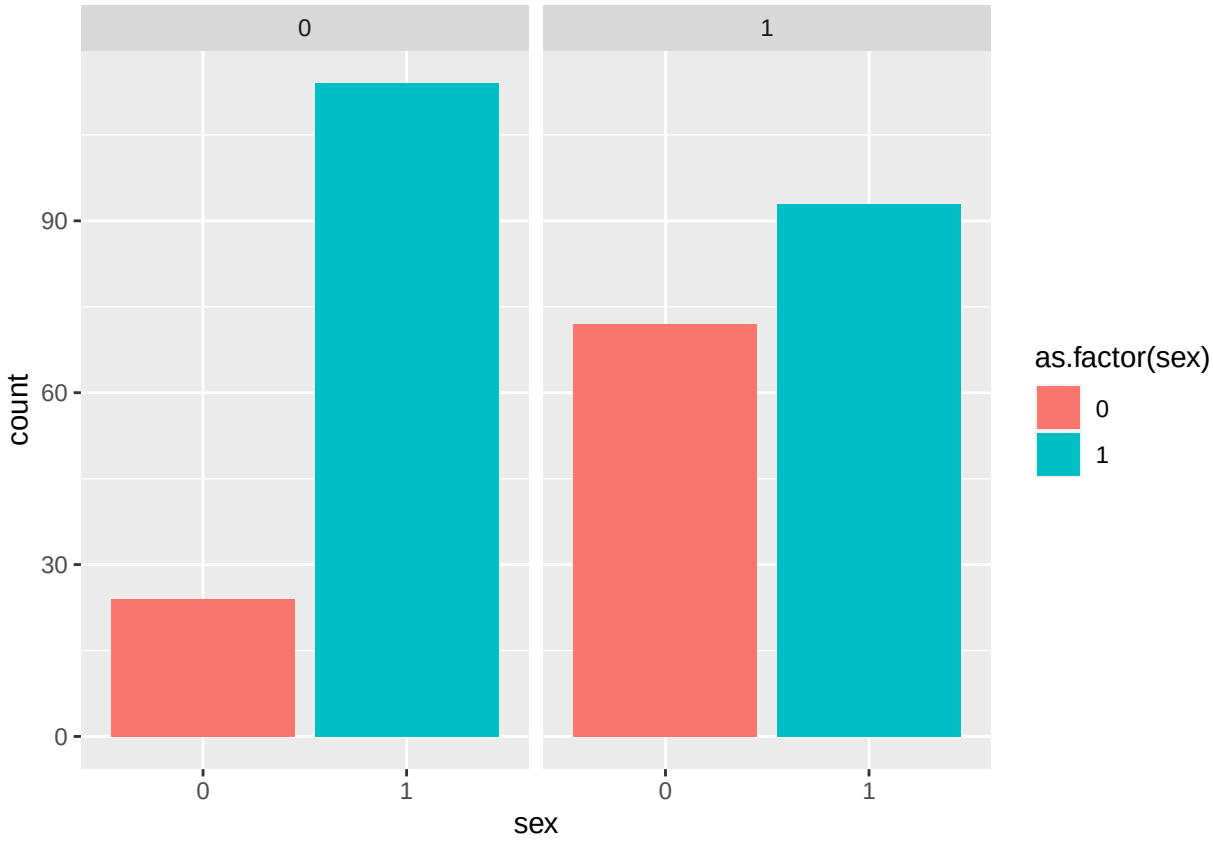
**The relationship between maximum heart rate and presence of CAD**




From above plots we can find that the effects of these variables are not significant in determining the possibility of having CAD.

Now we will explore more relationships in the data set.

**The relationship between patient's sex and presence of CAD**



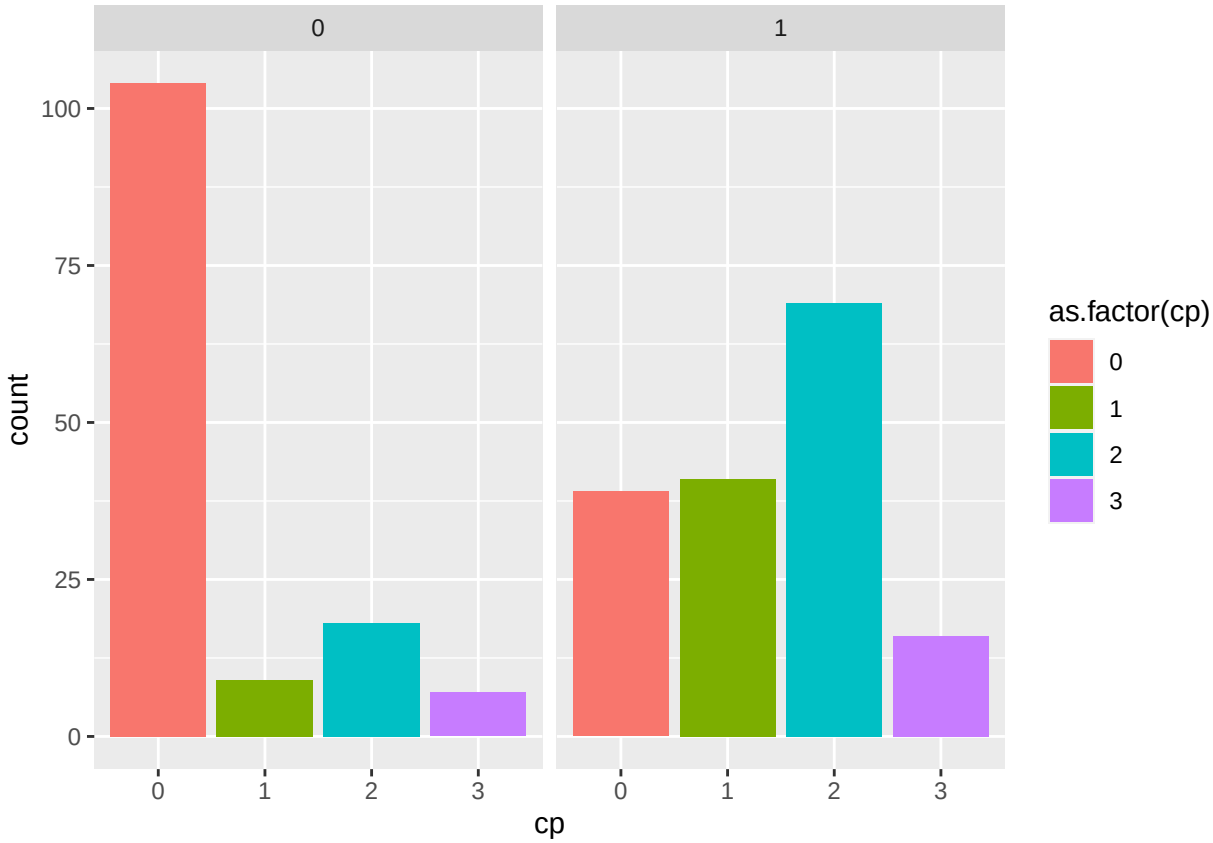

The relationship here is not significant because the data set contains data for male patients more than females.

```
table(heart_data$sex)
```

```
##
##   0   1
##  96 207
```

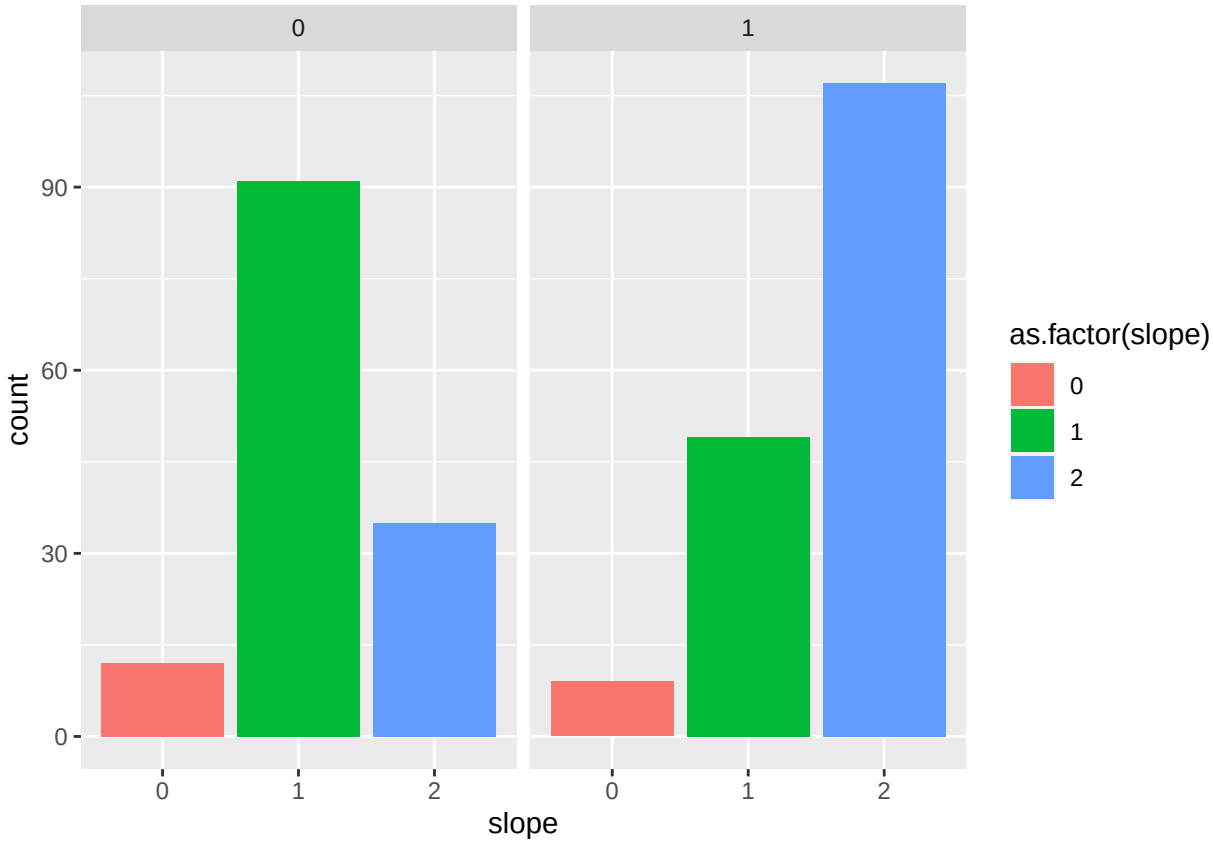
Number of male patients is almost double the number of females.

**The relationship between chest pain type and presence of CAD**

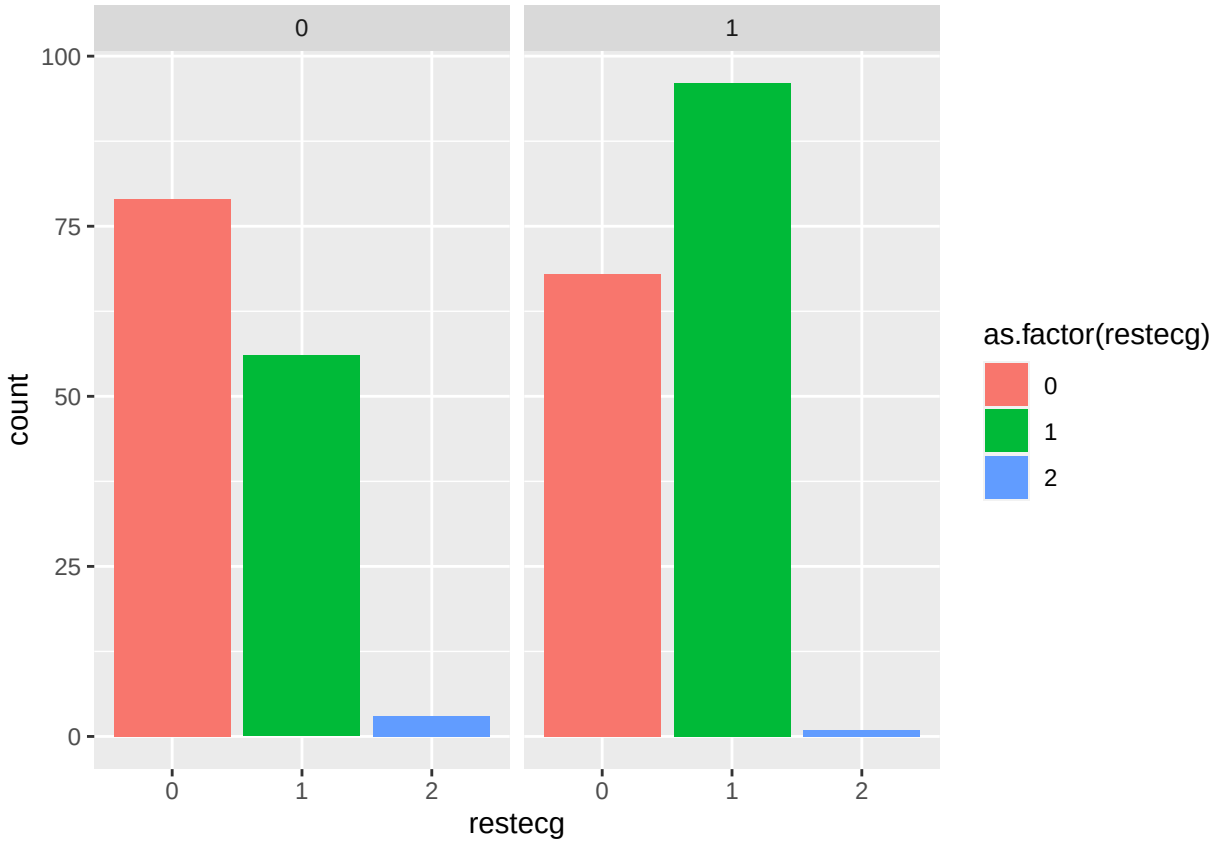



This plot is showing a significant relationship between chest pain and presence of CAD.

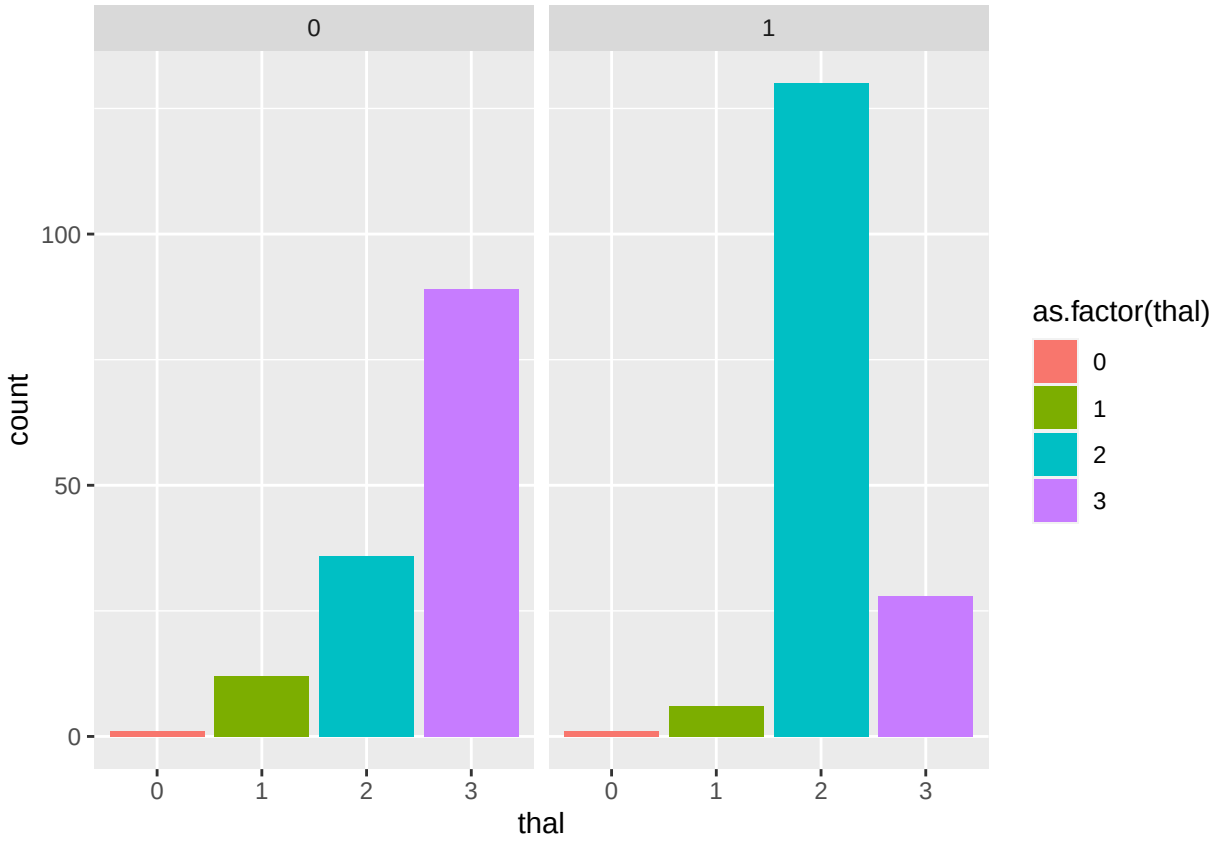
**The relationship between slope of the peak exercise ST segment and presence of CAD**



**The relationship between resting electrocardiographic results and presence of CAD**



**The relationship between having a heart defect and presence of CAD**




Also this plot is showing a significant relationship between having fixed heart defect and presence of CAD.

## Creating test and train data sets

```
set.seed(10,sample.kind = "Rounding")
test_index <- createDataPartition(heart_data$target ,times = 1, p=.3,list = FALSE)
train_heart <- heart_data[-test_index, ]
test_heart <- heart_data[test_index, ]
```

## Setting cross validation parameters

```
control <- trainControl(method = "cv", number = 10, p = .9)
```

After creating test and train data sets from heart data set and setting the cross validation parameters, we will start applying different machine learning models to our data and add the results of total accuracy and F1 score of each model to a results table.

## Model 1 - Logistic regression

```
set.seed(10, sample.kind="Rounding")
train_glm <- train(as.factor(target) ~ ., method = "glm",
                   data = train_heart,family = "binomial",
                   trControl = control)
glm_pred <- predict(train_glm, test_heart)
```

```
logistic_regression <- confusionMatrix(glm_pred, as.factor(test_heart$target),
                                       mode = "everything",
                                       positive = "1")
logistic_regression
```

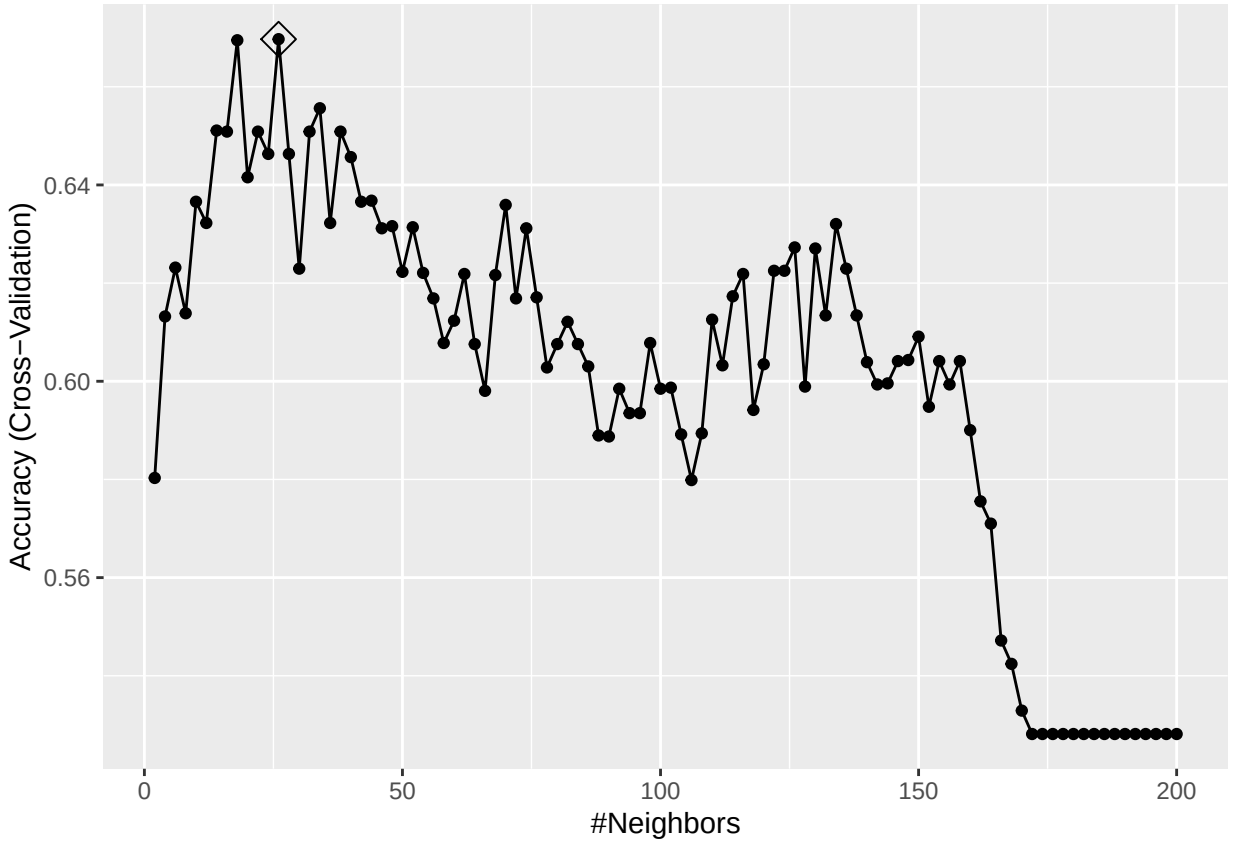
```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 28  6
##          1 10 47
##
##                Accuracy : 0.8242
##                  95% CI : (0.7302, 0.896)
##     No Information Rate : 0.5824
##     P-Value [Acc > NIR] : 7.693e-07
##
##                   Kappa : 0.6331
##
##  Mcnemar's Test P-Value : 0.4533
##
##             Sensitivity : 0.8868
##             Specificity : 0.7368
##          Pos Pred Value : 0.8246
##          Neg Pred Value : 0.8235
##               Precision : 0.8246
##                  Recall : 0.8868
##                      F1 : 0.8545
##              Prevalence : 0.5824
##          Detection Rate : 0.5165
##    Detection Prevalence : 0.6264
##       Balanced Accuracy : 0.8118
##
##        'Positive' Class : 1
##
```

```
results <- tibble(Model = "logistic_regression",
                  Accuracy = logistic_regression$overall["Accuracy"],
                  F1 = logistic_regression$byClass["F1"])
results %>% knitr::kable()
```

| Model | Accuracy | F1 |
|---|---|---|
| logistic_regression | 0.8241758 | 0.8545455 |

## Model 2 - K-nearest neighbors

```
set.seed(10, sample.kind="Rounding")
train_knn <- train(as.factor(target) ~ ., method = "knn",
                   data = train_heart,
                   tuneGrid = data.frame(k = seq(2, 200, 2)),
                   trControl = control)
#Find best tune
ggplot(train_knn, highlight = TRUE)
```




```
train_knn$bestTune
```

```
##     k
## 13 26
```

```
knn_pred <- predict(train_knn, test_heart)

knn <- confusionMatrix(knn_pred, as.factor(test_heart$target),
                       mode = "everything",
                       positive = "1")
knn
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 24 13
##          1 14 40
##
##                Accuracy : 0.7033
##                  95% CI : (0.5984, 0.7945)
##     No Information Rate : 0.5824
##     P-Value [Acc > NIR] : 0.01175
##
##                   Kappa : 0.3877
##
##  Mcnemar's Test P-Value : 1.00000
```

```
##
##             Sensitivity : 0.7547
##             Specificity : 0.6316
##          Pos Pred Value : 0.7407
##          Neg Pred Value : 0.6486
##               Precision : 0.7407
##                  Recall : 0.7547
##                      F1 : 0.7477
##              Prevalence : 0.5824
##          Detection Rate : 0.4396
##    Detection Prevalence : 0.5934
##       Balanced Accuracy : 0.6931
##
##        'Positive' Class : 1
##
```

```
results <- bind_rows(results,
                     data_frame(Model = "K-nearest neighbors",
                                Accuracy = knn$overall["Accuracy"],
                                F1 = knn$byClass["F1"]))
results %>% knitr::kable()
```

| Model | Accuracy | F1 |
|---|---|---|
| logistic_regression | 0.8241758 | 0.8545455 |
| K-nearest neighbors | 0.7032967 | 0.7476636 |

### Model 3 - Linear discriminant analysis

```
set.seed(10, sample.kind="Rounding")
train_lda <- train(as.factor(target) ~ ., method = "lda", data = train_heart,
                   trControl = control)
lda_pred <- predict(train_lda, test_heart)

lda <- confusionMatrix(lda_pred, as.factor(test_heart$target),
                       mode = "everything",
                       positive = "1")
lda
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 28  5
##          1 10 48
##
##                Accuracy : 0.8352
##                  95% CI : (0.7427, 0.9047)
##     No Information Rate : 0.5824
##     P-Value [Acc > NIR] : 2.194e-07
##
##                   Kappa : 0.6547
##
##  Mcnemar's Test P-Value : 0.3017
```

```
##
##             Sensitivity : 0.9057
##             Specificity : 0.7368
##          Pos Pred Value : 0.8276
##          Neg Pred Value : 0.8485
##               Precision : 0.8276
##                  Recall : 0.9057
##                      F1 : 0.8649
##              Prevalence : 0.5824
##          Detection Rate : 0.5275
##    Detection Prevalence : 0.6374
##       Balanced Accuracy : 0.8213
##
##        'Positive' Class : 1
##
```

```
results <- bind_rows(results,
                     data_frame(Model = "Linear discriminant analysis",
                                Accuracy = lda$overall["Accuracy"],
                                F1 = lda$byClass["F1"]))
results %>% knitr::kable()
```

| Model | Accuracy | F1 |
|---|---|---|
| logistic_regression | 0.8241758 | 0.8545455 |
| K-nearest neighbors | 0.7032967 | 0.7476636 |
| Linear discriminant analysis | 0.8351648 | 0.8648649 |

## Model 4 - Quadratic discriminant analysis

```
set.seed(10, sample.kind="Rounding")
train_qda <- train(as.factor(target) ~ ., method = "qda", data = train_heart,
                   trControl = control)
qda_pred <- predict(train_qda, test_heart)

qda <- confusionMatrix(qda_pred, as.factor(test_heart$target),
                       mode = "everything",
                       positive = "1")
qda
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 31  6
##          1  7 47
##
##                Accuracy : 0.8571
##                  95% CI : (0.7681, 0.9217)
##     No Information Rate : 0.5824
##     P-Value [Acc > NIR] : 1.415e-08
##
##                   Kappa : 0.7052
##
```

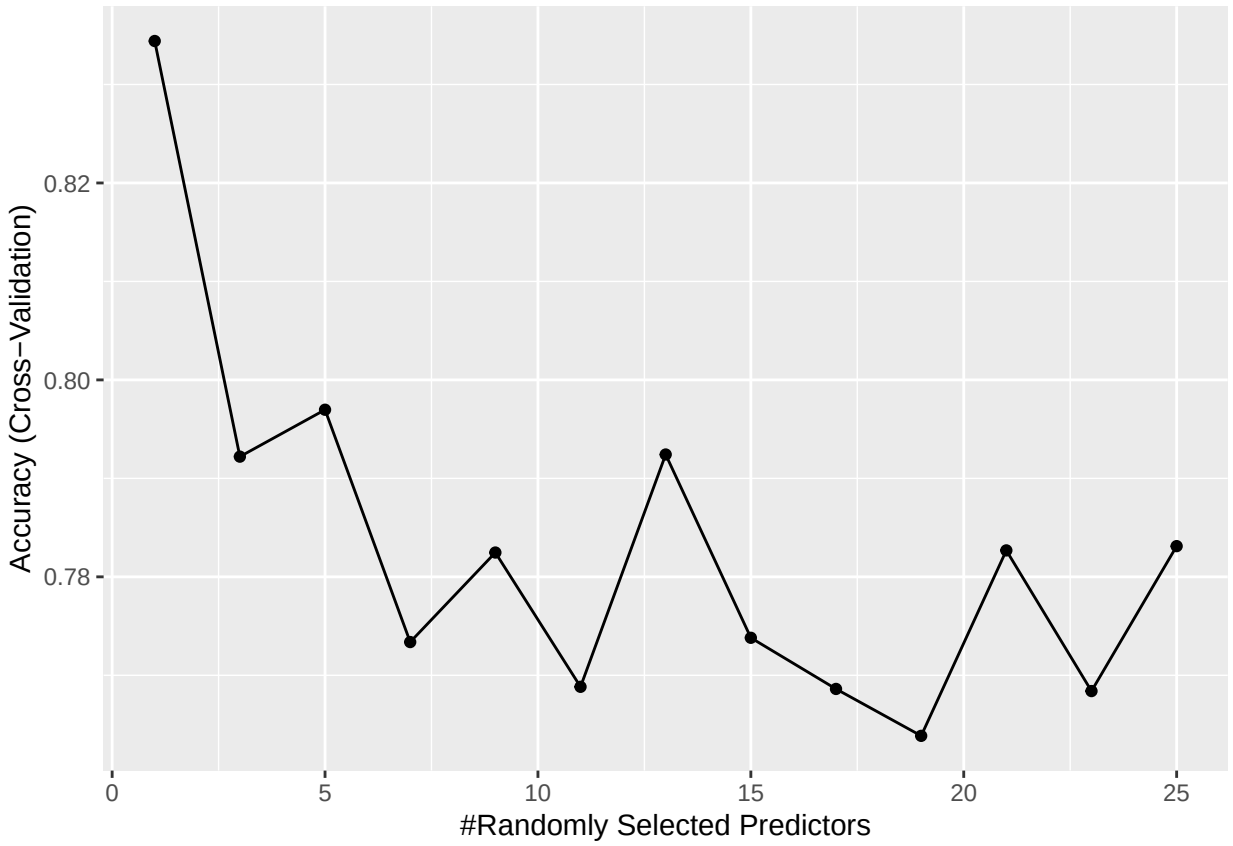
```
##  Mcnemar's Test P-Value : 1
##
##             Sensitivity : 0.8868
##             Specificity : 0.8158
##          Pos Pred Value : 0.8704
##          Neg Pred Value : 0.8378
##               Precision : 0.8704
##                  Recall : 0.8868
##                      F1 : 0.8785
##              Prevalence : 0.5824
##          Detection Rate : 0.5165
##    Detection Prevalence : 0.5934
##       Balanced Accuracy : 0.8513
##
##        'Positive' Class : 1
##
```

```
results <- bind_rows(results,
                     data_frame(Model = "Quadratic discriminant analysis",
                                Accuracy = qda$overall["Accuracy"],
                                F1 = qda$byClass["F1"]))
results %>% knitr::kable()
```

| Model | Accuracy | F1 |
|---|---|---|
| logistic_regression | 0.8241758 | 0.8545455 |
| K-nearest neighbors | 0.7032967 | 0.7476636 |
| Linear discriminant analysis | 0.8351648 | 0.8648649 |
| Quadratic discriminant analysis | 0.8571429 | 0.8785047 |

### Model 5 - Random Forest

```
set.seed(10, sample.kind="Rounding")
train_rf <- train(as.factor(target) ~ ., method = "rf",
                  data = train_heart,ntree = 100,
                  tuneGrid = data.frame(mtry =seq(1,25,2)),
                  trControl = control,
                  importance = TRUE)
ggplot(train_rf)
```




```
train_rf$bestTune
```

```
##   mtry
## 1    1
```

```
rf_pred <- predict(train_rf, test_heart)
rf <- confusionMatrix(rf_pred, as.factor(test_heart$target),
                mode = "everything",
                positive = "1")
rf
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 29  4
##          1  9 49
##
##                Accuracy : 0.8571
##                  95% CI : (0.7681, 0.9217)
##     No Information Rate : 0.5824
##     P-Value [Acc > NIR] : 1.415e-08
##
##                   Kappa : 0.7007
##
##  Mcnemar's Test P-Value : 0.2673
##
##             Sensitivity : 0.9245
```

```
##             Specificity : 0.7632
##          Pos Pred Value : 0.8448
##          Neg Pred Value : 0.8788
##               Precision : 0.8448
##                  Recall : 0.9245
##                      F1 : 0.8829
##              Prevalence : 0.5824
##          Detection Rate : 0.5385
##    Detection Prevalence : 0.6374
##       Balanced Accuracy : 0.8438
##
##        'Positive' Class : 1
##
```

```
results <- bind_rows(results,
                     data_frame(Model = "Random Forest",
                                Accuracy = rf$overall["Accuracy"],
                                F1 = rf$byClass["F1"]))
results %>% knitr::kable()
```

| Model | Accuracy | F1 |
|---|---|---|
| logistic_regression | 0.8241758 | 0.8545455 |
| K-nearest neighbors | 0.7032967 | 0.7476636 |
| Linear discriminant analysis | 0.8351648 | 0.8648649 |
| Quadratic discriminant analysis | 0.8571429 | 0.8785047 |
| Random Forest | 0.8571429 | 0.8828829 |

We can explore the most important factors that predict the presence of CAD using the variable importance function

```
varImp(train_rf)
```

```
## rf variable importance
##
##          Importance
## cp         100.0000
## ca          78.2180
## thal        73.7407
## slope       64.6573
## exang       63.8281
## thalach     55.9007
## oldpeak     55.7777
## sex         32.1908
## age         23.5400
## trestbps    12.8170
## restecg      7.1424
## chol         0.5068
## fbs          0.0000
```

## Conclusion

This project involved applying 5 machine learning models to the heart disease data set from University of California Irvine machine learning repository, in order to predict the presence of Coronary Artery Disease

(CAD). The aim of the project is to find a model with overall accuracy $> 85\%$ and returns the highest F1 score.
Two models returned over all accuracy $> 85$ %, Quadratic discriminant analysis and Random Forest both returned 85.7% accuracy. Random Forest model returned the highest F1 score 88.3. Using the variable importance function in Random Forest, the most important factors to determine the presence of CAD are the presence and type of chest pain, Number of major vessels that are working, and the presence of heart defect. These factors are consistent with our previous analysis that was done using data visualization.