HarvardX Data Science Capstone Project: Heart Disease

Matthew Joseph Diliberto

16 February 2021

Project Scope and Objective

This project represents the final step in the nine-course Data Science series, offered by HarvardX.

The objective of this project is to "apply machine learning techniques that go beyond standard linear regression". The structure of the report will be the following:

1. **Introduction to Database**
2. **Exploratory Data Analysis and Visualizations**
3. **Application of Machine Learning Techniques**
4. **Conclusions**

Introduction to Database

In the project, I will be analyzing the **Cleveland Heart Disease Data Set**. Having worked in the pharmaceutical sector for the past 5 years, I have chosen this area of research to further deepen my understanding of the underlying causes of the disease and how machine learning techniques can be applied.

The data set provided contains **303 instances** of patients reporting the presence or absence of heart disease. We are thus going to be dealing with a classification problem where we are trying to make a **prediction on whether the patients present heart disease or not**.

In order to make these predictions, we will be leveraging **the other 13 variables in the data set**, which provide more details about the patients conditions.

The complete list of the variables is the following:

0. **Heart Disease** (*Target Variable*). The original data presents 5 possible values: 0 represents the absence of heart disease whereas values 1, 2, 3 and 4 indicate the presence of heart disease. For the purpose of this project, the variable has been converted into a binary variable (**Yes**: *heart disease is present*, **No**: *heart disease is not present*).
1. **Age** of the patient.
2. **Sex** of the patient.
3. **Chest Pain Type**. Values range from 1-4 (1: *Typical Angina*, 2: *Atypical Angina*, 3: *Non-Anginal Pain*, 4: *Asymptomatic*)
4. **Blood Pressure at rest**
5. **Cholesterol level**
6. **Blood Sugar whilst fasting**
7. **ECG at rest**
8. **Maximum Heart Rate achieved**
9. **Exercise Induced Angina**
10. **Old Peak**: *ST depression induced by exercise relative to rest*
11. **Slope**: *the slope of the peak exercise ST segment*
12. **Number of Vessels** *coloured by flouroscopy*
13. **Defect Presence** *with possible values including: a) normal, b) fixed defect, c) reversable defect

Furthermore, we will be dividing the data into a 80/20 split between training and test data. The performance of the models that will be developed will be evaluated on the test data.

Exploratory Data Analysis and Data Visualizations

In the following section, we will try to acquire a betting understanding of the data and how variables are potentially linked to one another.

```r
# Quick Overview of Initial Data
summary(data_complete)
```

```
##       Age             Sex         Chest_Pain_Type Blood_Pressure_AR
##  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    
##   Cholesterol    Blood_Sugar_F        ECG_AR           Max_HR     
##  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  
##  Angina_Exercise     Old_Peak        Slope       Number_Vessels Defect_Presence
##  Min.   :0.0000   Min.   :0.00   Min.   :1.000   ?  :  4        ?  :  2        
##  1st Qu.:0.0000   1st Qu.:0.00   1st Qu.:1.000   0.0:176        3.0:166        
##  Median :0.0000   Median :0.80   Median :2.000   1.0: 65        6.0: 18        
##  Mean   :0.3267   Mean   :1.04   Mean   :1.601   2.0: 38        7.0:117        
##  3rd Qu.:1.0000   3rd Qu.:1.60   3rd Qu.:2.000   3.0: 20                       
##  Max.   :1.0000   Max.   :6.20   Max.   :3.000                                 
##  Heart_Disease   
##  Min.   :0.0000  
##  1st Qu.:0.0000  
##  Median :0.0000  
##  Mean   :0.9373  
##  3rd Qu.:2.0000  
##  Max.   :4.0000  
```

We can immediately notice than some variables (Number of Vessels, Defect Presence) present instances with values of '?'. For the purposes of the analysis, we will be excluding these values for the final dataset. Furthermore, we will convert the data type to factors where useful.

```r
# Quick Overview of Initial Data
glimpse(data_proc_2)
```

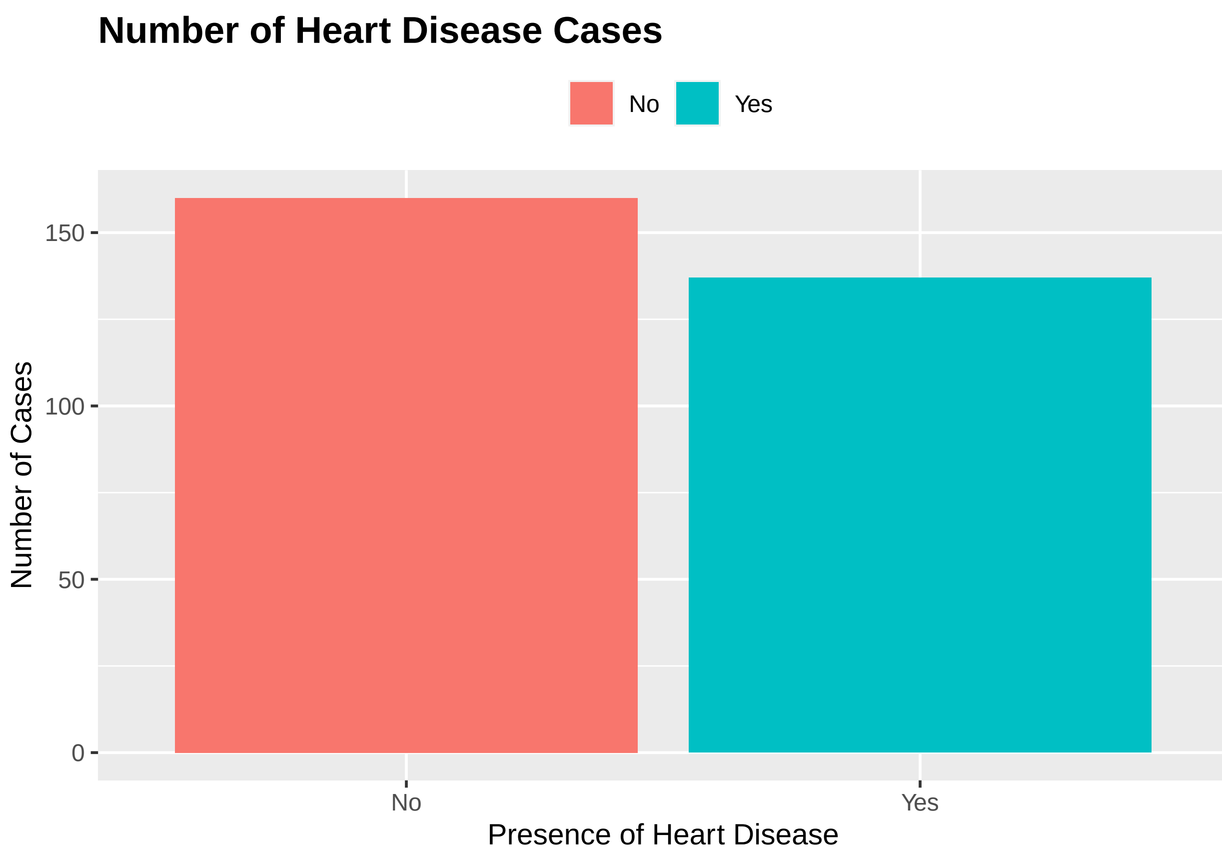
```
## Rows: 297
## Columns: 14
## $ Heart_Disease     <fct> No, Yes, Yes, No, No, No, Yes, No, Yes, Yes, No, ...
## $ Sex               <fct> M, M, M, M, F, M, F, F, M, M, M, F, M, M, M, M, M...
## $ Blood_Sugar_F     <fct> Greater than 120, Lesser or Equal to 120, Lesser ...
## $ Angina_Exercise   <fct> No, Yes, Yes, No, No, No, No, Yes, No, Yes, No, N...
## $ Chest_Pain_Type   <fct> Typical Angina, Asymptomatic, Asymptomatic, Non-A...
## $ ECG_AR            <fct> Probable or Definite, Probable or Definite, Proba...
## $ Slope             <fct> 3, 2, 2, 3, 1, 1, 3, 1, 2, 3, 2, 2, 2, 1, 1, 1, 3...
## $ Number_Vessels    <fct> 0.0, 3.0, 2.0, 0.0, 0.0, 0.0, 2.0, 0.0, 1.0, 0.0,...
## $ Defect_Presence   <fct> 6.0, 3.0, 7.0, 3.0, 3.0, 3.0, 3.0, 3.0, 7.0, 7.0,...
## $ Age               <dbl> 63, 67, 67, 37, 41, 56, 62, 57, 63, 53, 57, 56, 5...
## $ Blood_Pressure_AR <dbl> 145, 160, 120, 130, 130, 120, 140, 120, 130, 140,...
## $ Cholesterol       <dbl> 233, 286, 229, 250, 204, 236, 268, 354, 254, 203,...
## $ Max_HR            <dbl> 150, 108, 129, 187, 172, 178, 160, 163, 147, 155,...
## $ Old_Peak          <dbl> 2.3, 1.5, 2.6, 3.5, 1.4, 0.8, 3.6, 0.6, 1.4, 3.1,...
```

```r
summary(data_proc_2)
```

```
##  Heart_Disease Sex                    Blood_Sugar_F Angina_Exercise
##  No :160       F: 96   Greater than 120      : 43   No :200
##  Yes:137       M:201   Lesser or Equal to 120:254   Yes: 97
##
##
##
##
##        Chest_Pain_Type                  ECG_AR    Slope   Number_Vessels
##  Asymptomatic   :142    Abnormal            :  4   1:139   ?  :  0
##  Atypical Angina: 49    Normal              :147   2:137   0.0:174
##  Non-Anginal    : 83    Probable or Definite:146   3: 21   1.0: 65
##  Typical Angina : 23                                       2.0: 38
##                                                            3.0: 20
##
##  Defect_Presence      Age        Blood_Pressure_AR  Cholesterol
##  ?  :  0         Min.   :29.00   Min.   : 94.0     Min.   :126.0
##  3.0:164         1st Qu.:48.00   1st Qu.:120.0     1st Qu.:211.0
##  6.0: 18         Median :56.00   Median :130.0     Median :243.0
##  7.0:115         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_HR         Old_Peak
##  Min.   : 71.0   Min.   :0.000
##  1st Qu.:133.0   1st Qu.:0.000
##  Median :153.0   Median :0.800
##  Mean   :149.6   Mean   :1.056
##  3rd Qu.:166.0   3rd Qu.:1.600
##  Max.   :202.0   Max.   :6.200
```

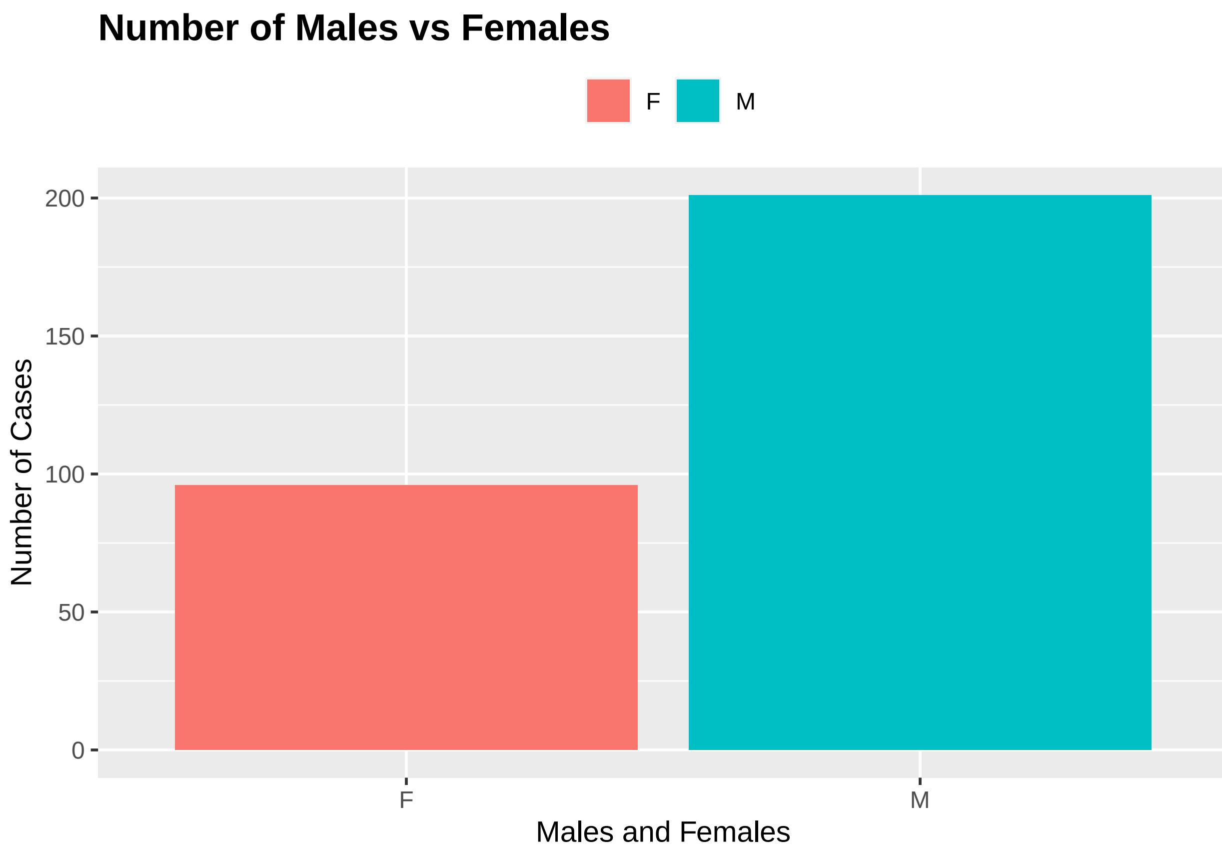
Let us now deep dive into the data through a series of visualizations.

```
# Number of Cases of Heart Disease (Count of Target Variable)
ggplot(data = data_proc_2,
       aes(x= Heart_Disease, fill = Heart_Disease)) +
       geom_bar() +
       ggtitle("Number of Heart Disease Cases") +
       xlab("Presence of Heart Disease") +
       ylab("Number of Cases") +
       theme(legend.position = "top", legend.title = element_blank(),
             plot.title = element_text(size = 14, face = "bold"))
```



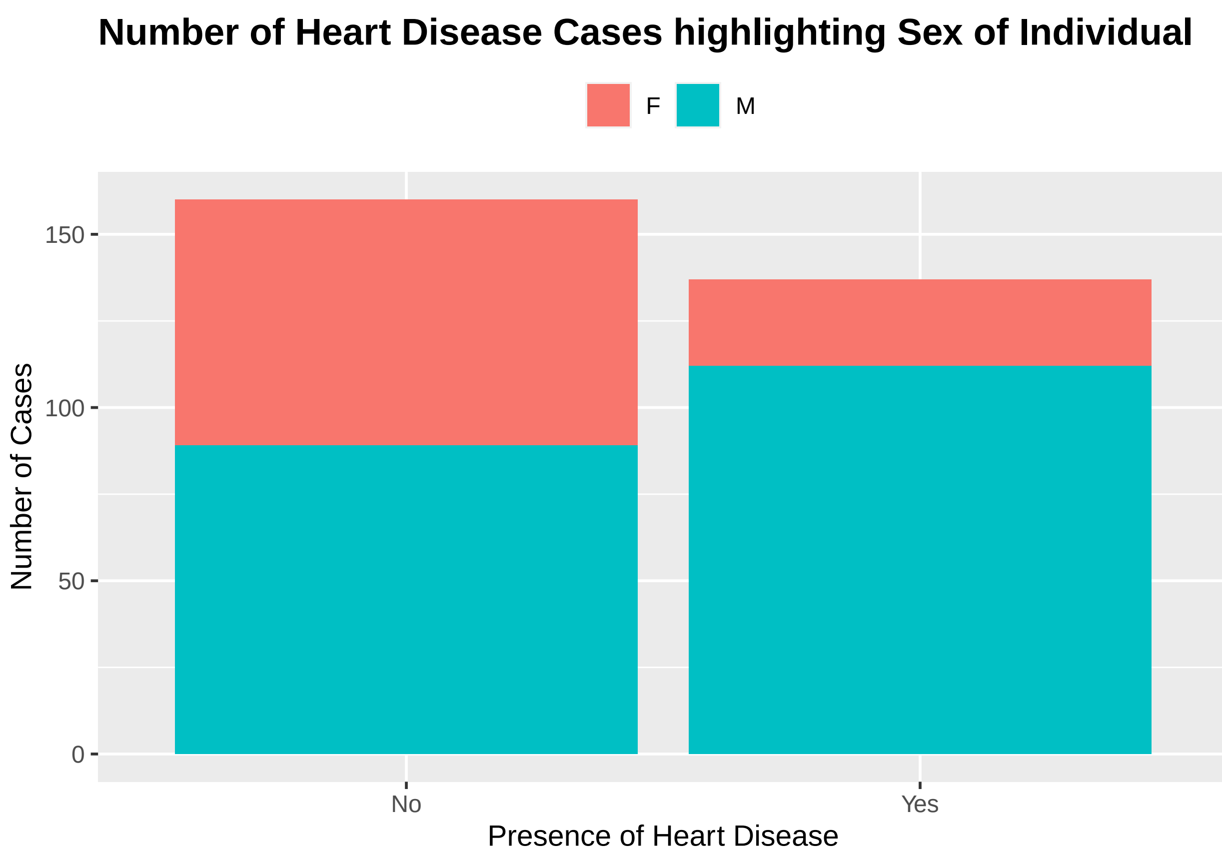
As we can see, our dataset appears to be balanced, with no class prevalence clearly larger than the other class.

```r
# Number of Males and Female
ggplot(data = data_proc_2,
       aes(x= Sex, fill = Sex)) +
       geom_bar() +
       ggtitle("Number of Males vs Females") +
       xlab("Males and Females") +
       ylab("Number of Cases") +
       theme(legend.position = "top", legend.title = element_blank(),
             plot.title = element_text(size = 14, face = "bold"))
```



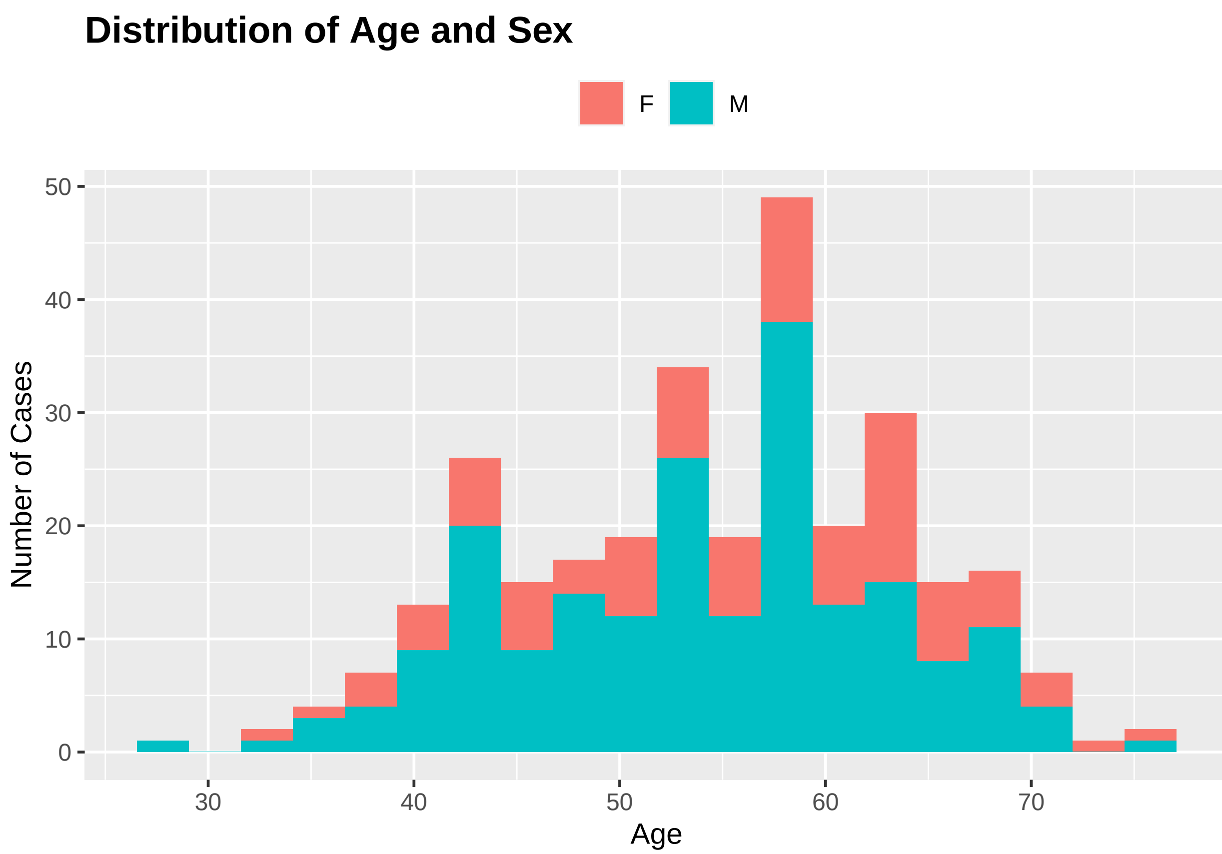
There appears to be a clear indication that there are more males compared to females in the dataset.

```r
# Number of Cases of Heart Disease by Sex
ggplot(data = data_proc_2,
       aes(x= Heart_Disease, fill = Sex)) +
       geom_bar() +
       ggtitle("Number of Heart Disease Cases highlighting Sex of Individual") +
       xlab("Presence of Heart Disease") +
       ylab("Number of Cases") +
       theme(legend.position = "top", legend.title = element_blank(),
             plot.title = element_text(size = 14, face = "bold"))
```

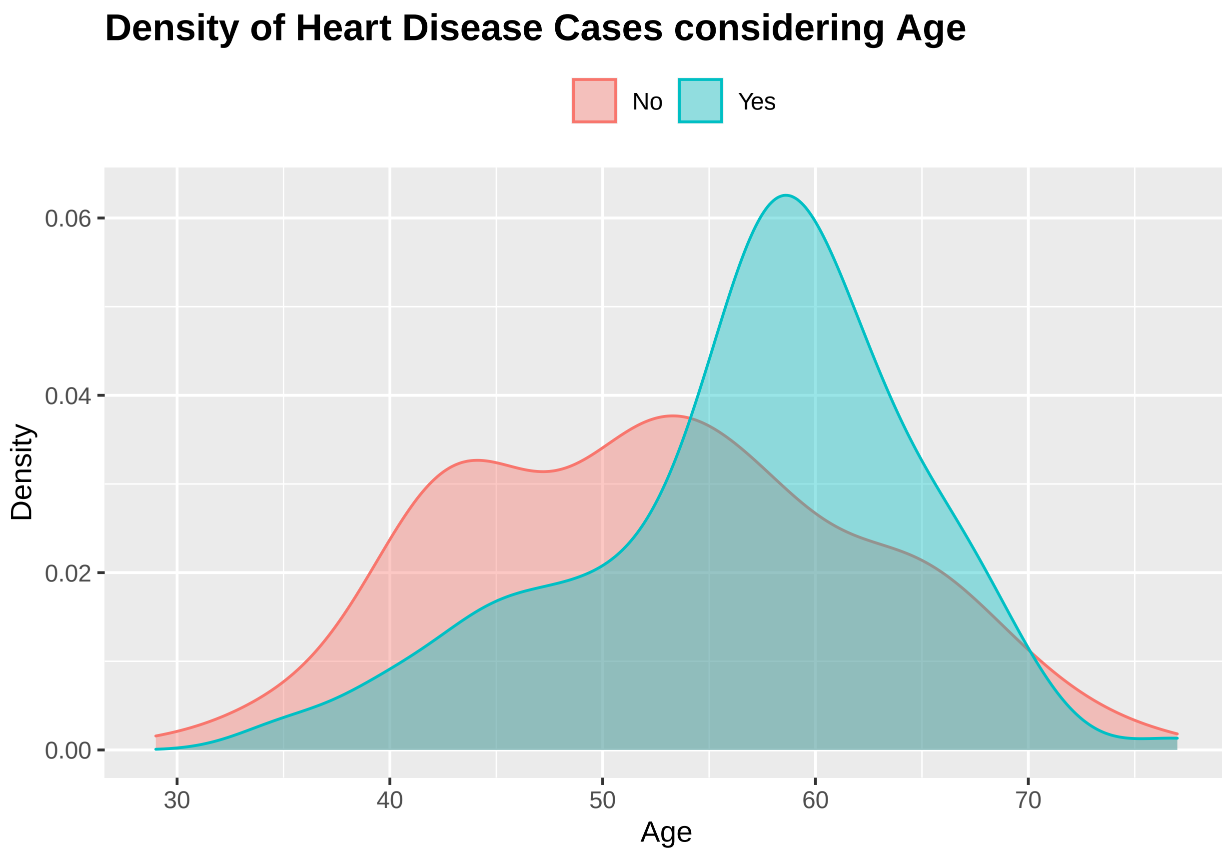


Based on the plot above, it appears that heart disease is a condition that affects men more than females. Sex will certainly be a relevant variable in the model.

```r
# Distribution of Age / Sex
ggplot(data = data_proc_2,
       aes(x= Age, fill = Sex)) +
       geom_histogram(bins = 20) +
       ggtitle("Distribution of Age and Sex") +
       xlab("Age") +
       ylab("Number of Cases") +
       theme(legend.position = "top", legend.title = element_blank(),
             plot.title = element_text(size = 14, face = "bold"))
```

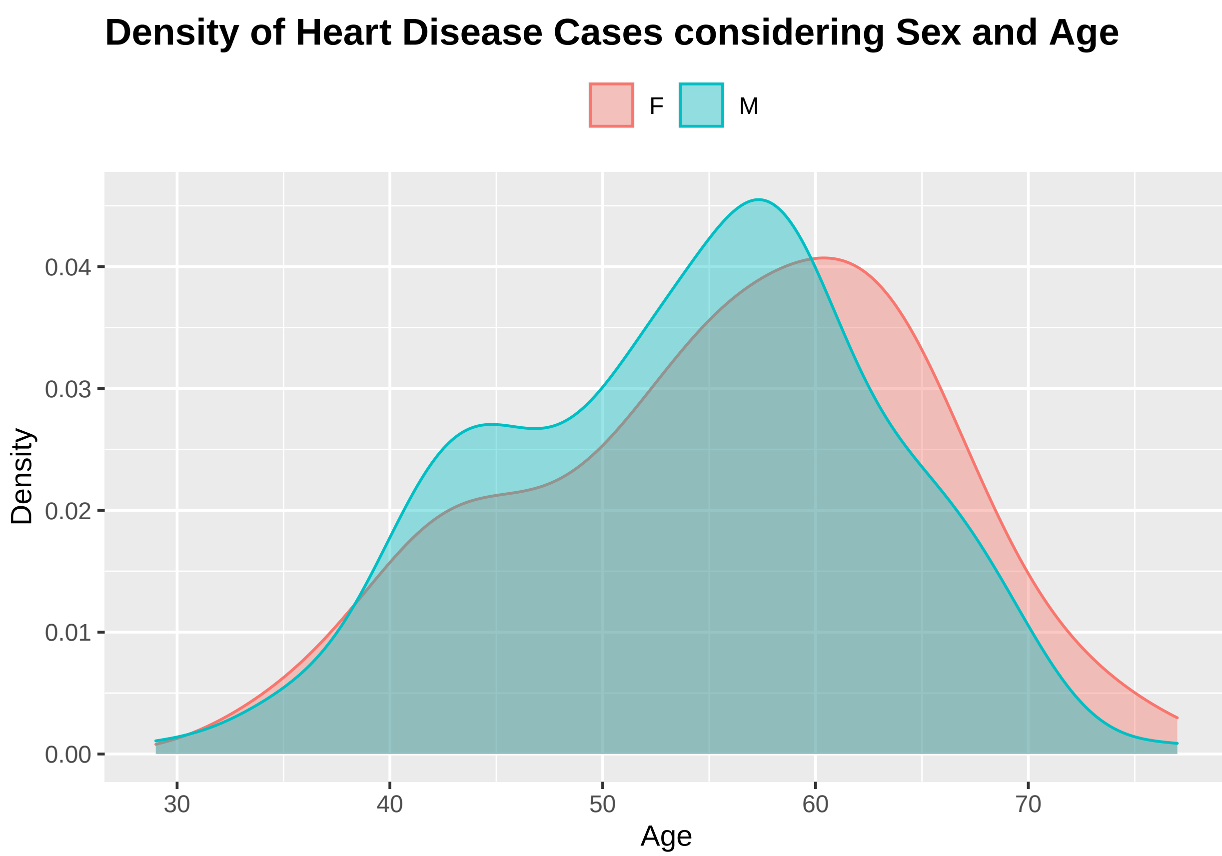


```r
# Density of Heart Disease and Age
ggplot(data = data_proc_2,
       aes(x= Age, col = Heart_Disease, fill = Heart_Disease)) +
  geom_density(alpha = 0.4) +
  ggtitle("Density of Heart Disease Cases considering Age") +
  xlab("Age") +
  ylab("Density") +
  theme(legend.position = "top", legend.title = element_blank(),
        plot.title = element_text(size = 14, face = "bold"))
```

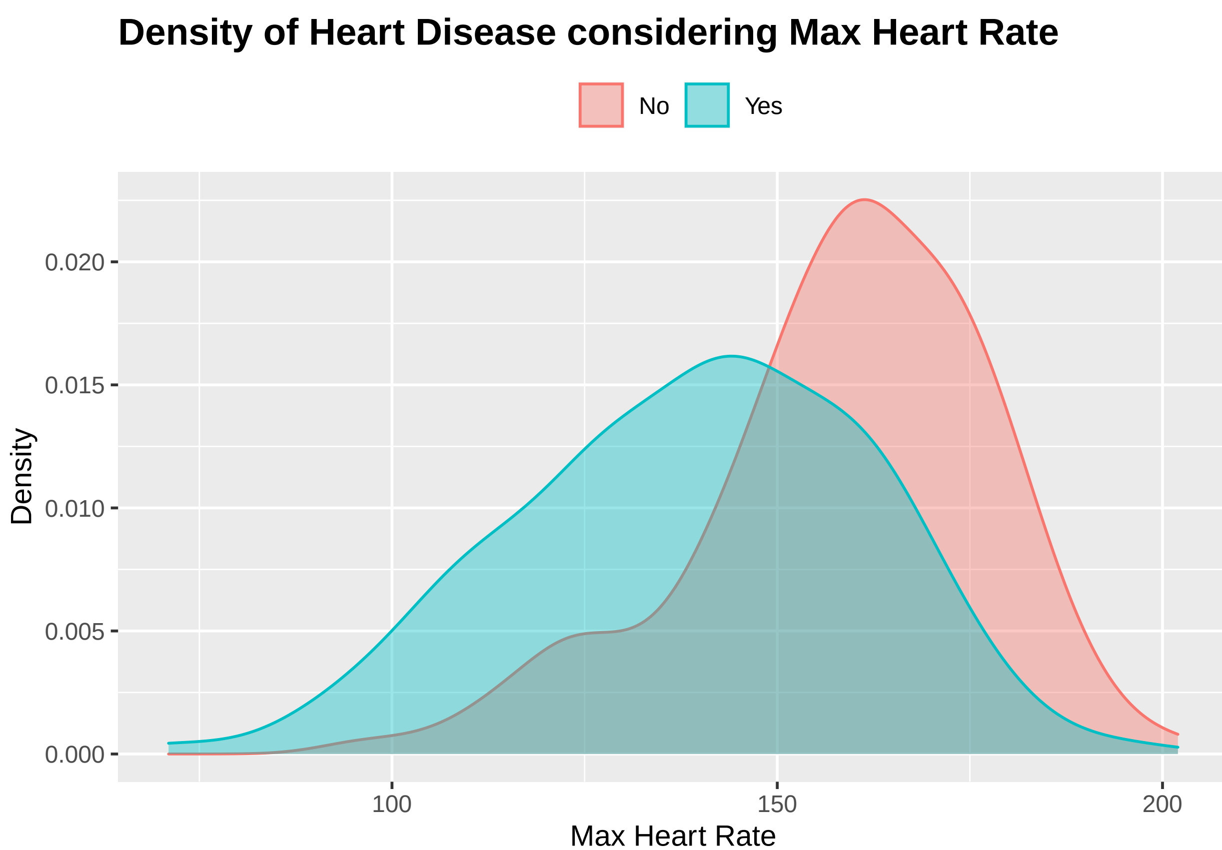


There appears to be a peak of heart disease between 50-70, in particular for males.

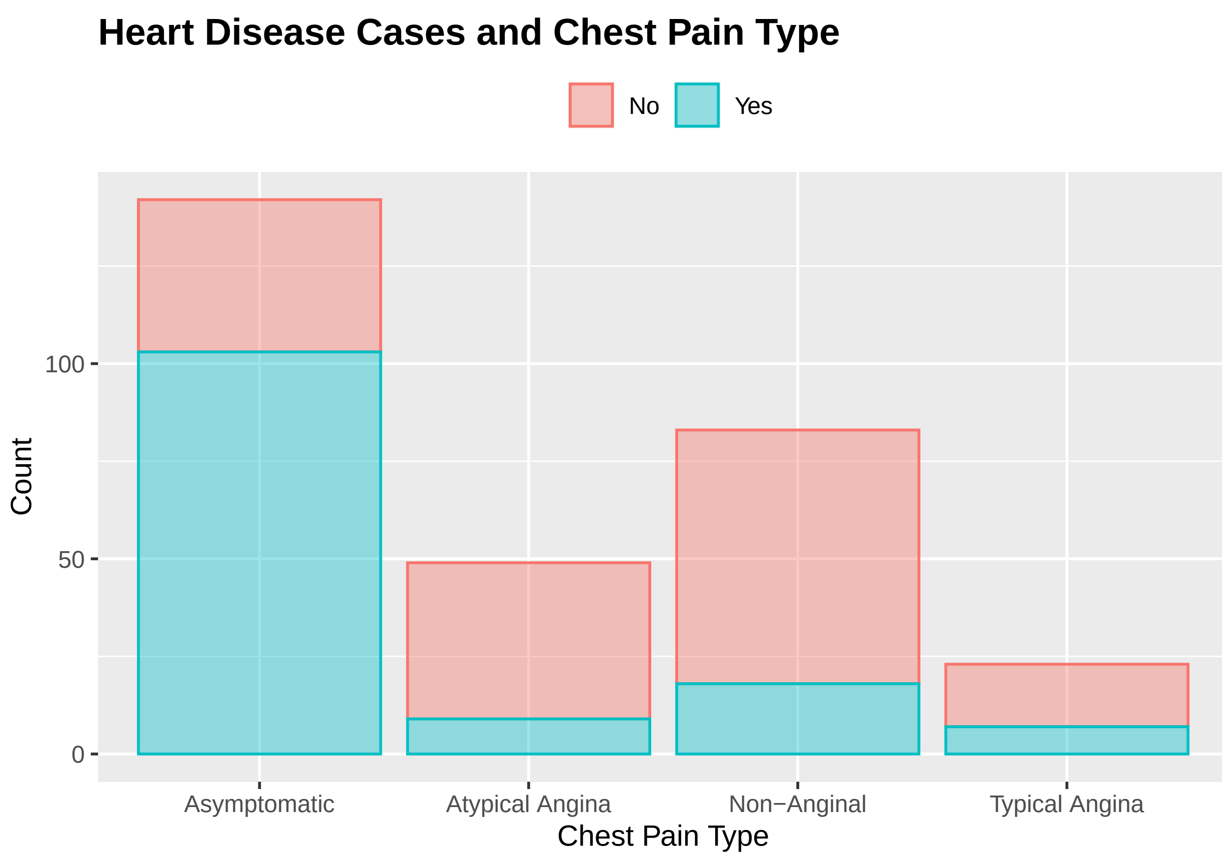
```r
# Density of Heart Disease considering Sex and Age
ggplot(data = data_proc_2,
       aes(x= Age, col = Sex, fill = Sex)) +
       geom_density(alpha = 0.4) +
       ggtitle("Density of Heart Disease Cases considering Sex and Age") +
       xlab("Age") +
       ylab("Density") +
       theme(legend.position = "top", legend.title = element_blank(),
             plot.title = element_text(size = 14, face = "bold"))
```



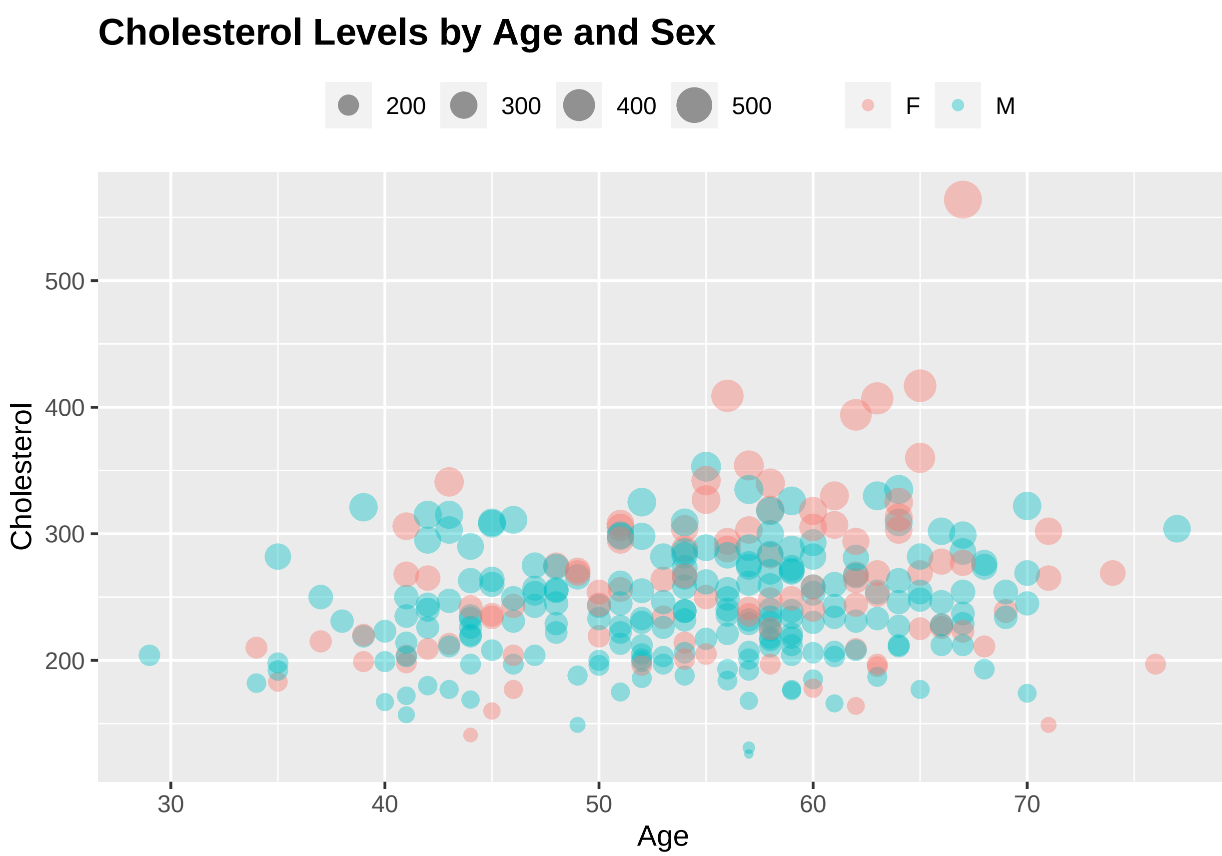
```r
# Density of Heart Disease and Heart Rate
ggplot(data = data_proc_2,
       aes(x= Max_HR, col = Heart_Disease, fill = Heart_Disease)) +
  geom_density(alpha = 0.4) +
  ggtitle("Density of Heart Disease considering Max Heart Rate") +
  xlab("Max Heart Rate") +
  ylab("Density") +
  theme(legend.position = "top", legend.title = element_blank(),
        plot.title = element_text(size = 14, face = "bold"))
```



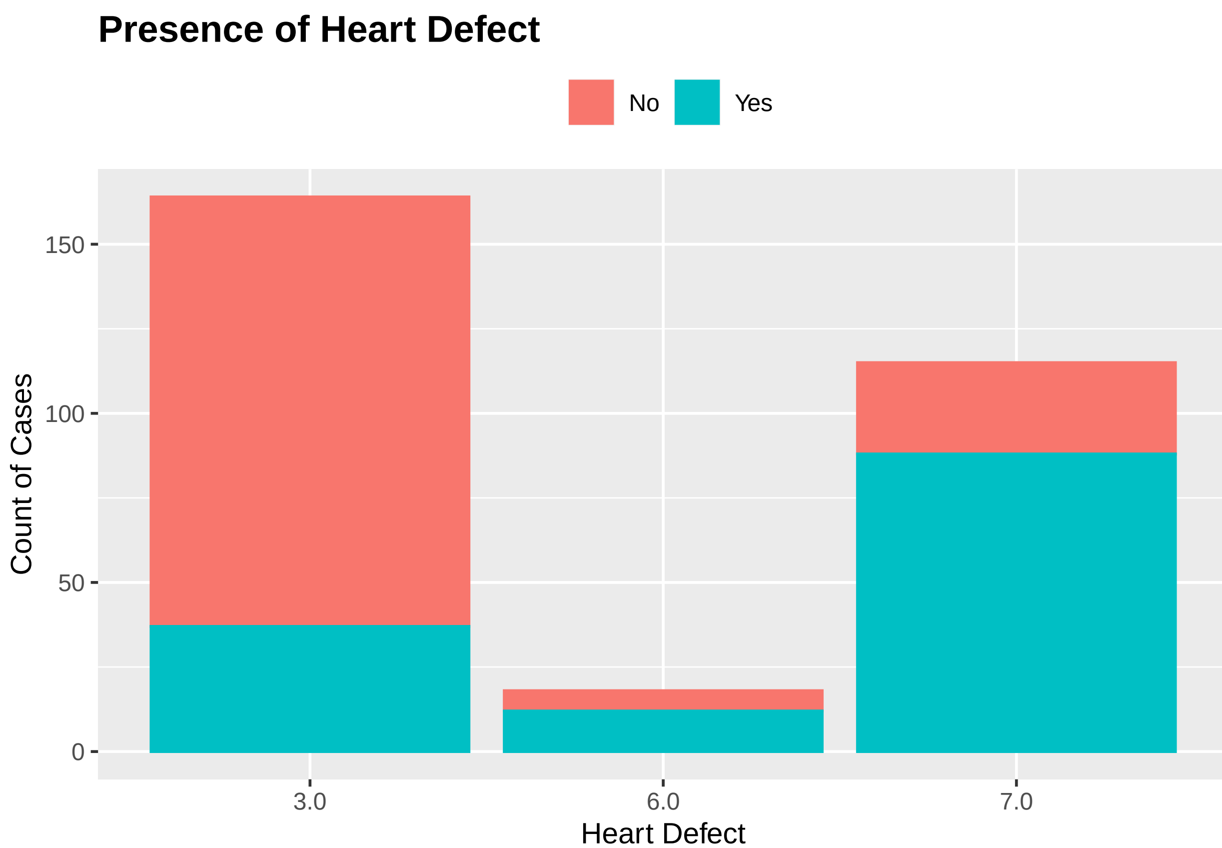
```r
# Density of Heart Disease and Heart Rate
ggplot(data = data_proc_2,
       aes(x= Chest_Pain_Type, col = Heart_Disease, fill = Heart_Disease)) +
  geom_bar(alpha = 0.4) +
  ggtitle("Heart Disease Cases and Chest Pain Type") +
  xlab("Chest Pain Type") +
  ylab("Count") +
  theme(legend.position = "top", legend.title = element_blank(),
        plot.title = element_text(size = 14, face = "bold"))
```



```r
# Cholesterol by Age and Sex
data_proc_2 %>%
  ggplot(aes(x=Age,y=Cholesterol,color=Sex, size=Cholesterol))+
  geom_point(alpha=0.4)+
  ggtitle("Cholesterol Levels by Age and Sex") +
  xlab("Age") +
  ylab("Cholesterol")+
  theme(legend.position = "top", legend.title = element_blank(),
        plot.title = element_text(size = 14, face = "bold"))
```



```
# Defect Presence
ggplot(data_proc_2, aes(x = Defect_Presence, col = Heart_Disease, fill = Heart_Disease))+
  geom_bar()+
  ggtitle("Presence of Heart Defect") +
  xlab("Heart Defect") +
  ylab("Count of Cases")+
theme(legend.position = "top", legend.title = element_blank(),
      plot.title = element_text(size = 14, face = "bold"))
```



There appears to be a considerable degree of correlation between the presence of a heart defect and presence of heart disease, in particular with values 6 and 7 (i.e. which means presence of heart defects).

Application of Machine Learning Techniques

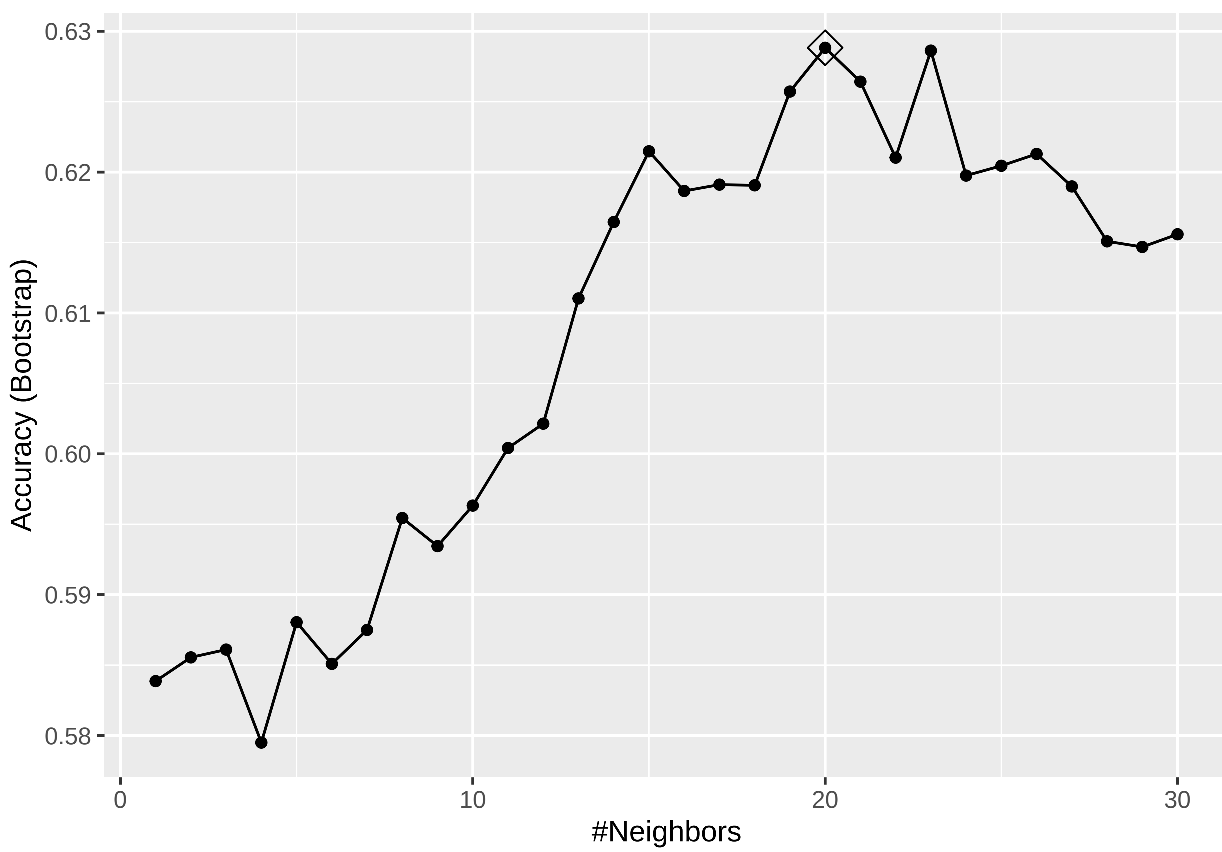
KNN-based Model

The first model evaluated in the project is based on the **KNN Technique**.

According to Wikipedia, *"in k-NN classification, the output is a class membership. An object is classified by a plurality vote of its neighbors, with the object being assigned to the class most common among its k nearest neighbors (k is a positive integer, typically small). If k = 1, then the object is simply assigned to the class of that single nearest neighbor."*

The **hyper-parameter** that must be tune is thus linked to the **number of nearest neighbors (k-value)** to be taken into consideration when making the prediction. The *caret* packages covers this need and we able to identify the optimal k value that should be incorporated in the model.

In the plot here below, we can see that a k-value equals to 20 provides the best value in terms of Accuracy.



When commenting the overall results of the model expressed in the Confusion Matrix, we can see that the model performs quite poorly, with an accuracy rate that is only slightly above the *no-information-rate* of the model.

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction No Yes
##        No  24  13
##        Yes  8  15
##
##                Accuracy : 0.65
##                  95% CI : (0.516, 0.7687)
##     No Information Rate : 0.5333
##     P-Value [Acc > NIR] : 0.04534
##
##                   Kappa : 0.2889
##
##  Mcnemar's Test P-Value : 0.38273
##
##             Sensitivity : 0.5357
##             Specificity : 0.7500
##          Pos Pred Value : 0.6522
##          Neg Pred Value : 0.6486
##              Prevalence : 0.4667
##          Detection Rate : 0.2500
##    Detection Prevalence : 0.3833
##       Balanced Accuracy : 0.6429
##
##        'Positive' Class : Yes
##
```

AdaBoost Classification Trees

The second model we will be looking at is based around the AdaBoost technique.

According to Wikipedia, *AdaBoost, short for Adaptive Boosting, is a machine learning meta-algorithm formulated by Yoav Freund and Robert Schapire, who won the 2003 Gödel Prize for their work. It can be used in conjunction with many other types of learning algorithms to improve performance. The output of the other learning algorithms ('weak learners') is combined into a weighted sum that represents the final output of the boosted classifier. AdaBoost is adaptive in the sense that subsequent weak learners are tweaked in favor of those instances misclassified by previous classifiers. In some problems it can be less susceptible to the overfitting problem than other learning algorithms. The individual learners can be weak, but as long as the performance of each one is slightly better than random guessing, the final model can be proven to converge to a strong learner.*

Furthermore, *AdaBoost (with decision trees as the weak learners) is often referred to as the best out-of-the-box classifier.*

When examining the confusion matrix results, we can see that there is a significant improvemente in terms of accuracy. This brings our current level of accuracy well above the no information rate.

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction No Yes
##        No  28  10
##        Yes  4  18
##
##                Accuracy : 0.7667
##                  95% CI : (0.6396, 0.8662)
##     No Information Rate : 0.5333
##     P-Value [Acc > NIR] : 0.0001655
##
##                   Kappa : 0.5249
##
##  Mcnemar's Test P-Value : 0.1814492
##
##             Sensitivity : 0.6429
##             Specificity : 0.8750
##          Pos Pred Value : 0.8182
##          Neg Pred Value : 0.7368
##              Prevalence : 0.4667
##          Detection Rate : 0.3000
##    Detection Prevalence : 0.3667
##       Balanced Accuracy : 0.7589
##
##        'Positive' Class : Yes
##
```

XGBOOST model

As reported in machinelearningmastery.com, *"This algorithm goes by lots of different names such as gradient boosting, multiple additive regression trees, stochastic gradient boosting or gradient boosting machines. Boosting is an ensemble technique where new models are added to correct the errors made by existing models. Models are added sequentially until no further improvements can be made. A popular example is the AdaBoost algorithm that weights data points that are hard to predict. Gradient boosting is an approach where new models are created that predict the residuals or errors of prior models and then added together to make the final prediction. It is called gradient boosting because it uses a gradient descent algorithm to minimize the loss when adding new models. This approach supports both regression and classification predictive modeling problems."*

In order to tune the model, we must take into consideration several parameters. Here below are some of the parameters taken into consideration.

```
xgb_grid <- expand.grid(nrounds = 1000,
                        max_depth = c(2,5,10),
                        eta = c(0.01),
                        gamma = c(0.5,1.0),
                        colsample_bytree = c(0.5),
                        subsample = c(0.5, 0.6),
                        min_child_weight = seq(1))
```

Furthermore, we will also take into consideration cross-validation in the training dataset. This will be covered by the following code:

```
train_control <- trainControl(method = "cv", number = 5)
# Train xgb model
train_xgb <- train(Heart_Disease ~ .,
                   data = TrainingSet,
                   method ="xgbTree",
                   tuneGrid = xgb_grid,
                   trControl = train_control)
```

Let us now look at the results for the model.

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction No Yes
##        No  29   9
##        Yes  3  19
##
##                Accuracy : 0.8
##                  95% CI : (0.6767, 0.8922)
##     No Information Rate : 0.5333
##     P-Value [Acc > NIR] : 1.609e-05
##
##                   Kappa : 0.5928
##
##  Mcnemar's Test P-Value : 0.1489
##
##             Sensitivity : 0.6786
```

```
##          Specificity : 0.9062
##       Pos Pred Value : 0.8636
##       Neg Pred Value : 0.7632
##           Prevalence : 0.4667
##       Detection Rate : 0.3167
## Detection Prevalence : 0.3667
##    Balanced Accuracy : 0.7924
##
##     'Positive' Class : Yes
##
```

As we can see, the accuracy has once again improved compared to the previous model. However, the model does seem to perform poorly in terms of sensitivity.

Weighted Space Random Forest Model

The Weighted Space Random Forest technique is, according to the creators of the method (Zhao, Williams and Huang, 2017), *A novel variable weighting method is used for variable subspace selection in place of the traditional approach of random variable sampling. This new approach is particularly useful in building models for high dimensional data.*

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction No Yes
##        No  28   9
##        Yes  4  19
##
##                Accuracy : 0.7833
##                  95% CI : (0.658, 0.8793)
##     No Information Rate : 0.5333
##     P-Value [Acc > NIR] : 5.405e-05
##
##                   Kappa : 0.5598
##
##  Mcnemar's Test P-Value : 0.2673
##
##             Sensitivity : 0.6786
##             Specificity : 0.8750
##          Pos Pred Value : 0.8261
##          Neg Pred Value : 0.7568
##              Prevalence : 0.4667
##          Detection Rate : 0.3167
##    Detection Prevalence : 0.3833
##       Balanced Accuracy : 0.7768
##
##        'Positive' Class : Yes
##
```

Compared to the previous model, Accuracy has decreased. This model confirms a poor performance in terms of Sensitivity.

Support Vector Machine Model

Let us now conclude with the last model, based on Support Vector Machines. An SVM can be defined in the following manner, according to Wikipedia: *A support-vector machine constructs a hyperplane or set of hyperplanes in a high- or infinite-dimensional space, which can be used for classification, regression, or other tasks like outliers detection.*

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction No Yes
##        No  29   7
##        Yes  3  21
##
##                Accuracy : 0.8333
##                  95% CI : (0.7148, 0.9171)
##     No Information Rate : 0.5333
##     P-Value [Acc > NIR] : 1.056e-06
##
##                   Kappa : 0.6622
##
##  Mcnemar's Test P-Value : 0.3428
##
##             Sensitivity : 0.7500
##             Specificity : 0.9062
##          Pos Pred Value : 0.8750
##          Neg Pred Value : 0.8056
##              Prevalence : 0.4667
##          Detection Rate : 0.3500
##    Detection Prevalence : 0.4000
##       Balanced Accuracy : 0.8281
##
##        'Positive' Class : Yes
##
```

This appears to be the best model, with a strong performance Accuracy and Specificity, while also improving in terms of sensitivity.

Conclusions

Let us now look at a comparison across all models evaluated.

	Accuracy	Balanced Accuracy	Sensitivity	Specificity
knn_results	0.650	0.643	0.536	0.750
ada_results	0.767	0.759	0.643	0.875
xgb_results	0.800	0.792	0.679	0.906
wsrf_results	0.783	0.777	0.679	0.875
svmR_results	0.833	0.828	0.750	0.906

Given the nature of the task, when comparing our models we will not take into consideration only Accuracy, but we will instead take into consideration an collection of metrics: Accuracy, Balanced Accuracy, Sensitivity and Specificity.

As we can see, we have a clear 'winner' in terms of all key metrics taken into consideration. Considering the real-life consequences of missing a patient actually affected by heart disease, we should pay particular attention to Sensitivity values of the model.