Harvardx - Capstone -Heart Disease predictor

Sajini Arumugam

12/28/2020

Contents

HarvardX PH:125.9 - Capstone : Heart disease Analysis

This analysis is a part of the Harvard edX data science project.

1 Introduction

There are different types of heart diseases at current day and age and most of them are always difficult to predict. Heart disease is a condition that affects the structure or function of the heart.Most people think of heart disease as one condition but there are various group of conditions with different root causes.

Heart disease can be classified into - Coronary artery disease (atherosclerosis) hardening of arteries
- Heart rhythm disorders (arrhythmia), heart beating too quickly or too slowly
- Structural heart disease, abnormalities of heart structure
- Heart failure, occurs when the heart becomes damaged or weakened

Heart disease can be caused by high cholesterol levels, diabetes, high blood pressure and many others, some of which, we are going to explore in this dataset.

This particular data set is from the UCI machine learning repository (https://archive.ics.uci.edu/ml/datasets/heart+disease) which is located in the kaggle public data sets (https://www.kaggle.com/ronitf/heart-disease-uci). It contains 76 attributes but we are going to be using a subset of 14 of them, mainly the Cleveland database section of it.

Report is going to include exploratory data analysis then the machine learning models to predict the accuracy of findings.

2 Exploratory Data Analysis

2.1 Dataset overview

After downloading the dataset, the first step is to look at the general overview of the data. Ours is a very small dataset and it's particularly tidy. We had to define column names for better understanding. Each row contains patient information with different test they underwent and the results of it.

- age
- sex (1 = Male, 0 = Female)
- chest_pain (0 - Asymptomatic, 1 - Atypical angina, 2 - Non-typical Angina, 3 - Typical angina)
- rest_bp- blood pressure at rest
- cholesterol
- fast_blood_sugar
- rest_ecg - (0 - Normal, 1 - Wave abnormality, 2 - Hypertrophy)
- maxHR - Maximum heart rate during stress
- exercise_angina - angina from exercise
- oldpeak - oldpeakST depression induced by exercise relative to rest
- slope - slope of the peak ST (2 - Ascending, 1 - flat, 3 - descending)
- ca - number of major blood vessels coloured by fluoroscopy
- thal - thallium stress (1 - fixed defect, 2 - nirmal, 3 - reversible defect)
- disease_indicator - (1 - No, 0 - Yes)

Data is downloaded from kaggle and stored in the public google drive repository and auto downloaded in here. Column names are changed for easier understanding.

```r
# Loading data from public google drive
heart_data <-read.csv(sprintf("https://drive.google.com/uc?id=1voGOsP6Hg3q4Xw3MBm9d_632dGvxVzr


# changing column names for easier understanding

colnames(heart_data) <- c("age", "sex", "chest_pain",
                          "rest_BP", "cholesterol",
                          "fast_blood_sugar", "rest_ecg",
                          "maxHR", "exercise_angina",
                          "oldpeak", "slope", "ca",
                          "thal", "disease_indicator")
```

Check for missing values or NAs. Dataset does'nt have any NAs.

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

We are now changing values of some variables for easier understanding. It includes setting sex to Male or Female instead of 0/1. Blood sugar to >120 and <=120, chest pain to Atypical, non-typical, typical and asymptomatic and so on.

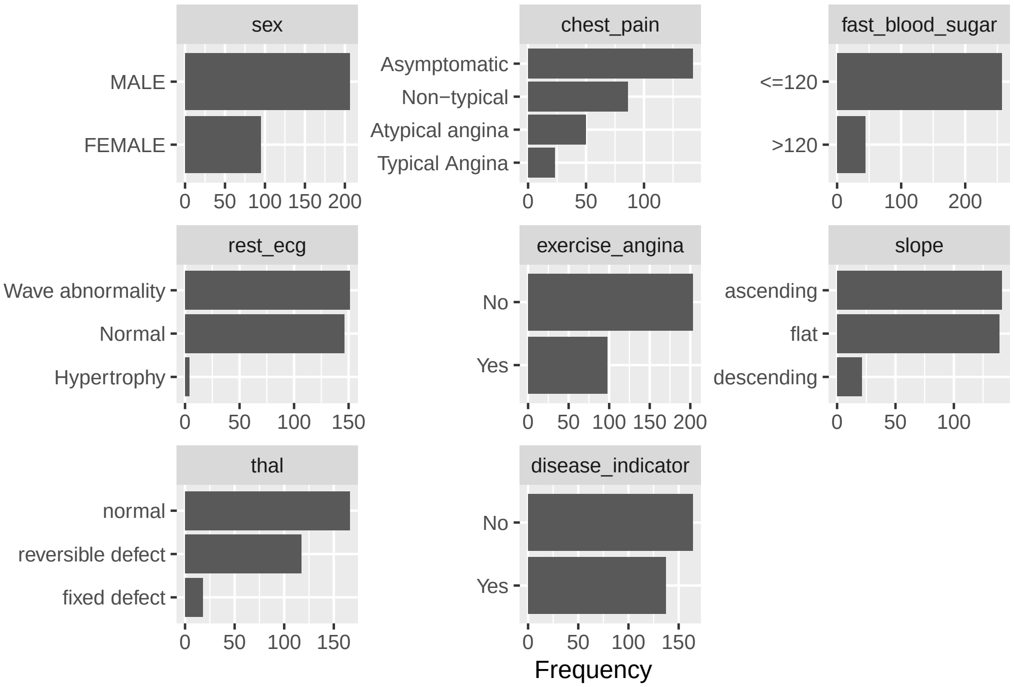
```r
#renaming values
heart_data <- heart_data %>%
  filter(thal != 0) %>%
  mutate(sex = if_else(sex == 1, "MALE", "FEMALE"),
         fast_blood_sugar = if_else(fast_blood_sugar == 1,
                                    ">120",  "<=120"),
         exercise_angina = if_else(exercise_angina == 1,
                                   "Yes" ,"No"),
         chest_pain = if_else(chest_pain == 1, "Atypical angina",
                              if_else(chest_pain == 2,
                                      "Non-typical",
                                      if_else(chest_pain == 0,
                                              "Asymptomatic",
                                              "Typical Angina"))),
         rest_ecg = if_else(rest_ecg == 0, "Normal",
                            if_else(rest_ecg == 1,
                                    "Wave abnormality",
                                    "Hypertrophy")),

         ca = as.numeric(ca),
         disease_indicator = if_else(disease_indicator == 1, "No", "Yes")
  ) %>%
  mutate_if(is.character, as.factor) #convert to factor
```

```r
#renaming values
heart_data <- heart_data %>%
  mutate(slope = if_else(slope == 2, "ascending",
                         if_else(slope == 1, "flat","descending")),

         thal = case_when(
           thal == 1 ~ "fixed defect",
           thal == 2 ~ "normal",
           thal == 3 ~ "reversible defect"
         )
         )%>%
           mutate_if(is.character, as.factor)
```

Table 1: Data

age	sex	chest_pain	rest_BP	cholesterol	fast_blood_sugar	rest_ecg	maxHR	exercise_angina	oldpeak	slope	ca	thal	disease_indicator
63	MALE	Typical Angina	145	233	>120	Normal	150	No	2.3	descending	0	fixed defect	No
37	MALE	Non-typical	130	250	<=120	Wave abnormality	187	No	3.5	descending	0	normal	No
41	FEMALE	Atypical angina	130	204	<=120	Normal	172	No	1.4	ascending	0	normal	No
56	MALE	Atypical angina	120	236	<=120	Wave abnormality	178	No	0.8	ascending	0	normal	No
57	FEMALE	Asymptomatic	120	354	<=120	Wave abnormality	163	Yes	0.6	ascending	0	normal	No
57	MALE	Asymptomatic	140	192	<=120	Wave abnormality	148	No	0.4	flat	0	fixed defect	No

```
## Rows: 301
## Columns: 14
## $ age               <int> 63, 37, 41, 56, 57, 57, 56, 44, 52, 57, 54, 48, 4...
## $ sex               <fct> MALE, MALE, FEMALE, MALE, FEMALE, MALE, FEMALE, M...
## $ chest_pain        <fct> Typical Angina, Non-typical, Atypical angina, Aty...
## $ rest_BP           <int> 145, 130, 130, 120, 120, 140, 140, 120, 172, 150,...
## $ cholesterol       <int> 233, 250, 204, 236, 354, 192, 294, 263, 199, 168,...
## $ fast_blood_sugar  <fct> >120, <=120, <=120, <=120, <=120, <=120, <=120, <...
## $ rest_ecg          <fct> Normal, Wave abnormality, Normal, Wave abnormalit...
## $ maxHR             <int> 150, 187, 172, 178, 163, 148, 153, 173, 162, 174,...
## $ exercise_angina   <fct> No, No, No, No, Yes, No, No, No, No, No, No, No, ...
## $ oldpeak           <dbl> 2.3, 3.5, 1.4, 0.8, 0.6, 0.4, 1.3, 0.0, 0.5, 1.6,...
## $ slope             <fct> descending, descending, ascending, ascending, asc...
## $ ca                <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ thal              <fct> fixed defect, normal, normal, normal, normal, fix...
## $ disease_indicator <fct> No, No, No, No, No, No, No, No, No, No, No, No, N...
```



2.2 Data Exploration

Let's look into the variables and see how most of them affect the outcome (disease_indicator).

We'll make a common density plot function and display the grid of plots and go into further detail.

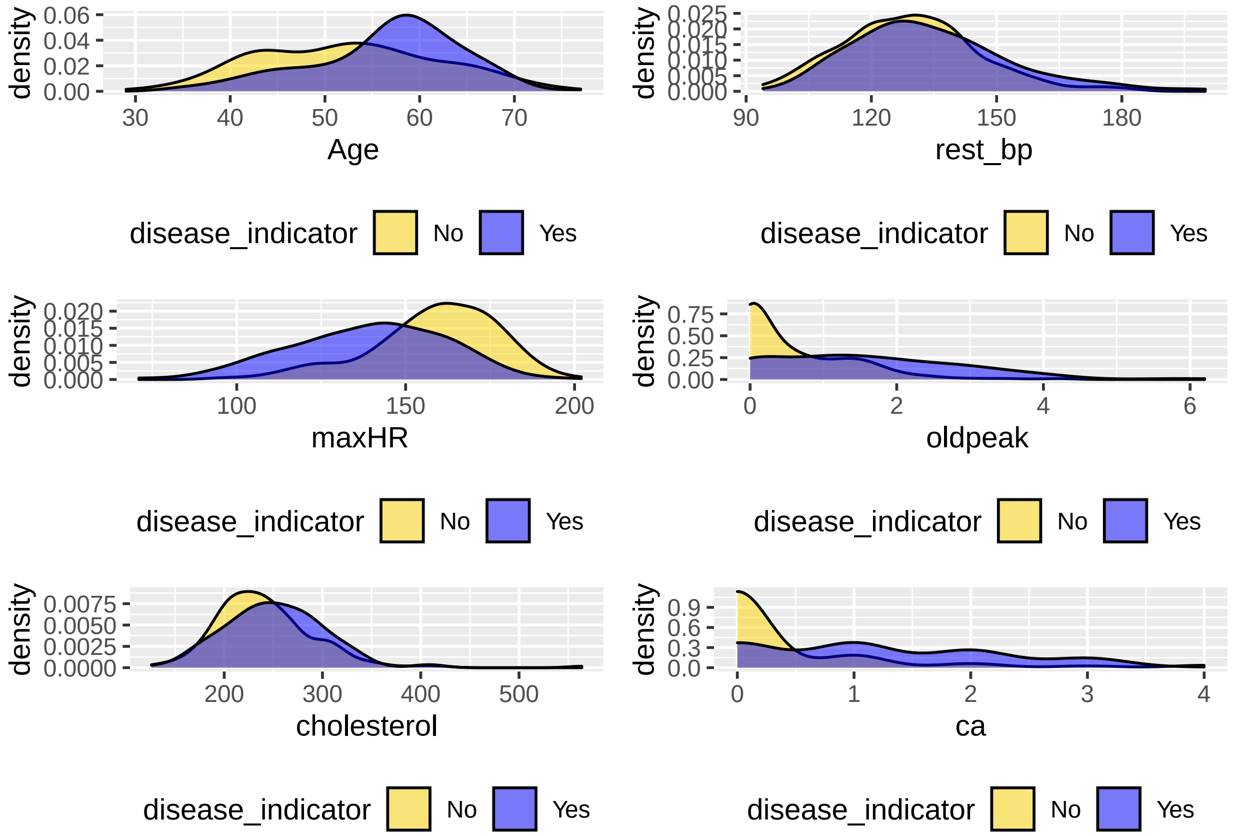


Figure 1: Density plot

2.2.1 Age

There is a over lap with the disease indicator on the density plot. Here when we look at the age barplot we can see that age doesn't seem to have much impact on deciding the heart disease. People around the age of 55 and 65 have the highest percentage of heart disease, but also the highest percentage of no disease as well. So age is not a clear indicator of the disease's impact.

2.2.2 Sex

```
## $x
## [1] "sex"
##
## $y
## [1] "patients"
```

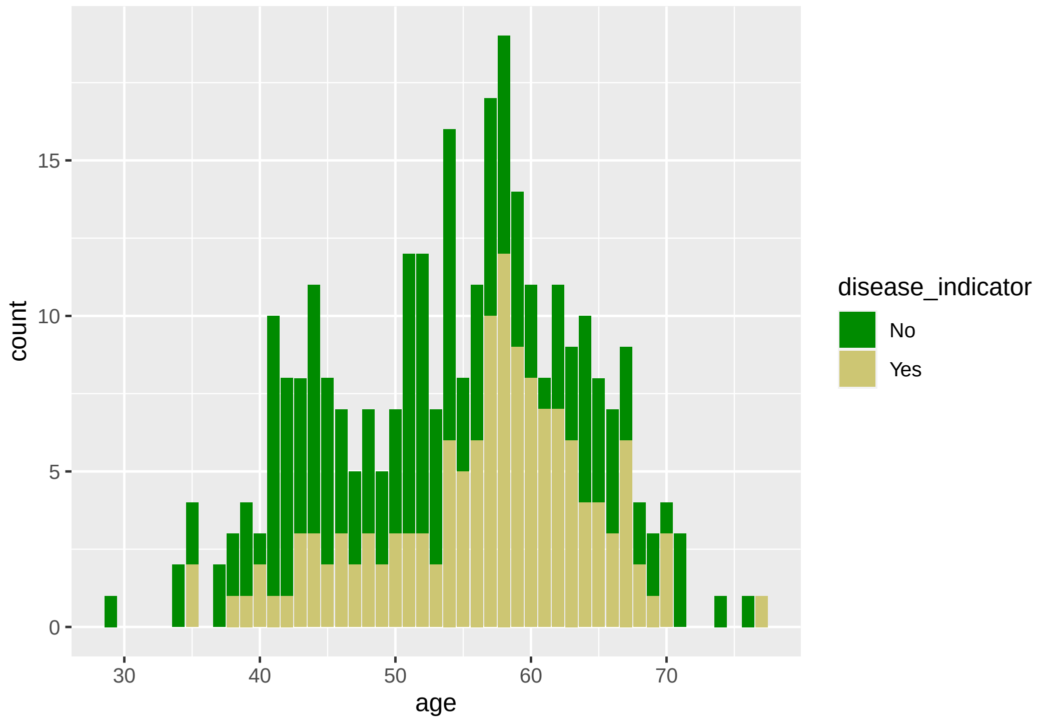


Figure 2: Age Barplot

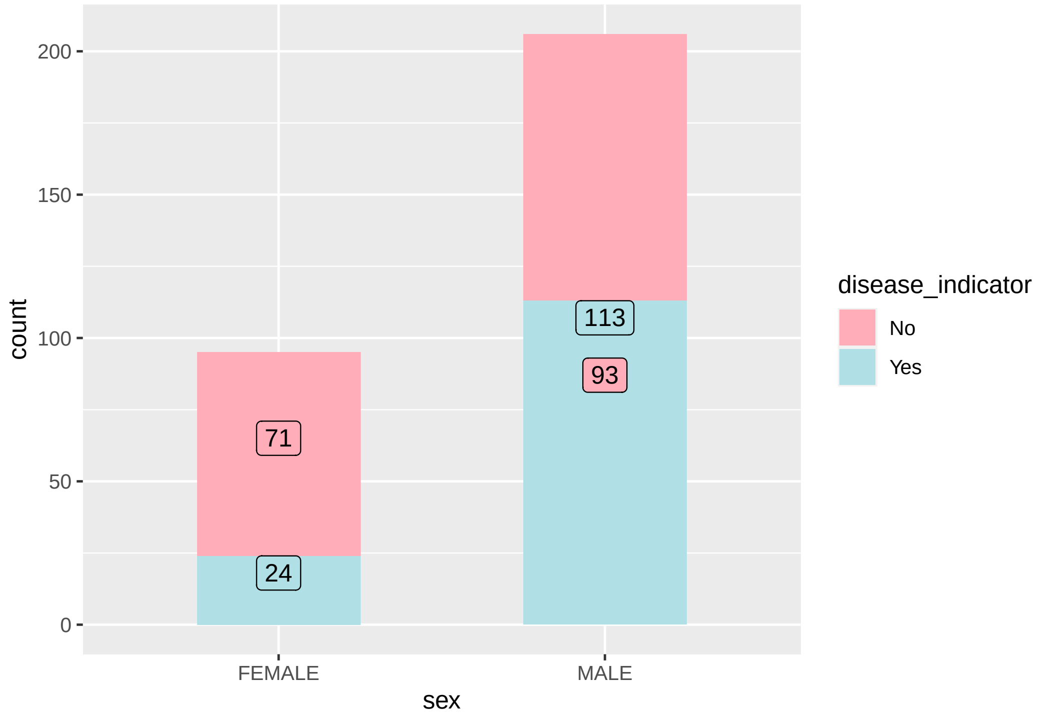


Figure 3: Sex barplot

```
##
## attr(,"class")
## [1] "labels"
```

The disease indicator is almost 50 - 50 in the male and with regards to female it's almost 75-25. Only 25% of female have the disease while 50% of male are affected.

2.2.3 Chest pain

As with chest pain, majority of it is asymptomatic.Meaning most people did not have any symptoms or only had minor symptoms of heart attack, but it's like any other symptoms where blood flow to a section of the heart is temporarily blocked. The second most common is non-typical pain, meaning the pain resembles chest pain unlike a definite chest pain. The other two are comparatively smaller proportions.

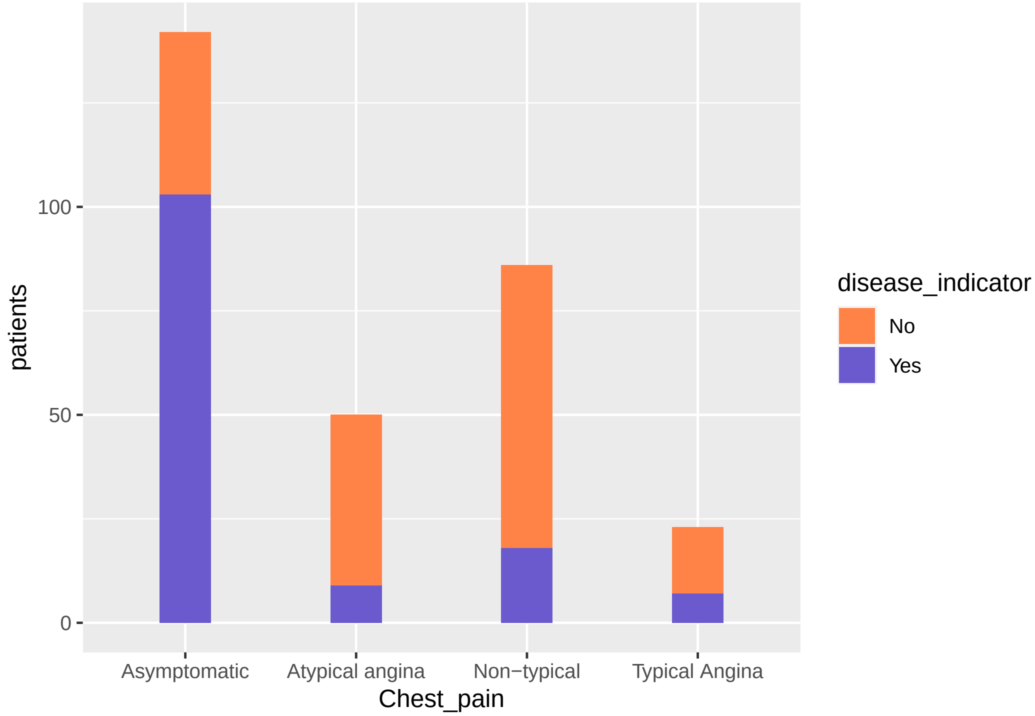


Figure 4: Chest pain barplot

2.2.4 Cholesterol with age

Now let's look into cholesterol. it's almost equally distributed in male and female with some female proportions showing high numbers but they don't seem to be the cause of disease. But there is a large proportion of cholesterol induced heart disease in men.

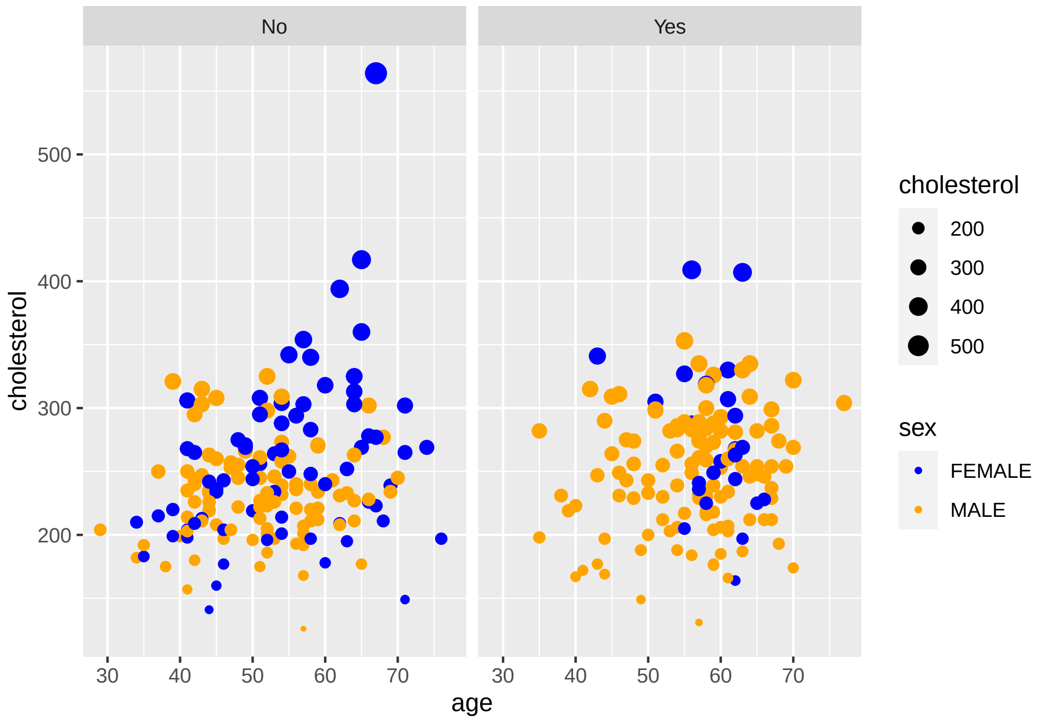


Figure 5: Cholesterol with Age

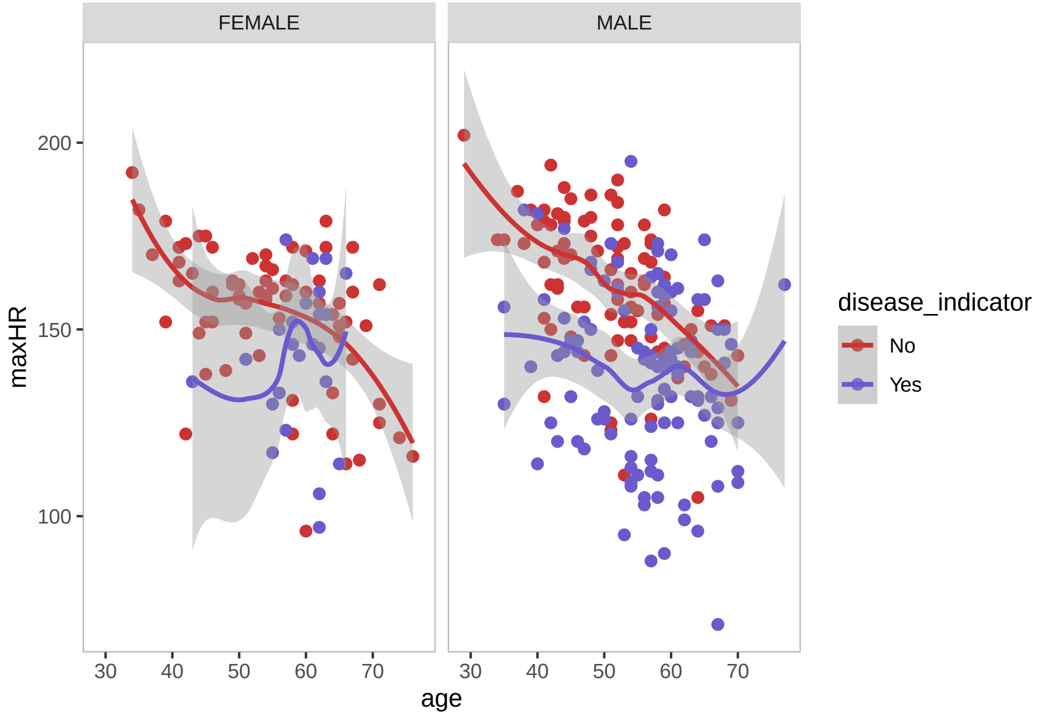


Figure 6: Age vs maxHR

2.2.5 Age vs maxHR

Firstly, maximum heart rate is varied among the age criteria, with some at 30s' having max heart rate and some at 70 having lower. Both in male and female, people between 130 and 150 HR seem to have been the most affected whereas some with even 200 maxHR don't seem to have the disease.

2.2.6 Blood pressure vs chest pain

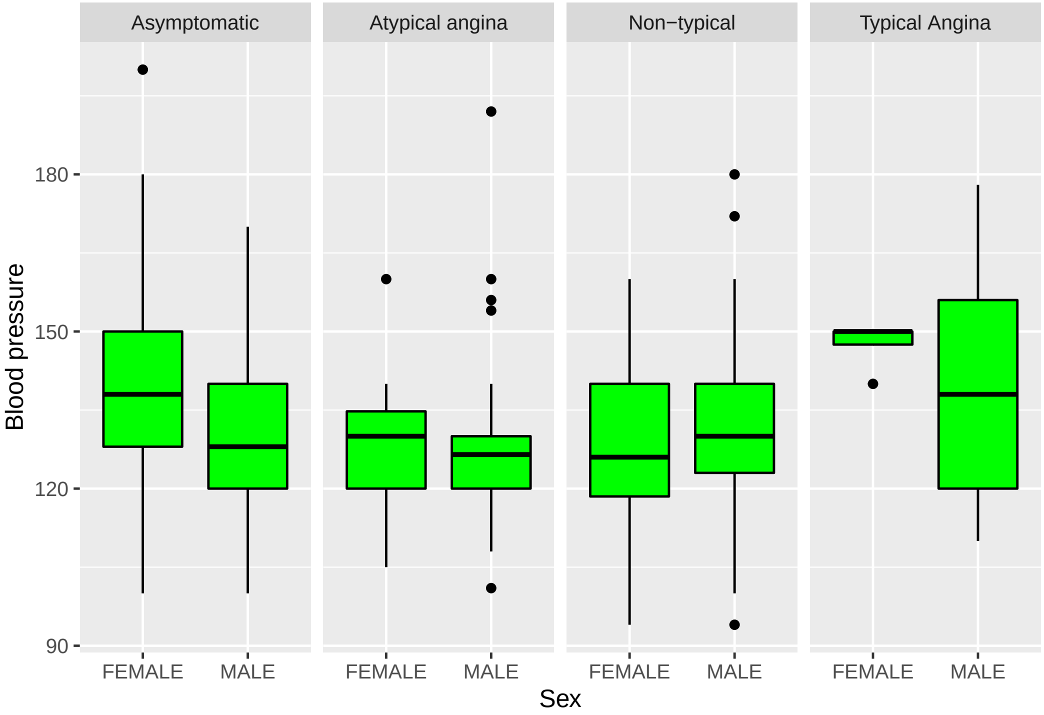
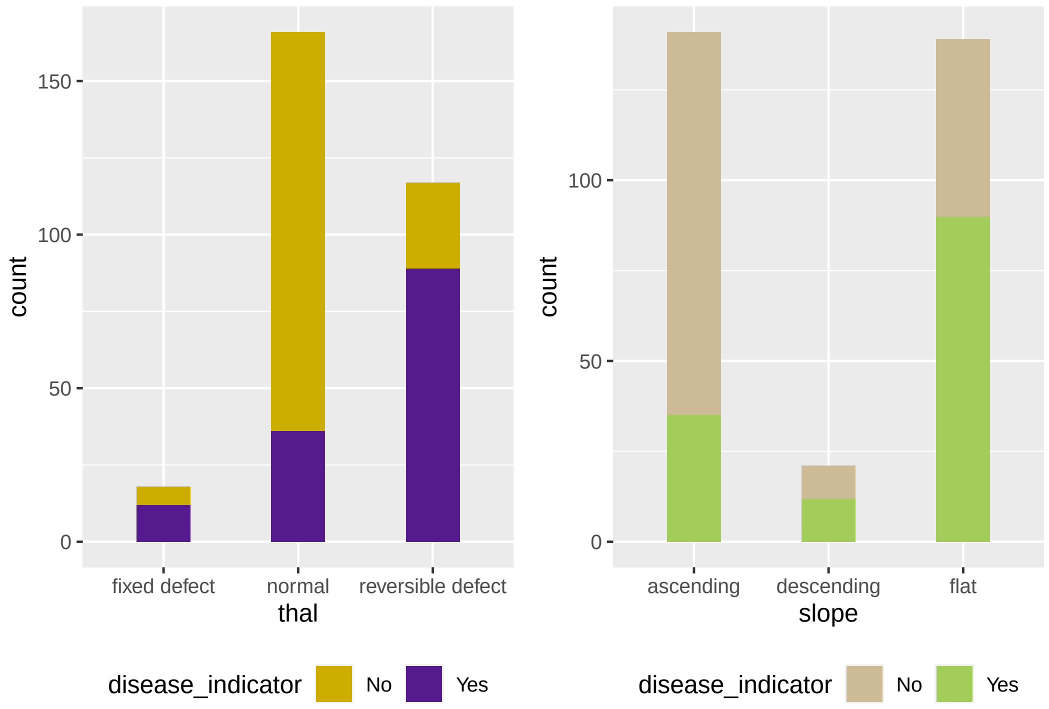


Figure 7: Blood pressure vs Chest pain

This shows the blood pressure distribution amongst the chest pain category. Blood pressure seems to vary with regards to different kinds of chest pain and female category seems to have comparatively higher blood pressure.

2.2.7 Slope and Thallium

Let's look at slope. Most people with the disease seem to have a flat slope compared to an ascending slope. As with thallium stress results, most with reversible defect show signs of disease.



2.2.8 Correlation plot & PCA

Table below shows how different variables are correlated and the principal component analysis for some of the variables.

```
##                age rest_BP cholesterol  maxHR oldpeak     ca
## age          1.000   0.279       0.213 -0.401   0.210  0.276
## rest_BP      0.279   1.000       0.122 -0.048   0.193  0.101
## cholesterol  0.213   0.122       1.000 -0.012   0.052  0.067
## maxHR       -0.401  -0.048      -0.012  1.000  -0.350 -0.217
## oldpeak      0.210   0.193       0.052 -0.350   1.000  0.221
## ca           0.276   0.101       0.067 -0.217   0.221  1.000
```

```
## Importance of components:
##                            PC1     PC2      PC3     PC4     PC5     PC6
## Standard deviation     52.0067 23.2756 17.51948 7.66468 1.10593 0.93430
## Proportion of Variance  0.7483  0.1499  0.08492 0.01625 0.00034 0.00024
## Cumulative Proportion   0.7483  0.8982  0.98317 0.99942 0.99976 1.00000
```

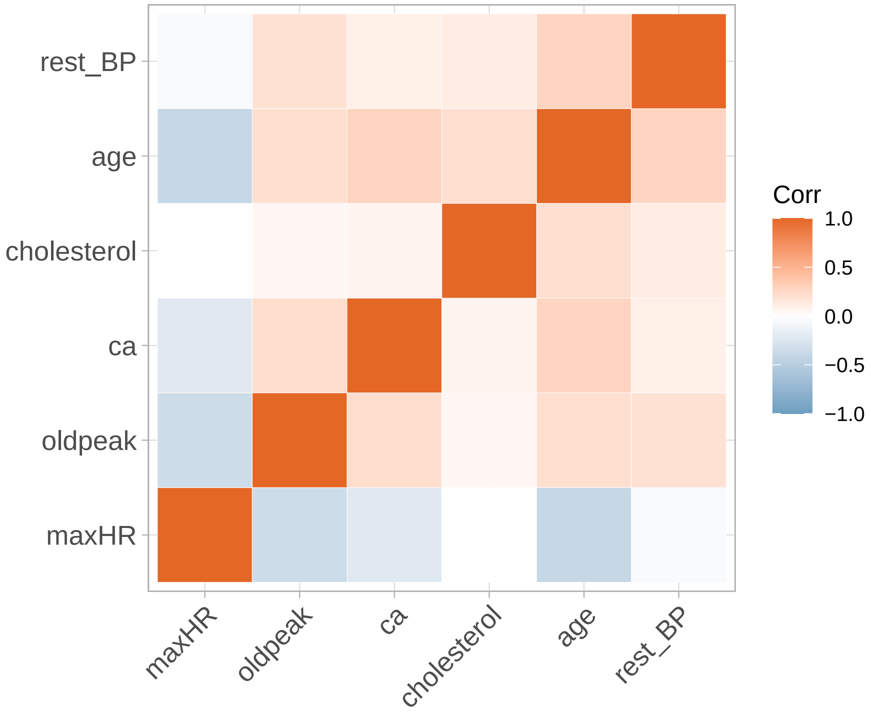


Figure 8: Correlation plot

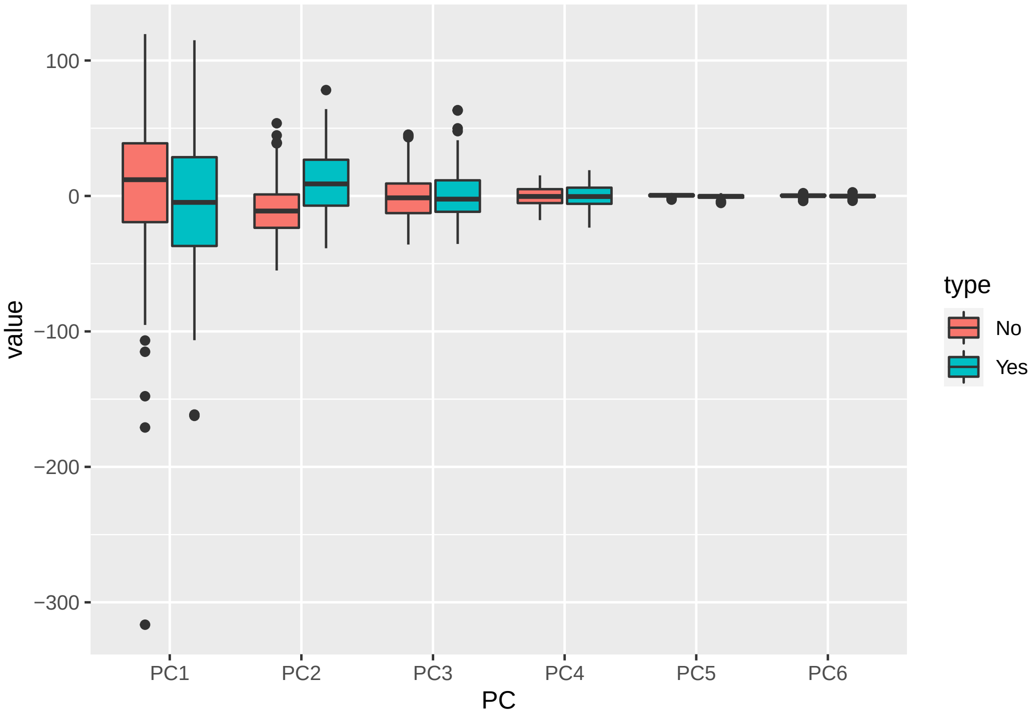
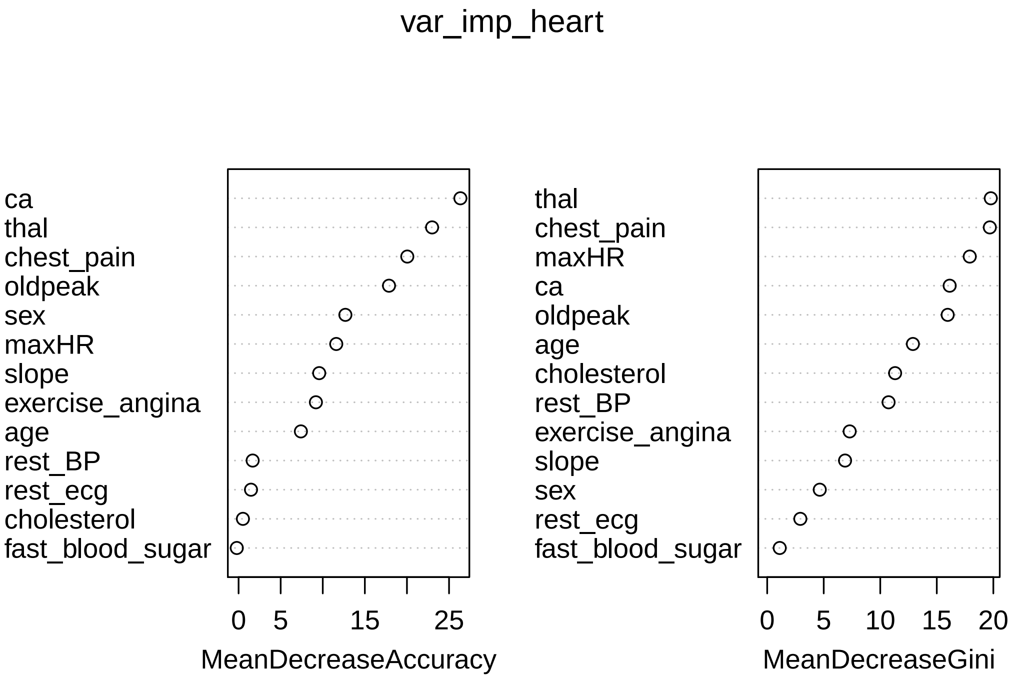


Figure 9: PCA Boxplot

3 Models

3.1 Data pre-processing

Now that we have analyzed the data, we can filter it and prepare for building the model. Let's look at a variable importance plot to see how the columns are ordered.



As we can see from the plot *fast_blood_sugar*, *cholestrol*, *rest_ecg* and *rest_BP* don't contribute much to our model so we will exclude it from our dataset while predicting accuracy.

3.2 Splitting test and train sets

Let's split our data into testing and training sets to do our training. Here we are doing a 70 - 30 split as it proved to give the most accuracy among other option. Although 80-20 and 90-10 were close to giving acceptable accuracies, 70-30 is chosen to build a better predicting algorithm.

Along with that we will also exclude the 4 columns with least importance.

```
set.seed(1, sample.kind = "Rounding")
train_index <- createDataPartition(y = heart_data$disease_indicator,
                                   times = 1, p = 0.7, list = FALSE)
train_heart <- heart_data[train_index,]
test_heart <- heart_data[-train_index,]


# sub-setting to exclude 4 columns
train_heart<-subset(train_heart,select = -c(cholesterol,fast_blood_sugar,rest_BP,rest_ecg))
test_heart<-subset(test_heart,select = -c(cholesterol,fast_blood_sugar,rest_BP,rest_ecg))
```

3.3 Models

3.3.1 Logistic Regression

It's the most commonly used form of generalized linear model. It assumes that the predictor X and the outcome Y follow a bivariate normal distribution.

```
set.seed(127, sample.kind = "Rounding")
train_glm <- train(disease_indicator ~.,
                  data = train_heart,
                  method = "glm",
                  trControl = fitControl, tuneGrid = NULL)
glm_predict <- predict(train_glm, test_heart)
#mean(glm_predict == test_heart$disease_indicator)

#confusion matrix of logistic regression
glm_c <- confusionMatrix(glm_predict,test_heart$disease_indicator, positive = "Yes")
```

Method	Accuracy
Logistic Regression	0.8666667

Accuracy of 0.8666667 which is a great start. Let's look into other models.

3.3.2 K-Nearest Neighbour

Starting Knn, let's first optimize for *k*. We will have to compute distance between each observation in the test set and each observation in the training set, we will use k-fold cross validation to improve speed.

Control was already set at the beginning with 10 fold cross validation.

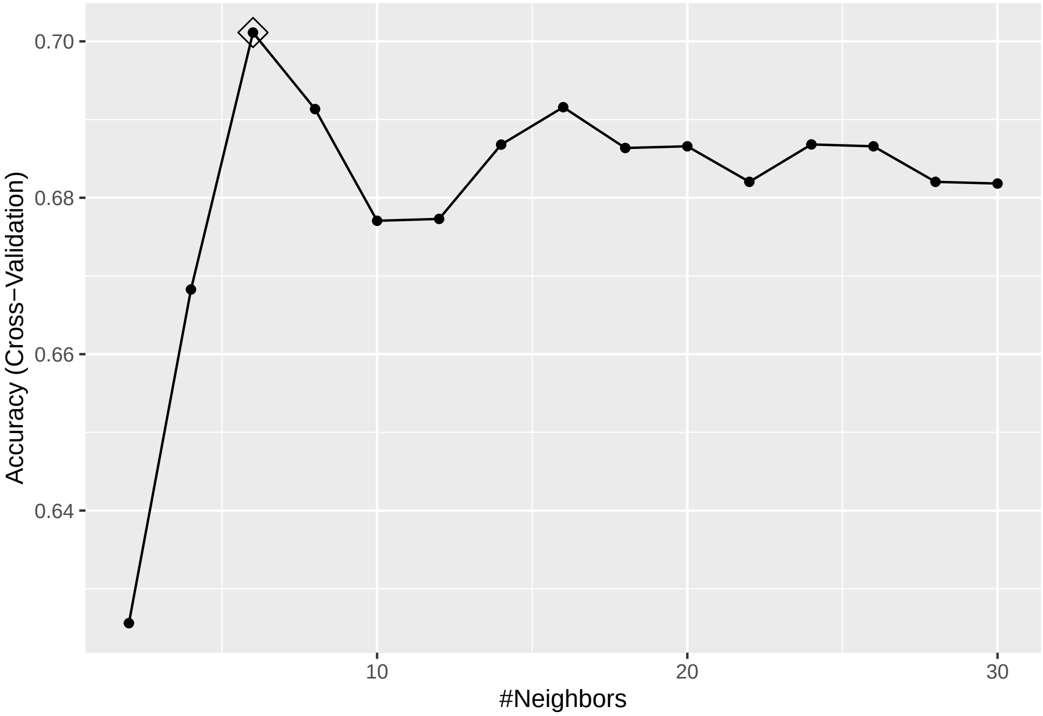
```
set.seed(2020, sample.kind = "Rounding")

train_kn <- train(disease_indicator ~. ,
                  data = train_heart,
                  method = "knn",
                  trControl = fitControl, #10 fold cv
                  tuneGrid = data.frame(k = seq(2, 30, 2)))

train_kn$bestTune #best tune
```

	k
3	6

```
kn_predict <- predict(train_kn, test_heart)
mean(kn_predict == test_heart$disease_indicator)
```

```
## [1] 0.7111111
```

```
plot_kn <- ggplot(train_kn,highlight = TRUE)
plot_kn
```



Method	Accuracy
Logistic Regression	0.8666667
KNN	0.7111111

accuracy of 0.7111111 which is less than the accuracy obtained from logistic regression.

3.3.3 Regression tree

Basic regression trees partition a data set into smaller groups and then fit a simple model for each subgroup. Most times a singles tree tends to be highly unstable and a poor predictor. However by bootstrapping regression trees, this technique can be quite powerful and effective.

Let's build a rpart method to predict accuracy

```
set.seed(13, sample.kind = "Rounding")
train_rpart <- train(disease_indicator ~.,
                     data = train_heart,
                     method = "rpart")

rpart_predict <- predict(train_rpart, test_heart)
mean(rpart_predict == test_heart$disease_indicator)
```

```
## [1] 0.7777778
```

Method	Accuracy
Logistic Regression	0.8666667
KNN	0.7111111
Regression Trees	0.7777778

accuracy of 0.7777778, slighlty better than knn.

3.3.4 Random forest model

The random forest algorithm works by aggregating the predictions made by multiple decision trees of varying depth. Every decision tree in the forest is trained on a subset of the dataset called the bootstrapped dataset.

When the random forest is used for classification and is presented with a new sample, the final prediction is made by taking the majority of the predictions made by each individual decision tree in the forest. In the event, it is used for regression and it is presented with a new sample, the final prediction is made by taking the average of the predictions made by each individual decision tree in the forest.

With random forest, computation time is a challenge. For each forest, we need to build hundreds of trees. We also have several parameters we can tune.

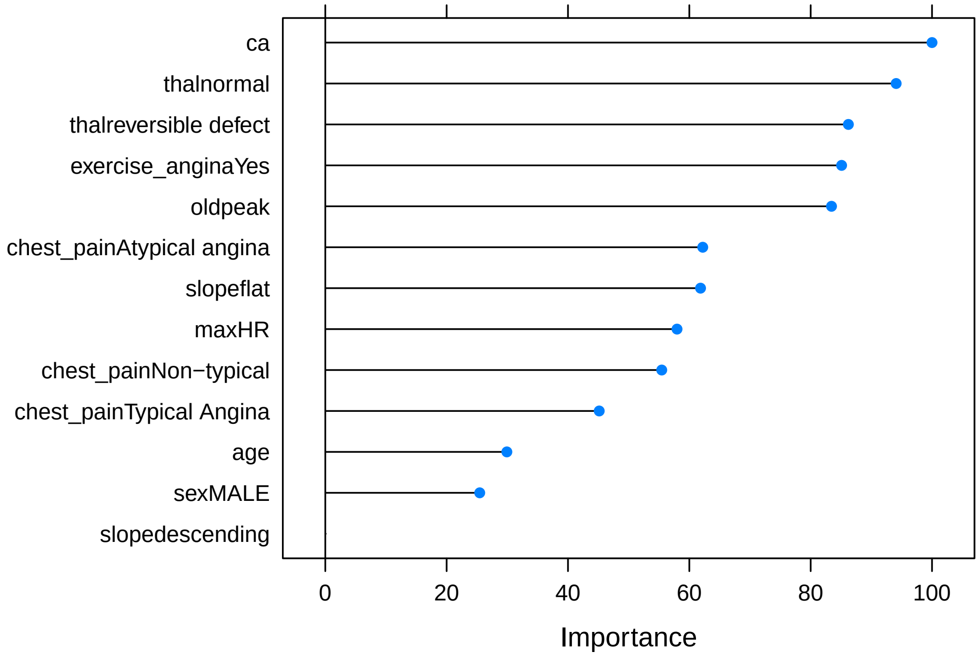
```
set.seed(13, sample.kind = "Rounding")
tuning <- data.frame(mtry = c(1,20,1))

train_rf <- train(disease_indicator ~.,
```

```r
                     data = train_heart,
                     method = "rf",
                  tuneGrid = tuning,
                  importance = TRUE)
train_rf$bestTune
```

mtry
1

```r
rf_predict <- predict(train_rf, test_heart)
mean(rf_predict == test_heart$disease_indicator)
```

```
## [1] 0.8666667
```



Method	Accuracy
Logistic Regression	0.8666667
KNN	0.7111111
Regression Trees	0.7777778
Random Forest	0.8666667

Accuracy of 0.8666667 which is similar to that of logistic regression. Let’s move on to our next model.

3.3.5 Adaptive boosting

AdaBoost helps you combine multiple weak classifiers into a single strong classifier. It is called Adaptive Boosting as the weights are re-assigned to each instance, with higher weights to incorrectly classified instances. This may take some time to run.

```
#adaptive boosting
set.seed(13, sample.kind = "Rounding")

train_ada <- train(disease_indicator ~.,
                   data = train_heart,
                   method = "adaboost")
ada_predict <- predict(train_ada, test_heart)
mean(ada_predict == test_heart$disease_indicator)
```

```
## [1] 0.8222222
```

Method	Accuracy
Logistic Regression	0.8666667
KNN	0.7111111
Regression Trees	0.7777778
Random Forest	0.8666667
Ada Boost	0.8222222

Accuracy of 0.8222222 which isn't so bad.

3.3.6 Quadratic Discriminant Analysis

```
train_qda <- train(disease_indicator ~.,
                     data = train_heart,
                     method = "qda")
qda_predict <- predict(train_ada, test_heart)
mean(qda_predict == test_heart$disease_indicator)
```

```
## [1] 0.8222222
```

Method	Accuracy
Logistic Regression	0.8666667
KNN	0.7111111
Regression Trees	0.7777778
Random Forest	0.8666667
Ada Boost	0.8222222
QDA	0.8222222

Accuracy of 0.8222222.

4 Results

Let's consolidate the results of confusion matrix in a table and go in to the details about them.

Sensitivity - Sensitivity is defined as the proportion of positive results out of the number of samples which were actually positive. When there are no positive results, sensitivity is not defined and a value of NA is returned. In our case, Sensitivity is to correctly identify those with the disease (true positive rate)

Specificity - Similarly, when there are no negative results, specificity is not defined and a value of NA is returned. it's a test to correctly identify those without the disease (true negative rate)

Positive pred value - Positive predictive value is the probability that subjects with a positive screening test truly have the disease.

negative pred value - Negative predictive value is the probability that subjects with a negative screening test truly don't have the disease.

```
##          Event No Event
## Event    "A"   "B"
## No Event "C"   "D"
```

$$Sensitivity = A/(A+C)$$

$$Specificity = D/(B+D)$$

$$Prevalence = (A+C)/(A+B+C+D)$$

$$PPV = (sensitivity*Prevalence)/((sensitivity*Prevalence)+((1-specificity)*(1-Prevalence)))$$

$$NPV = (specificity*(1-Prevalence))/(((1-sensitivity)*Prevalence)+((specificity)*(1-Prevalence)))$$

From the results table we can conclude that the logistic regression and Random forest model yielded the maximum accuracy among other models followed by Adaptive boosting and QDA. The confusion matrix results show the true positive and true negative rates of all models.

A patient correctly identified as having heart disease will be True Positive and a patient correctly identified as not having disease will be the True Negative value.

Sensitivity ranges between 0.85 and 0.63 which is not that great for predictions. Same goes to the specifictiy, improving these will help increasing our accuracy.

Table 2: Matrix results

	Logistic_Regression	Knn	Regression_Trees	Random_Forest	Ada_boost	QDA
Sensitivity	0.8536585	0.6341463	0.7073171	0.8292683	0.8048780	0.8048780
Specificity	0.8775510	0.7755102	0.8367347	0.8979592	0.8367347	0.8367347
Pos Pred Value	0.8536585	0.7027027	0.7837838	0.8717949	0.8048780	0.8048780
Neg Pred Value	0.8775510	0.7169811	0.7735849	0.8627451	0.8367347	0.8367347
Precision	0.8536585	0.7027027	0.7837838	0.8717949	0.8048780	0.8048780
Recall	0.8536585	0.6341463	0.7073171	0.8292683	0.8048780	0.8048780
F1	0.8536585	0.6666667	0.7435897	0.8500000	0.8048780	0.8048780
Prevalence	0.4555556	0.4555556	0.4555556	0.4555556	0.4555556	0.4555556
Detection Rate	0.3888889	0.2888889	0.3222222	0.3777778	0.3666667	0.3666667
Detection Prevalence	0.4555556	0.4111111	0.4111111	0.4333333	0.4555556	0.4555556
Balanced Accuracy	0.8656048	0.7048283	0.7720259	0.8636137	0.8208064	0.8208064

5 Conclusion

The objective of this project is to use the Cleveland heart disease data set to correctly diagnose people with heart diseases. An explanatory data analysis was done and it revealed how different variables in the dataset help us predict the disease. It also revealed how some factors don't directly influence the results and those factors were later removed to improve our model.

Different machine learning models were built to optimize the accuracy of the prediction and the ones that proved most successful were Logistic Regression model and the Random forest model. The least successful one is the KNN model. Although our accuracy was on an acceptable level, our sensitivity and specificity were still below 90% which is concerning. But with the given set of data this is an efficient outcome.

Many other models were trained but they dint quite improve on the accuracy and hence weren't included in the report. Having more volume of data will enable an improvement in the model with much higher sample set. Also, using feature selection might also improve in a much accurate model.

6 References

https://www.heartandstroke.ca/heart-disease/what-is-heart-disease/types-of-heart-disease

https://archive.ics.uci.edu/ml/datasets/heart+disease

https://uc-r.github.io/regression_trees

https://towardsdatascience.com/random-forest-in-r-f66adf80ec9 http://finzi.psych.upenn.edu/R/library/caret/html/sensitivity.html

https://rafalab.github.io/dsbook/machine-learning-in-practice.html