Heart Disease

Pablo Parra

20/02/2022

Contents

1 Introduction

This is the last project of the Professional Certificate Program of Data Science, organized by Harvard University in the platform edX. There are not restrictions about the dataset selected and the methods applied.

The dataset selected contains information about 918 people from five different datasets, grouped by *fedesoriano*[1]. The original creators of each data information are M.D. Andras Janosi (from Hungarian Institute of Cardiology, Budapest), M.D. William Steinbrunn (from University Hospital, Zurich), M.D. Matthias Pfisterer (from University Hospital, Basel) and M.D. and Ph.D.Robert Detrano (from V.A. Medical Center, Long Beach and Cleveland Clinic Foundation).

This dataset contains different information about the patients as age, sex, cholesterol levels, blood pressure or chest pain type (all the features will be described in section 2.1). But the most important feature is *HeartDisease*, which indicates if that person has a heart disease or not. The distribution of each attribute of the data will be showed and a data cleaning analysis will be performed in order to guarantee that data has the shape needed.

The main goal of the project is to find the machine learning method that achieve the best accuracy possible. The best way to train the different models is splitting the data into two groups: train data and test data. The accuracy percentage will be computed comparing the real test data with the predicted test data.

The machine learning methods utilized in the project are GLM, KNN, LDA, rpart, rborist, GBM and an ensemble with the three best accuracy methods. And then, the most accuracy method will be selected.

[1] fedesoriano. (September 2021). Heart Failure Prediction Dataset. Retrieved [Date Retrieved] from https://www.kaggle.com/fedesoriano/heart-failure-prediction.

2 Methods / Analysis

The colors chosen for this report are cadet blue and steel blue (and its variations), instead of the default colors. Based on the number of different values, these particular palettes of colors will be used.

```
# Selected colors
coloursOfTheProject1 <- c("steelblue3")
coloursOfTheProject2 <- c("cadetblue3", "steelblue3")
coloursOfTheProject3 <- c("cadetblue3", "steelblue3", "steelblue")
coloursOfTheProject4 <- c("cadetblue3", "steelblue3", "steelblue", "steelblue4")
```

2.1 Data structure

Description for the 12 columns/features contained in the data is already provided:

- **Age**: Age of the patient.
- **Sex**: Sex of the patient (M: Male, F: Female).
- **ChestPainType**: Type of chest pain (TA: Typical Angina, ATA: Atypical Angina, NAP: Non-Anginal Pain, ASY: Asymptomatic).
- **RestingBP**: Resting blood pressure (mm Hg).
- **Cholesterol**: Serum cholesterol (mm/dl).
- **FastingBS**: Fasting blood sugar (1: if FastingBS > 120 mg/dl, 0: otherwise).
- **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).
- **MaxHR**: Maximum heart rate achieved (Numeric value between 60 and 202).
- **ExerciseAngina**: Exercise-induced angina (Y: Yes, N: No).
- **Oldpeak**: oldpeak = ST (Numeric value measured in depression).
- **ST_Slope**: the slope of the peak exercise ST segment (Up: upsloping, Flat: flat, Down: downsloping).
- **HeartDisease**: Output feature (1: heart disease, 0: Normal).

The first step is to load the data. It could be downloaded **here**. In the next chunk of code, there will be presented three different ways to download/use the data. By default the code will only read the data from the online repository, but if the user prefers to download the data, there are the other two ways.

```
# Loading data - Different options
# If the user only need to read the data online;
data <- read_csv("https://raw.githubusercontent.com/p-parra/Heart-Project/main/heart.csv")

# If the user prefer to download it in your PC:
# url <- "https://raw.githubusercontent.com/p-parra/Heart-Project/main/heart.csv"
# download.file(url, destfile = "heart.csv")
# data <- read_csv("heart.csv")

# If data it is already downloaded (in the active directory):
# data <- read_csv("heart.csv")
```

To achieve a general understanding of the information of all the features of the dataset, the structure of the data and the first 6 rows of the data will be showed:

```r
# Data structure
str(data)
```

```
## spec_tbl_df [918 x 12] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ Age           : num [1:918] 40 49 37 48 54 39 45 54 37 48 ...
##  $ Sex           : chr [1:918] "M" "F" "M" "F" ...
##  $ ChestPainType : chr [1:918] "ATA" "NAP" "ATA" "ASY" ...
##  $ RestingBP     : num [1:918] 140 160 130 138 150 120 130 110 140 120 ...
##  $ Cholesterol   : num [1:918] 289 180 283 214 195 339 237 208 207 284 ...
##  $ FastingBS     : num [1:918] 0 0 0 0 0 0 0 0 0 0 ...
##  $ RestingECG    : chr [1:918] "Normal" "Normal" "ST" "Normal" ...
##  $ MaxHR         : num [1:918] 172 156 98 108 122 170 170 142 130 120 ...
##  $ ExerciseAngina: chr [1:918] "N" "N" "N" "Y" ...
##  $ Oldpeak       : num [1:918] 0 1 0 1.5 0 0 0 0 1.5 0 ...
##  $ ST_Slope      : chr [1:918] "Up" "Flat" "Up" "Flat" ...
##  $ HeartDisease  : num [1:918] 0 1 0 1 0 0 0 0 1 0 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   Age = col_double(),
##   ..   Sex = col_character(),
##   ..   ChestPainType = col_character(),
##   ..   RestingBP = col_double(),
##   ..   Cholesterol = col_double(),
##   ..   FastingBS = col_double(),
##   ..   RestingECG = col_character(),
##   ..   MaxHR = col_double(),
##   ..   ExerciseAngina = col_character(),
##   ..   Oldpeak = col_double(),
##   ..   ST_Slope = col_character(),
##   ..   HeartDisease = col_double()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

```r
# First rows of data
head(data)
```

```
## # A tibble: 6 x 12
##     Age Sex   ChestPainType RestingBP Cholesterol FastingBS RestingECG MaxHR
##   <dbl> <chr> <chr>             <dbl>       <dbl>     <dbl> <chr>      <dbl>
## 1    40 M     ATA                 140         289         0 Normal       172
## 2    49 F     NAP                 160         180         0 Normal       156
## 3    37 M     ATA                 130         283         0 ST            98
## 4    48 F     ASY                 138         214         0 Normal       108
## 5    54 M     NAP                 150         195         0 Normal       122
## 6    39 M     NAP                 120         339         0 Normal       170
## # ... with 4 more variables: ExerciseAngina <chr>, Oldpeak <dbl>,
## #   ST_Slope <chr>, HeartDisease <dbl>
```

2.2 Data cleaning

In order to prevent process data with missing values, it is necessary observe is the is any NA.

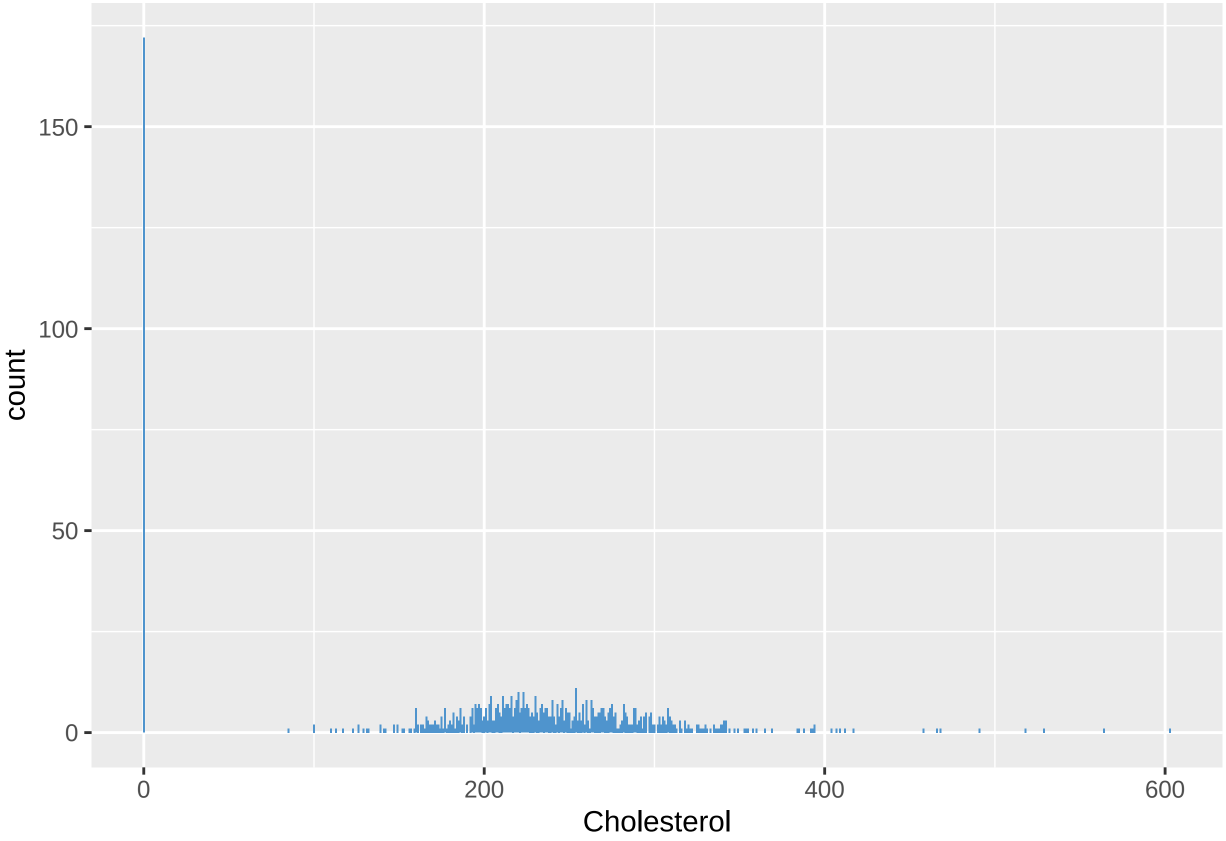
```
is.na(data) %>% sum()
```

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

As could be seen, there is not any NA or missing values.

Anyway, if 'Cholesterol' feature is observed in detail, there are several '0's that must be taken into account. The next bar chart represent the frequency of the values of 'Cholesterol' feature.

```
data %>% ggplot(aes(Cholesterol)) + geom_bar() + geom_bar(fill=coloursOfTheProject1)
```



```
values_with_0 <- data %>% filter(Cholesterol == 0) %>% nrow()
percentage_0 <- round(values_with_0 / nrow(data) * 100, 2)
```

The 172 values with zeros in 'Cholesterol' represent the 18.74% of the total (very high percentage of the data). It could be interpreted as the missing values of 'Cholesterol' in the original data was completed with 'zero' values, which is necessary to several machine learning techniques.

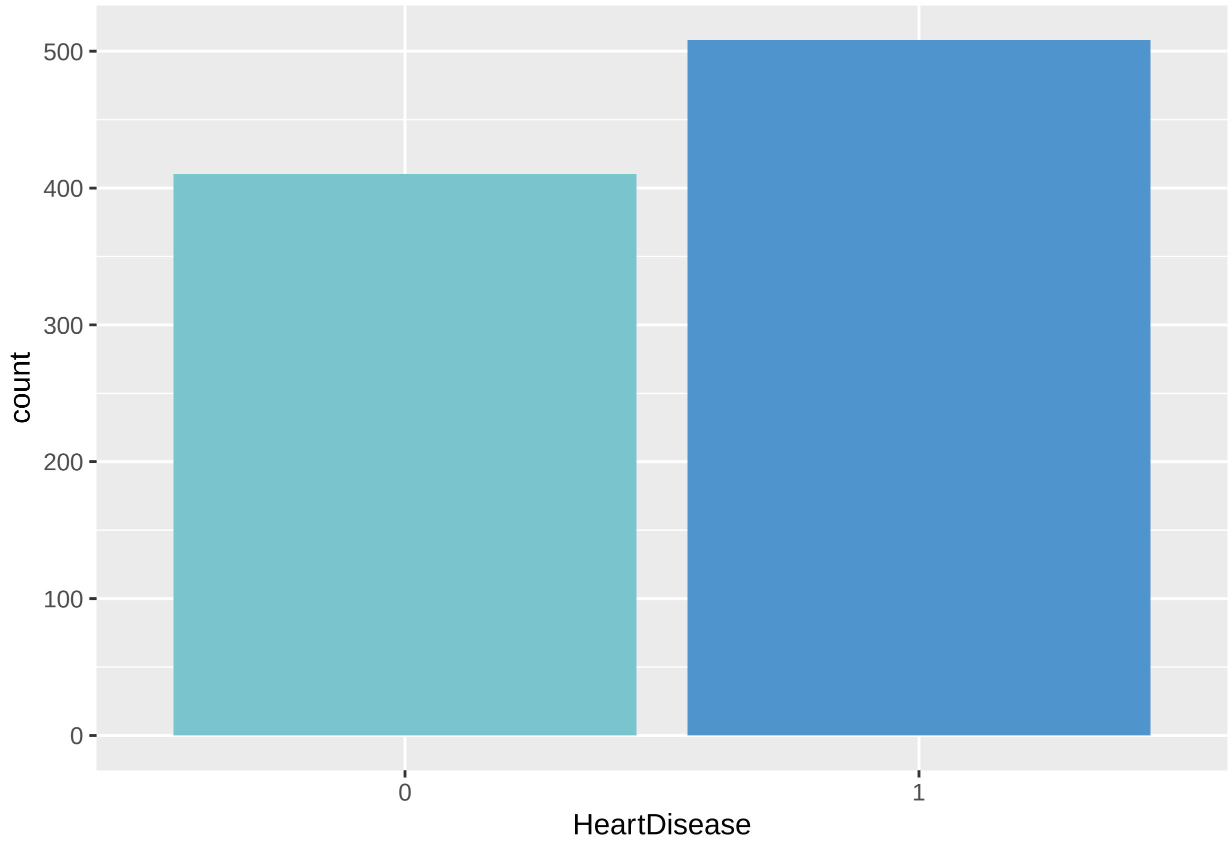
Section '1.2 Data structure' of this report shows that several columns (features) of the data are character columns with only a few different options of each feature. Also, in a great variety of machine learning techniques is more efficient convert this character columns in factors. So, the columns *Sex*, *ChestPainType*, *RestingECG*, *ExerciseAngina*, and *ST_Slope* are converted into factors. *HeartDisease* column is also converted into factor because it contains discrete values (0 or 1).

```
# Convert features in factors
data <- data %>% mutate(Sex = as.factor(Sex)) %>%
  mutate(ChestPainType = as.factor(ChestPainType)) %>%
  mutate(RestingECG = as.factor(RestingECG)) %>%
  mutate(ExerciseAngina = as.factor(ExerciseAngina)) %>%
  mutate(ST_Slope = as.factor(ST_Slope)) %>%
  mutate(HeartDisease = as.factor(HeartDisease))
```

2.3 Data exploration and data visualization

The first step is explore the *HeartDisease* feature. It storage the main important feature: 0 if it has not heart disease, 1 if it has it. The next graph represents the *HeartDisease* distribution.

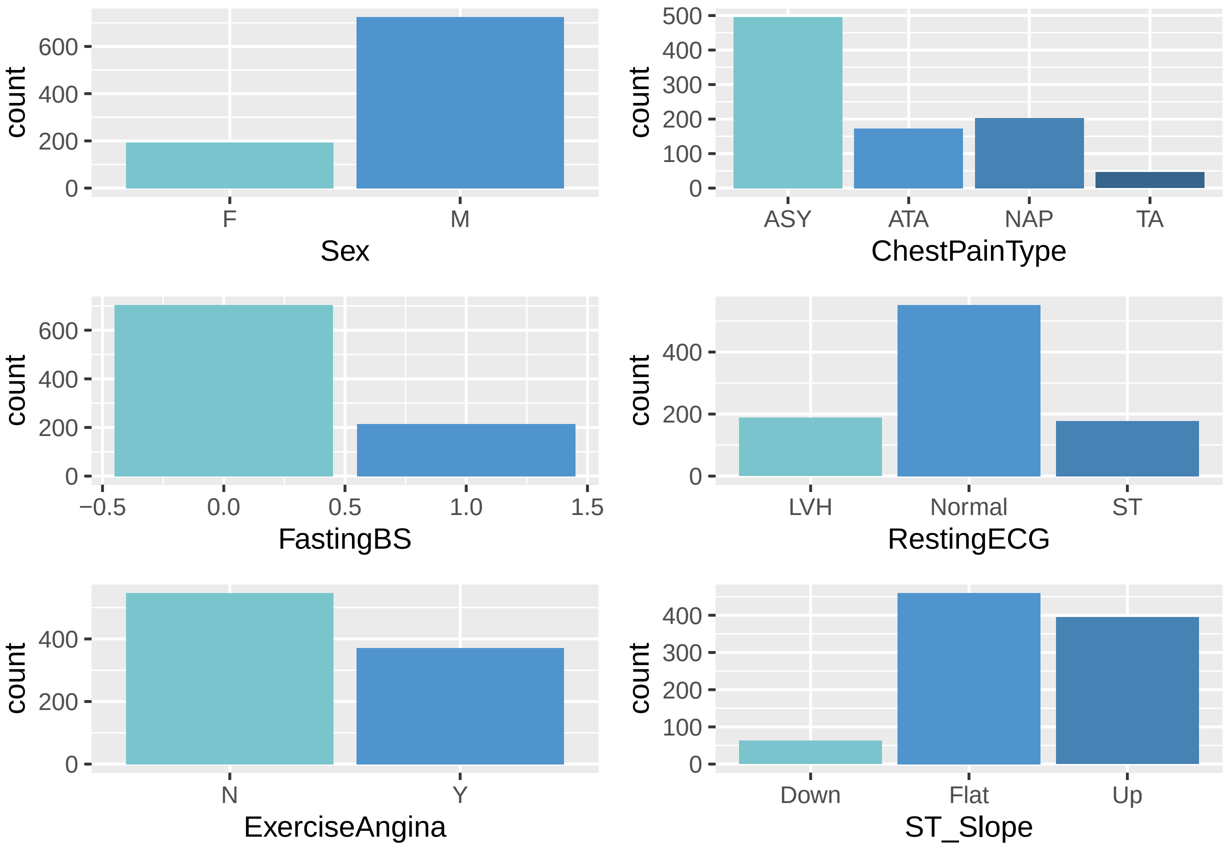
```
data %>% ggplot(aes(HeartDisease)) +
  geom_bar(fill=coloursOfTheProject2)
```



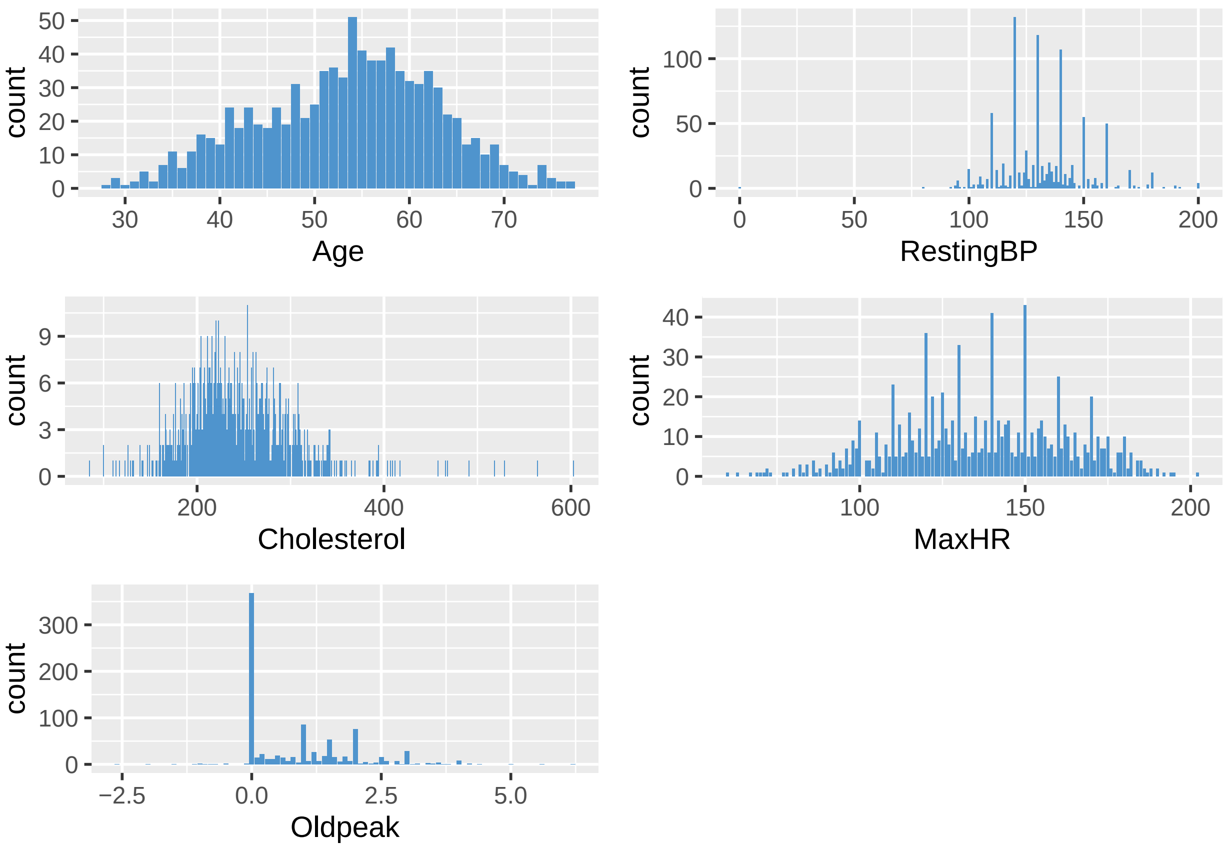
As it could be observed, the *HeartDisease* feature is well balanced.

In order to achieve a general understanding of each feature, its distribution will be presented (separated in discrete and continuous features).

```r
# Discrete features
d1 <- data %>% ggplot(aes(Sex)) + geom_bar(fill=coloursOfTheProject2)
d2 <- data %>% ggplot(aes(ChestPainType)) + geom_bar(fill = coloursOfTheProject4)
d3 <- data %>% ggplot(aes(FastingBS)) + geom_bar(fill=coloursOfTheProject2)
d4 <- data %>% ggplot(aes(RestingECG)) + geom_bar(fill=coloursOfTheProject3)
d5 <- data %>% ggplot(aes(ExerciseAngina)) + geom_bar(fill=coloursOfTheProject2)
d6 <- data %>% ggplot(aes(ST_Slope)) + geom_bar(fill=coloursOfTheProject3)
grid.arrange(d1, d2, d3, d4, d5, d6, ncol=2)
```



```r
# Continuous features
c1 <- data %>% ggplot(aes(Age)) + geom_bar(fill=coloursOfTheProject1)
c2 <- data %>% ggplot(aes(RestingBP)) + geom_bar(fill=coloursOfTheProject1)
c3 <- data %>% filter(Cholesterol>0) %>%
  ggplot(aes(Cholesterol)) + geom_bar(fill=coloursOfTheProject1)
c4 <- data %>% ggplot(aes(MaxHR)) + geom_bar(fill=coloursOfTheProject1)
c5 <- data %>% ggplot(aes(Oldpeak)) + geom_bar(fill=coloursOfTheProject1)
grid.arrange(c1, c2, c3, c4, c5, ncol=2)
```



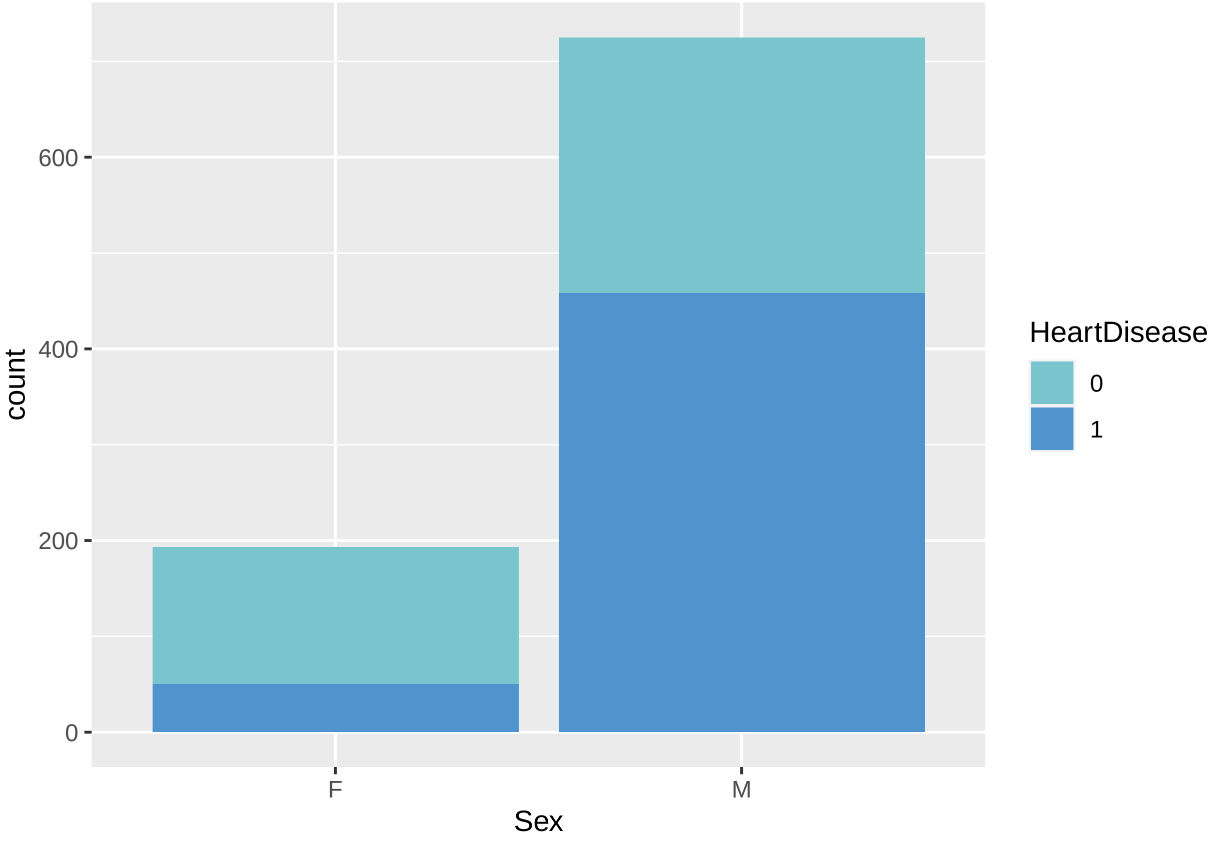
For Cholesterol feature, the zero values were filtered in order to observe the distribution of the other values.

Now, for each of the features, a table and a column chart of its distribution will be presented. These graphs will also distinguish if the different categories of the feature has the disease or not. Reminder: “1” means that this person has the disease, and “0” means that this person has not the disease.

```
# 1. Sex
data %>% group_by(Sex) %>%
  summarize(HeartDisease = mean(HeartDisease == 1), n=n()) %>% knitr::kable()
```

Sex	HeartDisease	n
F	0.2590674	193
M	0.6317241	725

```
data %>% ggplot(aes(Sex, fill=HeartDisease)) + geom_bar() +
  scale_fill_manual(values=coloursOfTheProject2)
```



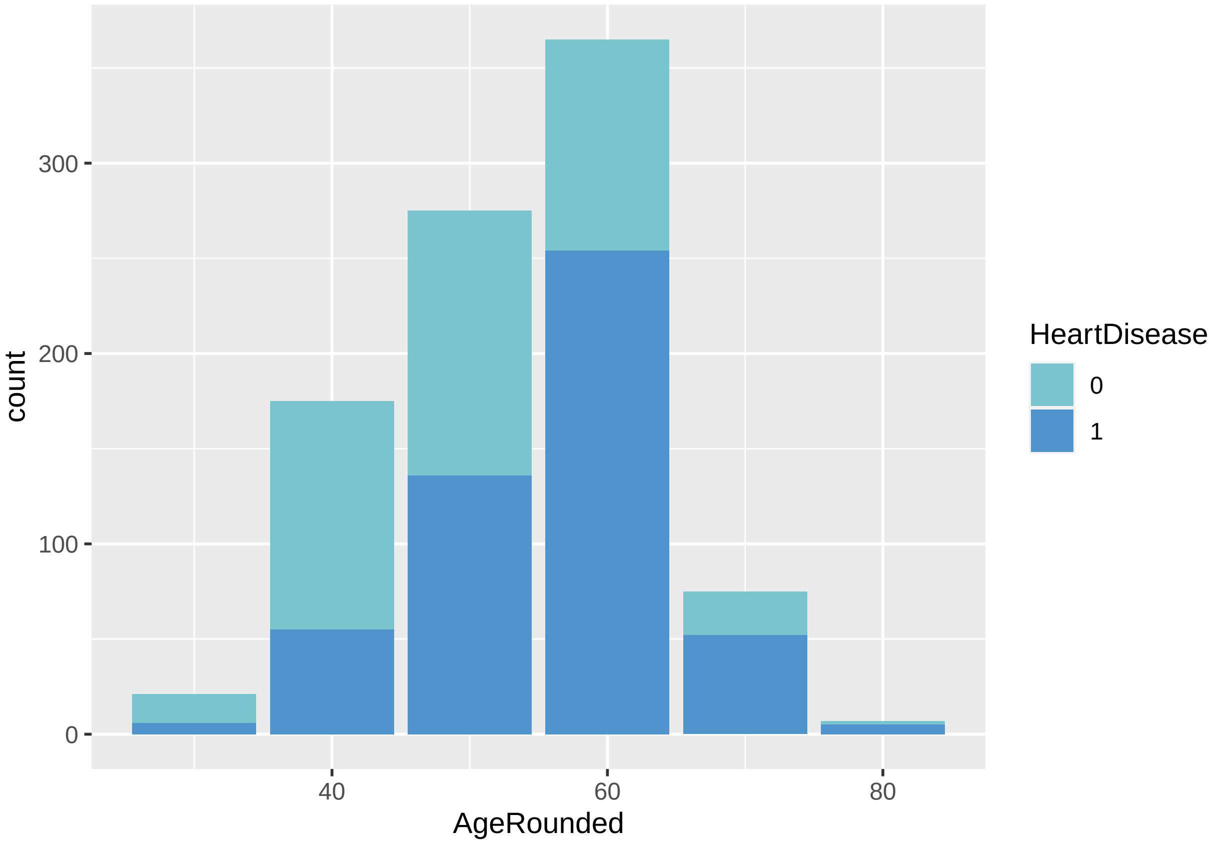
As could be observed, there are more information related to male gender. Another conclusion is that males have clearly a higher probability to suffer a heart disease.

For the purpose of seeing a more representative age column chart, the age is rounded to the nearest tens.

```
# 2. Age
data %>% mutate(AgeRounded = round(Age, -1)) %>%
  group_by(AgeRounded) %>%
  summarize(HeartDisease = mean(HeartDisease == 1), n=n()) %>% knitr::kable()
```

AgeRounded	HeartDisease	n
30	0.2857143	21
40	0.3142857	175
50	0.4945455	275
60	0.6958904	365
70	0.6933333	75
80	0.7142857	7

```
data %>% mutate(AgeRounded = round(Age, -1)) %>%
  ggplot(aes(AgeRounded, fill=HeartDisease)) + geom_bar() +
  scale_fill_manual(values=coloursOfTheProject2)
```

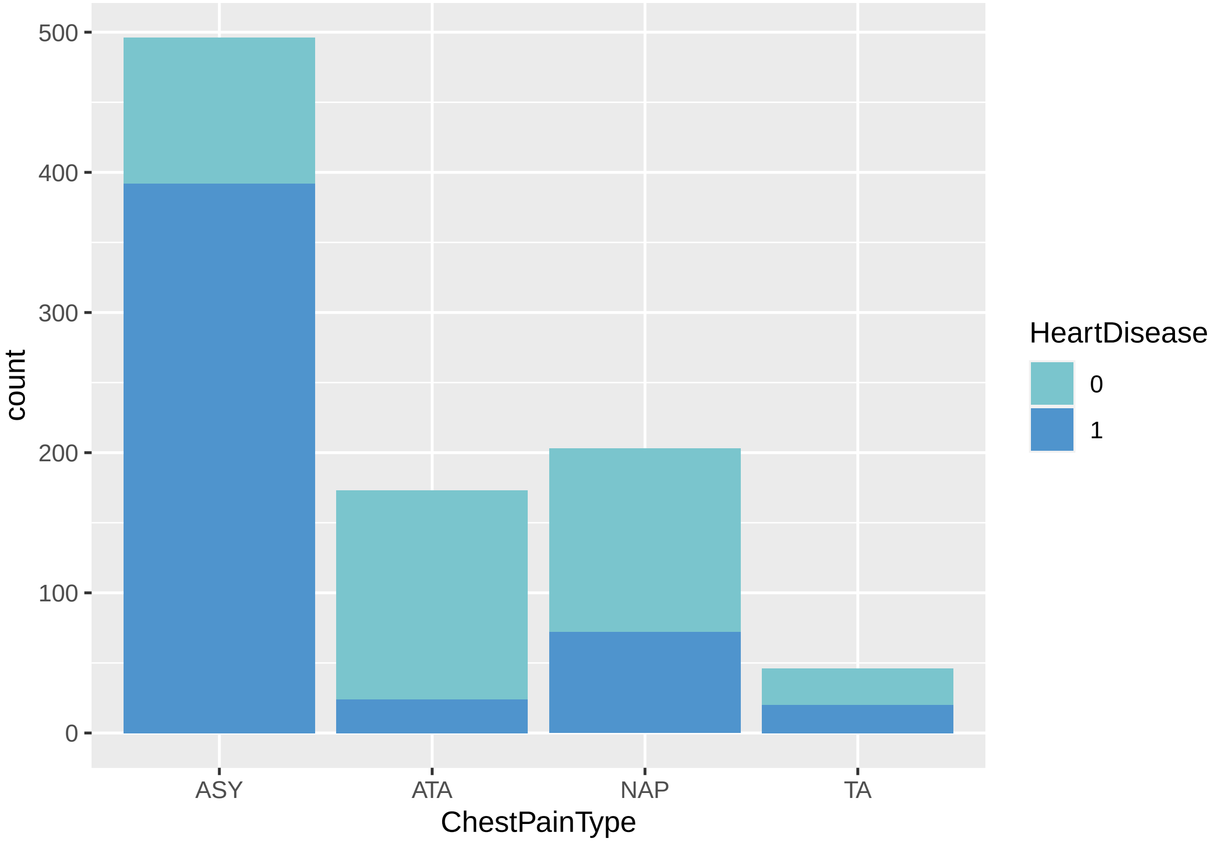


As could be expected, the older the person is, the higher its probability to suffer a heart problem.

```r
# 3. ChestPainType
data %>% group_by(ChestPainType) %>%
  summarize(HeartDisease = mean(HeartDisease == 1), n=n()) %>% knitr::kable()
```

ChestPainType	HeartDisease	n
ASY	0.7903226	496
ATA	0.1387283	173
NAP	0.3546798	203
TA	0.4347826	46

```r
data %>% ggplot(aes(ChestPainType, fill=HeartDisease)) + geom_bar() +
  scale_fill_manual(values=coloursOfTheProject2)
```

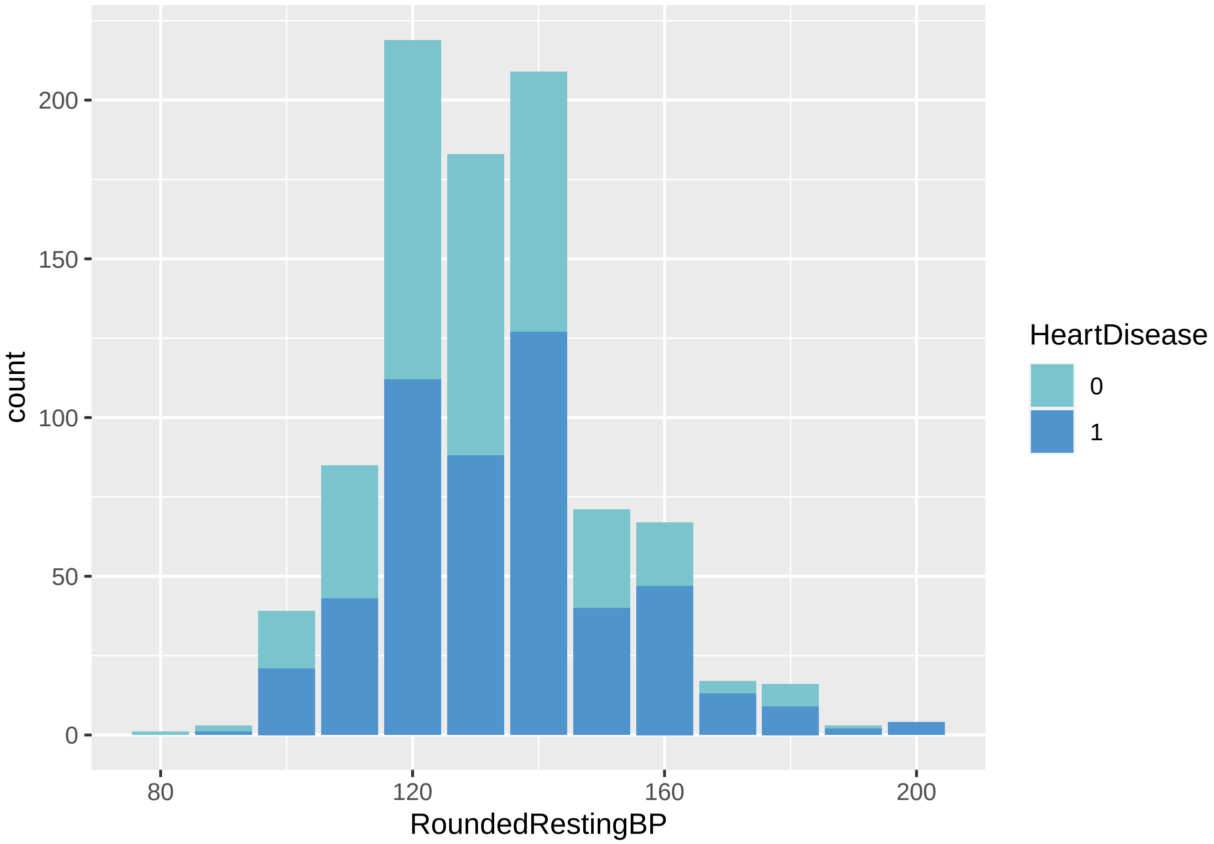


The majority of people are asymptomatic and also, these people have the higher probability of developing heart disease.

```r
# 4. RestingBP
data %>% mutate(RoundedRestingBP = round(RestingBP, -1)) %>%
  group_by(RoundedRestingBP) %>%
  summarize(HeartDisease = mean(HeartDisease == 1), n=n()) %>% knitr::kable()
```

RoundedRestingBP	HeartDisease	n
0	1.0000000	1
80	0.0000000	1
90	0.3333333	3
100	0.5384615	39
110	0.5058824	85
120	0.5114155	219
130	0.4808743	183
140	0.6076555	209
150	0.5633803	71
160	0.7014925	67
170	0.7647059	17
180	0.5625000	16
190	0.6666667	3
200	1.0000000	4

```r
data %>% mutate(RoundedRestingBP = round(RestingBP, -1)) %>%
  filter(RoundedRestingBP>0) %>%
  ggplot(aes(RoundedRestingBP, fill=HeartDisease)) + geom_bar() +
  scale_fill_manual(values=coloursOfTheProject2)
```



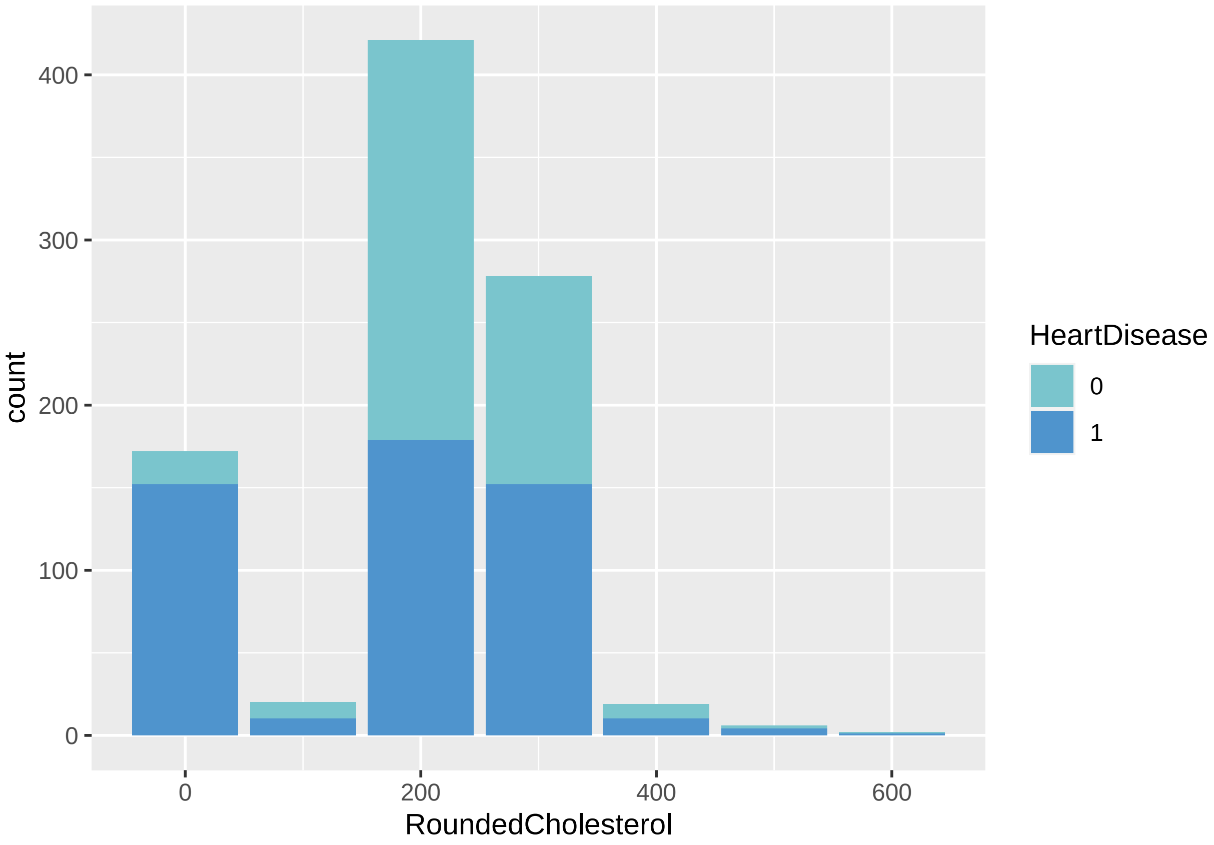
The values of the *RestingBP* feature were rounded trying to gather insights of the graph. As could be seen in the table, there is one person with 0 *RestingBP*. This value was filtered in the graph to only show the relevant values.

The most common values of resting blood pressure are between 120 and 150. If the people have a resting blood pressure higher that 160 (mm Hg), the chances of endure a heart disease sharply increases.

```
# 5. Cholesterol
data %>% mutate(RoundedCholesterol = round(Cholesterol, -2)) %>%
  group_by(RoundedCholesterol) %>%
  summarize(HeartDisease = mean(HeartDisease == 1), n=n()) %>% knitr::kable()
```

RoundedCholesterol	HeartDisease	n
0	0.8837209	172
100	0.5000000	20
200	0.4251781	421
300	0.5467626	278
400	0.5263158	19
500	0.6666667	6
600	0.5000000	2

```
data %>% mutate(RoundedCholesterol = round(Cholesterol, -2)) %>%
  ggplot(aes(RoundedCholesterol, fill=HeartDisease)) + geom_bar() +
  scale_fill_manual(values=coloursOfTheProject2)
```



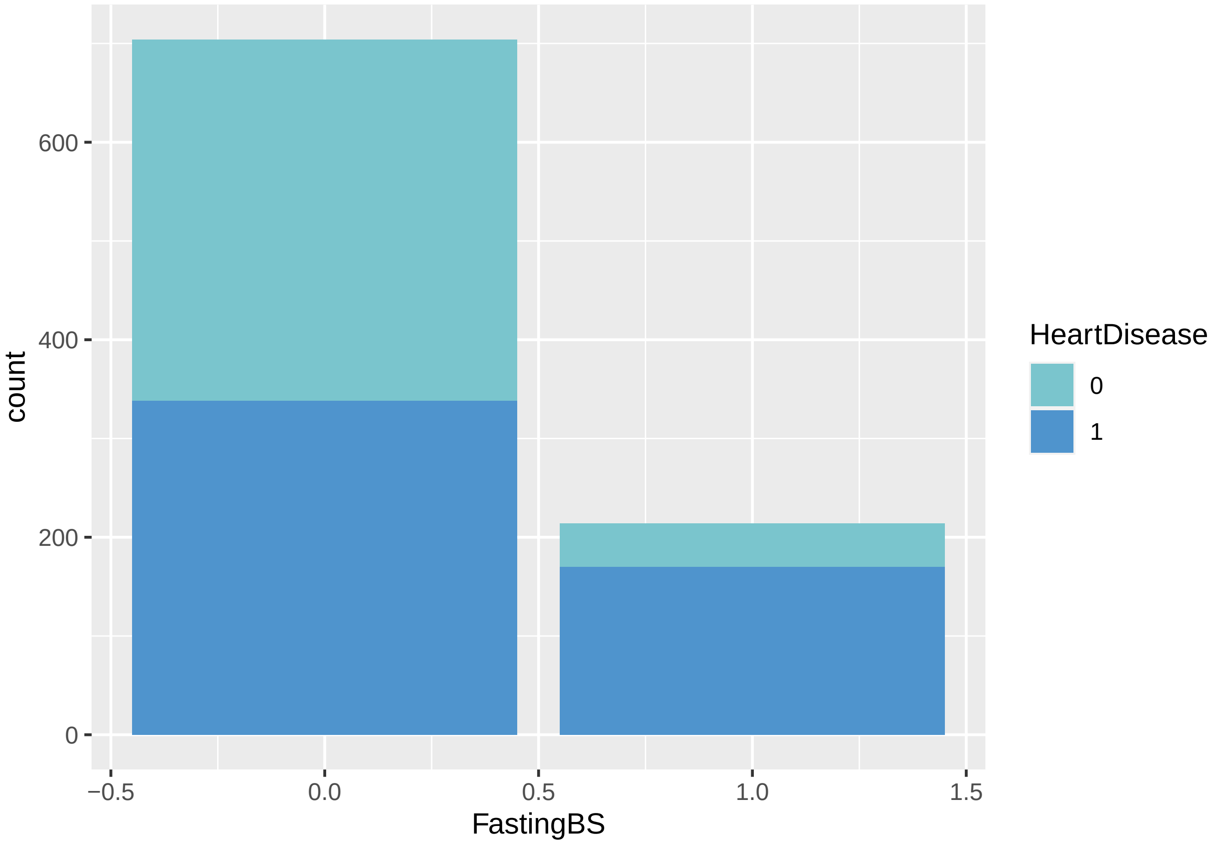
That feature is rounded to the nearest hundred. The most common values of cholesterol are between 200 and 300.

As was mentioned in section *2.2 Data cleaning*, the 18.74% of the total have a cholesterol level of 0. It could be explained as the cholesterol level were not recorded. In the table and in the graph could be observed that these people has a higher probability to suffer a heart disease.

```
# 6. FastingBS
data %>% group_by(FastingBS) %>%
  summarize(HeartDisease = mean(HeartDisease == 1), n=n()) %>% knitr::kable()
```

FastingBS	HeartDisease	n
0	0.4801136	704
1	0.7943925	214

```
data %>% ggplot(aes(FastingBS, fill=HeartDisease)) + geom_bar() +
  scale_fill_manual(values=coloursOfTheProject2)
```

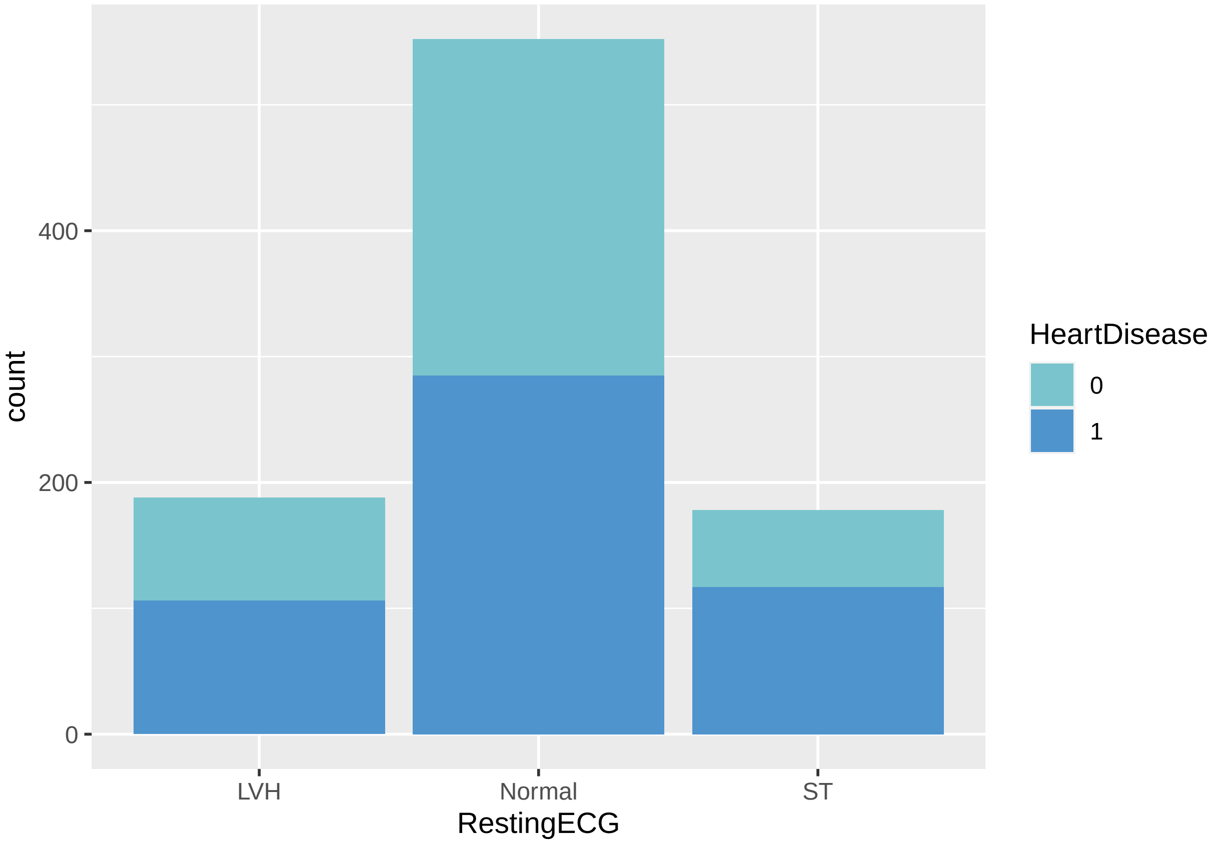


It is pretty obvious that it the fasting blood sugar is higher than 120 mg/dl (reminder: *FastingBS* feature is 1 if the fasting blood sugar is higher than 120 mg/dl; otherwise is 0), the probability of enduring the heart problem increases.

```
# 7. RestingECG
data %>% group_by(RestingECG) %>%
  summarize(HeartDisease = mean(HeartDisease == 1), n=n()) %>% knitr::kable()
```

RestingECG	HeartDisease	n
LVH	0.5638298	188
Normal	0.5163043	552
ST	0.6573034	178

```
data %>% ggplot(aes(RestingECG, fill=HeartDisease)) + geom_bar() +
  scale_fill_manual(values=coloursOfTheProject2)
```

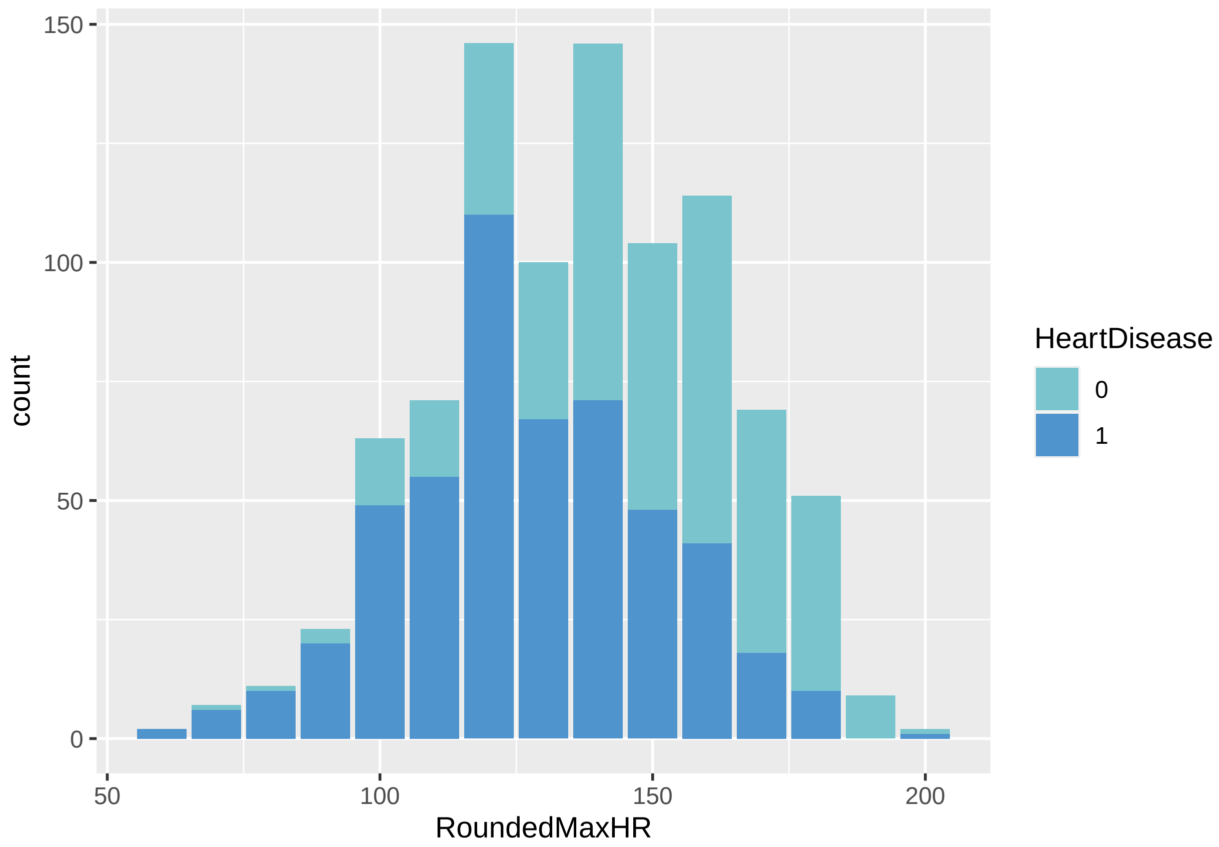


This feature is related to the results of the resting electrocardiogram. The *normal* electrocardiogram is the most popular value. The highest probability of suffer a heart disease is observed when people has a *ST* electrocardiogram (in *section 2.1 Data structure* the ST electrocardiogram was described).

```r
# 8. MaxHR
data %>% mutate(RoundedMaxHR = round(MaxHR, -1)) %>%
  group_by(RoundedMaxHR) %>%
  summarize(HeartDisease = mean(HeartDisease == 1), n=n()) %>% knitr::kable()
```

RoundedMaxHR	HeartDisease	n
60	1.0000000	2
70	0.8571429	7
80	0.9090909	11
90	0.8695652	23
100	0.7777778	63
110	0.7746479	71
120	0.7534247	146
130	0.6700000	100
140	0.4863014	146
150	0.4615385	104
160	0.3596491	114
170	0.2608696	69
180	0.1960784	51
190	0.0000000	9
200	0.5000000	2

```r
data %>% mutate(RoundedMaxHR = round(MaxHR, -1)) %>%
  ggplot(aes(RoundedMaxHR, fill=HeartDisease)) + geom_bar() +
  scale_fill_manual(values=coloursOfTheProject2)
```

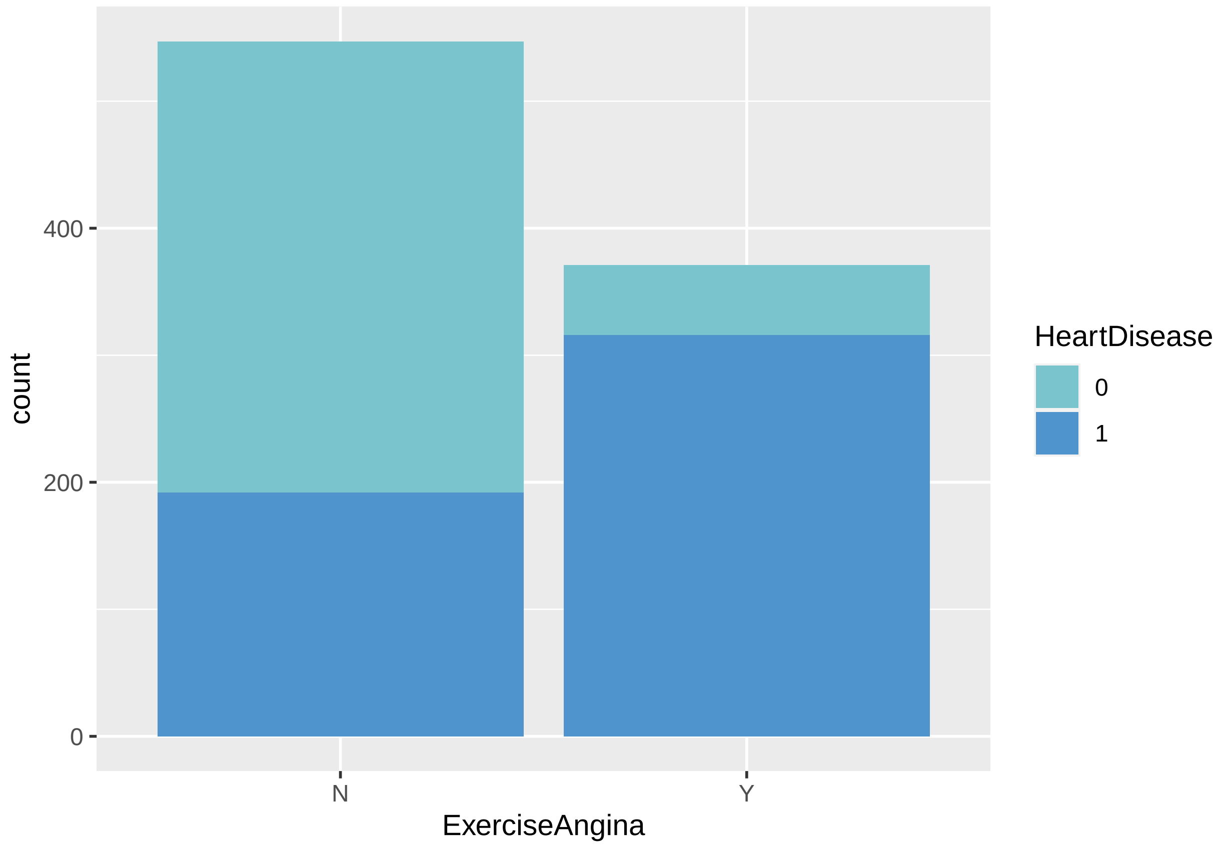


The maximum heart rate achieved is rounded to the nearest tens. The lower the rounded *MaxHR* is, the higher its probability to endure a heart problem.

```r
# 9. ExerciseAngina
data %>% group_by(ExerciseAngina) %>%
  summarize(HeartDisease = mean(HeartDisease == 1), n=n()) %>% knitr::kable()
```

ExerciseAngina	HeartDisease	n
N	0.3510055	547
Y	0.8517520	371

```r
data %>% ggplot(aes(ExerciseAngina, fill=HeartDisease)) + geom_bar() +
  scale_fill_manual(values=coloursOfTheProject2)
```

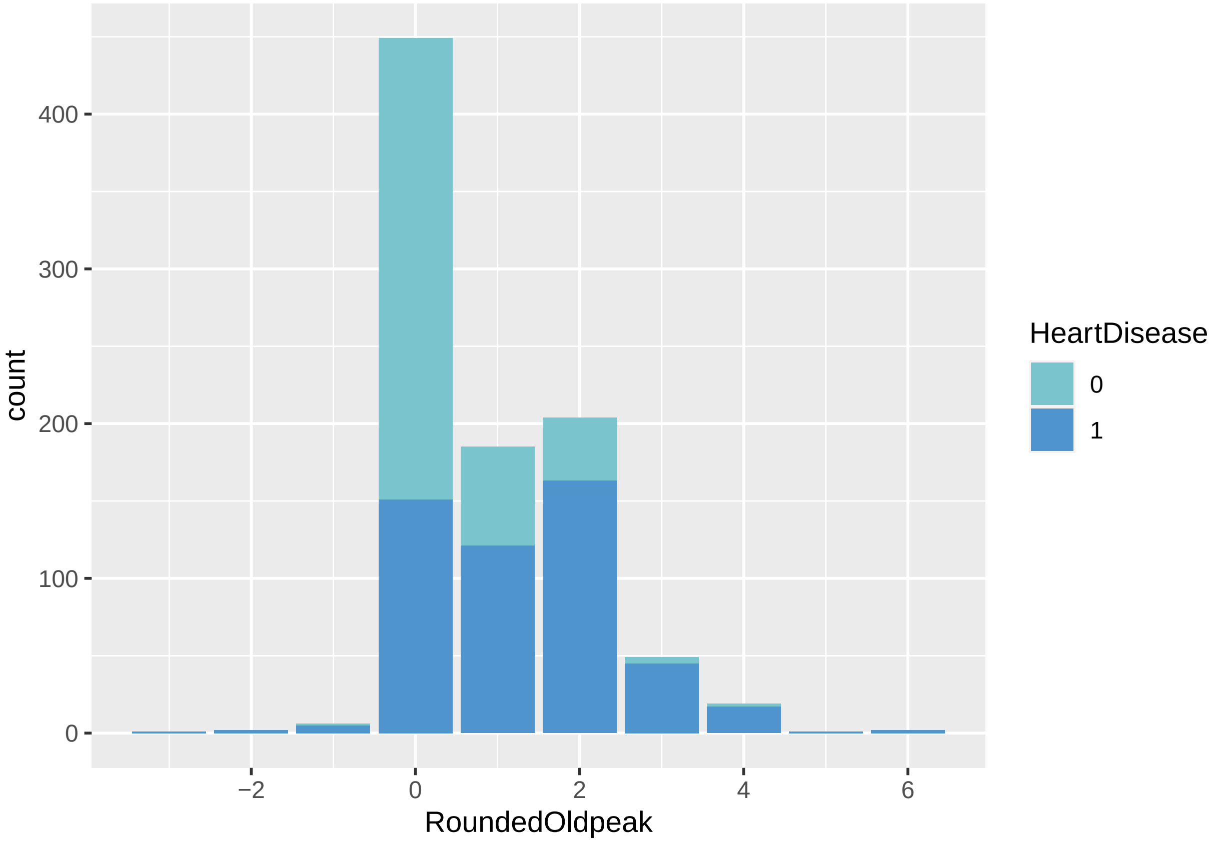


As could be observed, the odds of endure a heart disease clearly increases when people has exercise-inducted angina.

```
# 10. Oldpeak
data %>% mutate(RoundedOldpeak = round(Oldpeak)) %>%
  group_by(RoundedOldpeak) %>%
  summarize(HeartDisease = mean(HeartDisease == 1), n=n()) %>% knitr::kable()
```

RoundedOldpeak	HeartDisease	n
-3	1.0000000	1
-2	1.0000000	2
-1	0.8333333	6
0	0.3363029	449
1	0.6540541	185
2	0.7990196	204
3	0.9183673	49
4	0.8947368	19
5	1.0000000	1
6	1.0000000	2

```
data %>% mutate(RoundedOldpeak = round(Oldpeak)) %>%
  ggplot(aes(RoundedOldpeak, fill=HeartDisease)) + geom_bar() +
  scale_fill_manual(values=coloursOfTheProject2)
```

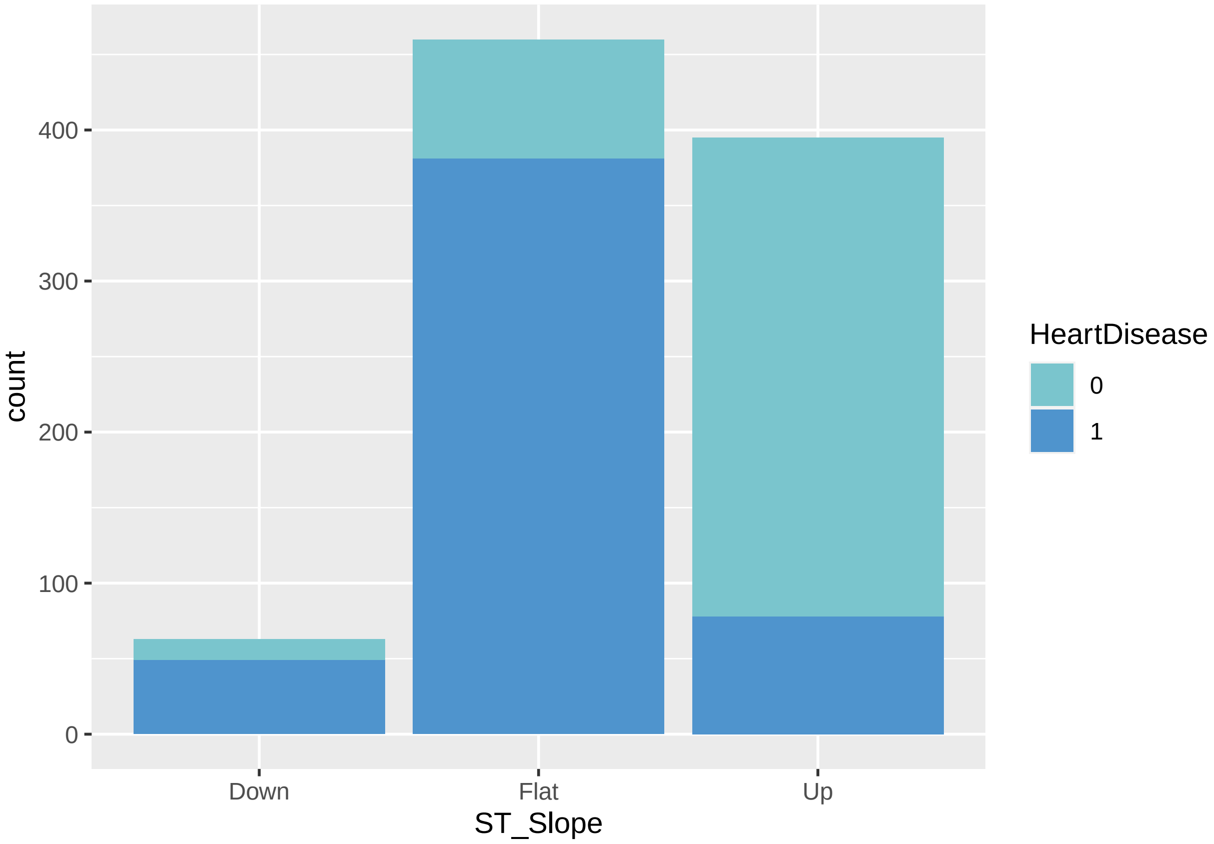


The *oldpeak* feature is rounded to the nearest unit. It is shown that the most frequent value is zero; and also, different from zero values have higher probabilities of suffering a heart disease.

```
# 11. ST_Slope
data %>% group_by(ST_Slope) %>%
  summarize(HeartDisease = mean(HeartDisease == 1), n=n()) %>% knitr::kable()
```

ST_Slope	HeartDisease	n
Down	0.7777778	63
Flat	0.8282609	460
Up	0.1974684	395

```
data %>% ggplot(aes(ST_Slope, fill=HeartDisease)) + geom_bar() +
  scale_fill_manual(values=coloursOfTheProject2)
```



Down and flat slope of the peak exercise ST segment means a higher probability to suffer a heart disease.

2.4 Modeling approach

There are multiple machine learning algorithms available. In this project 6 methods are selected. With the three best accuracy results, an ensemble is generated.

These are the models used:

- Model 1: GLM
- Model 2: KNN
- Model 3: LDA
- Model 4: rpart
- Model 5: Rborist
- Model 6: GBM
- Model 7: Ensemble

In order to train the data, a split is required. The *createDataPartition* function is used, and the data is split in 50% training and 50% test, that lead to better accuracy results (it could be explained due to the small number of rows).

```r
# Creating data partition:
set.seed(2022, sample.kind = "Rounding")
index <- createDataPartition(data$HeartDisease, 1, 0.5, FALSE)
train <- data %>% slice(-index)
test <- data %>% slice(index)
```

2.4.1 Model 1: GLM

The first algorithm selected is GLM (generalized linear model), which estimates regression models for specific results through exponential distributions. For this model, *glm* function might be used, but *train* function is the chosen due to the ease of testing different algorithms (all suported by *caret* package). In all models *train* function is used, and the algorithm is selected in the *method* argument.

```r
# GLM
fit_glm <- train(HeartDisease ~ ., data = train, method = "glm")
pred_glm <- predict(fit_glm, test)
acc_glm <- mean(pred_glm == test$HeartDisease)
```

The accuracy of GLM algorithm, compared to the real outcome of the *HeartDisease* is 0.8518519.

2.4.2 Model 2: kNN

The second algorithm selected is kNN (k-Nearest Neighbors) which classify neighbors with similar characteristics and choose the most common outcome of each of these group neighbors.

```r
# kNN
fit_knn <- train(HeartDisease ~ ., data = train, method = "knn")
pred_knn <- predict(fit_knn, test)
acc_knn <- mean(pred_knn == test$HeartDisease)
```

The accuracy of kNN algorithm for this dataset is 0.6971678. This accuracy value is clearly lower than GLM algorithm.

2.4.3 Model 3: LDA

The third algorithm used is LDA (Linear Discriminant Analysis) which is a good option for classification predictive modeling problems specially when there are a great variety of parameters.

```
# LDA
fit_lda <- train(HeartDisease ~ ., data = train, method = "lda")
pred_lda <- predict(fit_lda, test)
acc_lda <- mean(pred_lda == test$HeartDisease)
```

The accuracy of this LDA algorithm is 0.8474946.

2.4.4 Model 4: rpart

The forth algorithm selected is rpart which is a decision tree method that split the data into different heterogeneous groups.

```
# rpart
fit_rpart <- train(HeartDisease ~ ., data = train, method = "rpart")
pred_rpart <- predict(fit_rpart, test)
acc_rpart <- mean(pred_rpart == test$HeartDisease)
```

The final accuracy of this model is 0.8409586.

2.4.5 Model 5: Rborist

In the fifth model, the algorithm used is rborist which is a random forest, this means that is an ensemble of several decision trees and which outcome will be the average prediction of these decision trees.

```
# Rborist
fit_Rborist <- train(HeartDisease ~ ., data = train, method = "Rborist")
pred_Rborist <- predict(fit_Rborist, test)
acc_Rborist <- mean(pred_Rborist == test$HeartDisease)
```

The accuracy of this method is 0.8845316, the best precision of the first five models.

2.4.6 Model 6: GBM

The sixth algorithm selected is GBM (Gradient Boosting Machines) which is an ensemble that combines several weak predictors converting it into a strong predictor.

```
# GBM
fit_gbm <- train(HeartDisease ~ ., data = train, method = "gbm")
pred_gbm <- predict(fit_gbm, test)
acc_gbm <- mean(pred_gbm == test$HeartDisease)
```

For this model, the obtained accuracy is 0.8496732.

2.4.7 Model 7: Ensemble

The seventh and last model is an ensemble of the best three methods used. Based on the accuracy calculated, the the best methods are: *GLM*, *Rborist* and *GBM*. In order to achieve an average of these three methods, as the prediction values are factors (of *HeartDisease*), a conversion to numeric values is required.

Despite the factors values were "0" and "1", when these values are converted to numeric, their values are "1" and "2". This is the reason that a "-3" is used to calculate the final values. If the final value is greater than 1, it will be a "1"; otherwise it will be a "0".

```
# Ensemble
ensemble <- tibble(glm = pred_glm, Rborist = pred_Rborist, gbm = pred_gbm)
ensemble <- ensemble %>%
  mutate(final = as.numeric(glm) + as.numeric(Rborist) + as.numeric(gbm) - 3) %>%
  mutate(final = ifelse(final>1, 1, 0)) %>%
  mutate(final = as.factor(final))
acc_ensemble <- mean(ensemble$final == test$HeartDisease)
```

The accuracy of that particular ensemble is 0.8627451.

3 Results

Finally, all the accuracy values of the different machine learning methods are gather in a table. These are the results:

```r
results <- tibble(method = "GLM", accuracy = acc_glm) %>%
  bind_rows(tibble(method="kNN", accuracy = acc_knn)) %>%
  bind_rows(tibble(method="LDA", accuracy = acc_lda)) %>%
  bind_rows(tibble(method="rpart", accuracy = acc_rpart)) %>%
  bind_rows(tibble(method="Rborist", accuracy = acc_Rborist)) %>%
  bind_rows(tibble(method="GBM", accuracy = acc_gbm)) %>%
  bind_rows(tibble(method="Ensemble", accuracy = acc_ensemble))
results %>% knitr::kable()
```

method	accuracy
GLM	0.8518519
kNN	0.6971678
LDA	0.8474946
rpart	0.8409586
Rborist	0.8845316
GBM	0.8496732
Ensemble	0.8627451

As could be observed, the best machine learning method used for this dataset is *Rborist*. Surprisingly, the ensemble method which uses the top 3 methods obtain worse accuracy than the *Rborist* method.

Finally, the importance of the features for the best method used, *Rborist*, are showed:

```r
# Importance of the features in Rborist method (best method)
varImp(fit_Rborist)$importance %>% arrange(desc(Overall)) %>% knitr::kable()
```

	Overall
Oldpeak	100.0000000
ST_SlopeUp	20.1685830
ST_SlopeFlat	13.7350900
ExerciseAnginaY	12.3314784
Cholesterol	12.1869957
MaxHR	11.8230687
Age	10.7273932
RestingBP	8.6740266
SexM	7.7131755
ChestPainTypeATA	7.2473624
ChestPainTypeNAP	2.2861044
FastingBS	2.1183776
RestingECGNormal	0.7847762
ChestPainTypeTA	0.1214163
RestingECGST	0.0000000

As could be observed, for this method the main important feature is *Oldpeak*, followed by *ST_Slope* (up and flat responses) and *ExerciseAngina* (yes response). This could be corroborated seeing the graphs in section *2.3 Data exploration and data visualization*.

4 Conclusion

To sum up, this dataset presents an interesting approximation to health data, understanding the importance of healthy habits which could reduce the probability of developing heart disease.

It has been demonstrated that age, fasting blood sugar, low heart rate achieved and a down or flat ST slope affect strongly in the likelihood of suffering a heart problem. And also, men have a bigger probability of developing heart disease.

This project has used six different machine learning algorithms (plus an ensemble), but there is a great variety of more of them which might be used trying to obtain better accuracy results.

One possible limitation is the number of observations (918). With more observations the predictions could be more accurate.

Another prospective action could be tuning any of the algorithms used in the project as a way of improving the accuracy results.