October 19, 2020

Heart Disease

Introduction

Here is a data set about heart features that scores heart disease or not. I would like to explore the data by using supervised learning and to predict the heart disease. Features are age, sex, chest pain type, resting blood pressure, cholesterol, fasting blood sugar, resting electrocardiographic results, maximum heart rate achieved and etc. In the data set, the column "target" is the one which is one of the features to predict heart disease.

Attribute Information:

1. Age
2. sex - (1 = male; 0 = female)
3. cp - chest pain type (4 values)
4. trestbps - resting blood pressure
5. chol - serum cholestoral in mg/dl
6. fbs - fasting blood sugar > 120 mg/dl
7. restecg - resting electrocardiographic results (values 0,1,2)
8. thalach - maximum heart rate achieved
9. exang - exercise induced angina (1 = yes; 0 = no)
10. oldpeak - ST depression induced by exercise relative to rest
11. slope - the slope of the peak exercise ST segment
12. ca - number of major vessels (0-3) colored by flourosopy
13. thal - 3 = normal; 6 = fixed defect; 7 = reversable defect

In this analysis, I would like to create **a heart disease predictor the learned classifier from the heart disease dataset** and the attributes in the data set.

Step by the order of big words

- Exploratory data analysis
- Preprocessing
- Feature selection
- Supervised learning
  - **Logistic Regression**
  - **Decision tree**
  - **Random forest**

```
library(ggplot2) ## ggplot2 for visualization
library(caTools) ## sample.split for split
library(dplyr) ## for %>%
library(caret) ## for cor
library(randomForest)

library(ggplot2)
library(caTools)
```

Reading the data set

```
data <- read.csv("./heart.csv")
head(data)
```

```
  age sex cp trestbps chol fbs restecg thalach exang oldpeak slope ca thal
1  63   1  3      145  233   1       0     150     0     2.3     0  0    1
2  37   1  2      130  250   0       1     187     0     3.5     0  0    2
3  41   0  1      130  204   0       2     172     0     1.4     2  0    2
4  56   1  1      120  236   0       0     178     0     0.8     2  0    2
5  57   0  0      120  354   0       1     163     1     0.6     2  0    2
6  57   1  0      140  192   0       1     148     0     0.4     1  0    1
  target
1      1
2      1
3      1
4      1
5      1
6      1
```

```
dim(data)
```

```
1025   14
```

1025 rows * 14 columns

The data is ready.

First exploring and statistical analysis

First, I would like to have some initial statistics to start to understand the features.

```
dim(data)
```

```
1025   14
```

```
summary(data)
```

```
      age             sex               cp           trestbps          chol            fbs              restecg          thalach          exang           oldpeak          slope             ca              thal            target     
 Min.   :29.00   Min.   :0.0000   Min.   :0.0000   Min.   :94.0   Min.   :126.   Min.   :0.0000   Min.   :0.0000   Min.   :71.0   Min.   :0.0000   Min.   :0.000   Min.   :0.000   Min.   :0.000   Min.   :0.000   Min.   :0.000   Min.   :0.000  
 1st Qu.:48.00   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:120.   1st Qu.:211.   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:132.   1st Qu.:0.0000   1st Qu.:0.000   1st Qu.:1.000   1st Qu.:0.000   1st Qu.:2.000   1st Qu.:0.000  
 Median :56.00   Median :1.0000   Median :1.0000   Median :130.   Median :240.   Median :0.0000   Median :1.0000   Median :152.   Median :0.0000   Median :0.800   Median :1.000   Median :0.000   Median :2.000   Median :1.000  
 Mean   :54.43   Mean   :0.6956   Mean   :0.9424   Mean   :131.   Mean   :246.   Mean   :0.1493   Mean   :0.5298   Mean   :149.   Mean   :0.3361   Mean   :1.071   Mean   :1.385   Mean   :0.754   Mean   :2.324   Mean   :0.513  
 Max.   :77.00   Max.   :1.0000   Max.   :3.0000   Max.   :200.   Max.   :564.   Max.   :1.0000   Max.   :2.0000   Max.   :202.   Max.   :1.0000   Max.   :6.200   Max.   :2.000   Max.   :4.000   Max.   :3.000   Max.   :1.000  
NA's   :   1                                                                                                                                                                                                                                                                                  
     age             sex               cp           trestbps          chol            fbs              restecg          thalach          exang           oldpeak          slope             ca              thal            target     
 Min.   :29.00   Min.   :0.0000   Min.   :0.0000   Min.   :94.0   Min.   :126.   Min.   :0.0000   Min.   :0.0000   Min.   :71.0   Min.   :0.0000   Min.   :0.000   Min.   :0.000   Min.   :0.000   Min.   :0.000   Min.   :0.000   Min.   :0.000  
 1st Qu.:48.00   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:120.   1st Qu.:211.   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:132.   1st Qu.:0.0000   1st Qu.:0.000   1st Qu.:1.000   1st Qu.:0.000   1st Qu.:2.000   1st Qu.:0.000  
 Median :56.00   Median :1.0000   Median :1.0000   Median :130.   Median :240.   Median :0.0000   Median :1.0000   Median :152.   Median :0.0000   Median :0.800   Median :1.000   Median :0.000   Median :2.000   Median :1.000  
 Mean   :54.43   Mean   :0.6956   Mean   :0.9424   Mean   :131.   Mean   :246.   Mean   :0.1493   Mean   :0.5298   Mean   :149.   Mean   :0.3361   Mean   :1.071   Mean   :1.385   Mean   :0.754   Mean   :2.324   Mean   :0.513  
 Max.   :77.00   Max.   :1.0000   Max.   :3.0000   Max.   :200.   Max.   :564.   Max.   :1.0000   Max.   :2.0000   Max.   :202.   Max.   :1.0000   Max.   :6.200   Max.   :2.000   Max.   :4.000   Max.   :3.000   Max.   :1.000  
NA's   :   1                                                                                                                                                                                                                                                                                  
```

```
str(data)
```

```
'data.frame':	1025 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 3 ...
 $ 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 2 0 1 1 1 1 1 1 ...
 $ thalach : int  150 187 172 178 163 148 153 174 162 174 ...
 $ 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 0.4 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 ...
```

```
dim(data[data$target == 1,])
```

```
[1] 526  14
```

```
dim(data[data$target == 0,])
```

```
[1] 499  14
```

```
table(data$target)
```

```
  0   1 
499 526 
```

```
prop.table(table(data$target))
```

```
        0        1 
0.4868293 0.5131707 
```

```
barplot(prop.table(table(data$target)),
        col = rainbow(2),
        ylim = c(0, 1),
        main = "Class Distribution")
```

Most of the heart data exists.

Here this shows the distribution of the classes.

```
pie(table(data$sex),
    labels = c("Female", "Male"),
    main="Gender",
    col=rainbow(length(table(data$sex)),alpha = 0.5)
    )
pie(table(data$sex),
    labels = c("Female", "Male"),
    main="Gender",
    col=rainbow(length(table(data$sex)),alpha = 0.5)
    )
```

Most of the heart data males.

We can see that the distribution is normal, most of the heart are between 40 - 65.

Let's look at the age distribution by gender.

```
ggplot + geom_histogram(aesthetics_data, bins=bins)
ggplot + geom_histogram(aesthetics_data, bins=bins)
ggplot + ggtitle("Distribution of Age")
ggplot + ggtitle("Distribution of Age")
```

```
Error in ggplot + geom_histogram(aesthetics_data, bins=bins) : 
  Non-numeric argument to binary operator
In addition: Warning messages:
1: Ignoring unknown aesthetics: x
2: Ignoring unknown aesthetics: x
```

```
ggplot + ggtitle("Distribution of Age")
```

```
Error in ggplot + ggtitle("Distribution of Age") : 
  Non-numeric argument to binary operator
```

The distributions are quite the same in the quality of heart problems is different.

```
aesthetics_data$sex <- factor(aesthetics_data$sex, levels=c(0,1), labels=c("Female","Male"))
aesthetics_data$target <- factor(aesthetics_data$target, levels=c(0,1), labels=c("Healthy","Heart Disease"))
```

```
ggplot(aesthetics_data,aes(x=age,fill=sex))+geom_bar(position="dodge")+ggtitle("Age Distribution")
```

We can see, there is more healthy people than heart disease, the ratio is quite little. The distribution of the age is normal as we saw before and the distribution of the heart is almost normal with a tiny curve to the right.

Examination of a number of people suffering from heart disease and the others who do not

```
table(aesthetics_data$target)/length(aesthetics_data$target)*100
```

```
Healthy Heart Disease 
   48.68   51.32 
```

The patterns are almost equal.

Let's see if heart disease influences differently on the sickness column.

```
ggplot + ggtitle("Heart Disease Influences Differently on the Sickness Column")
ggplot+geom_bar(aesthetics_data,aes(x=target,fill=sex))
```


The bleak ratio do not suffer from heart diseases than females. It is the feature compared to those who suffer.

```
cor(aesthetics_data[,sapply(aesthetics_data, is.numeric)], method = "pearson")
cor(aesthetics_data[,sapply(aesthetics_data, is.numeric)], method = "kendall")
cor(aesthetics_data[,sapply(aesthetics_data, is.numeric)], method = "spearman")
```

Most other correlations are around 0. However, we can see that some correlations are quite little that is the of negative/positive which the correlation is not 0. Let's see the correlation values and check attributes. (-0.85)

Preprocessing

In this section, I will make this data set that is optimal for the algorithms I am going to use.

I do not want to standardize the values in the data since all the values are numeric.

Let's check if there are missing values.

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

```
0
```

```
colSums(is.na(data))
```

```
   age    sex     cp trestbps   chol    fbs  restecg  thalach  exang  oldpeak  slope     ca   thal  target 
     0      0      0        0      0      0        0        0      0        0      0      0      0       0 
```

```
sum(complete.cases(data))
```

```
1025
```

```
dim(data)
```

```
1025   14
```

There are no missing values.

Feature selection

In this section, I am going to choose which can help me to fit the Machine that make the learn "reproduce" between the people who have heart disease and those who do not have.

```
library(FSelector) ## information.gain, cutoff.k
library(Boruta) ## Boruta, getSelectedAttributes
library(randomForest) ## randomForest, importance
```

Method 1 – Two important Feature Selection and Revision Forest Classification

In this method, I will use **Random Forest** algorithm to select if the person has heart disease. After that, I will use "feature importance" to check the confidence of what feature is the problem.

```
set.seed(111)
boruta.train <- Boruta(target~., data = data, doTrace = 2)
print(boruta.train)
```

```
Boruta performed 99 iterations in 1.1 hours.
 13 attributes confirmed important: age cp ca chol exang oldpeak sex slope thal thalach trestbps ...
 1 attributes confirmed unimportant: fbs
 1 tentative attributes left: restecg
```

```
boruta.train$timeTaken
```

```
Time difference of 1.1 hours
```

```
fix(boruta.train)
getSelectedAttributes(boruta.train, withTentative = F)
boruta.df <- attStats(boruta.train)
class(boruta.df)
```

```
[1] "data.frame"
```

```
boruta.df
```

```
            meanImp medianImp ...
age       6.0000000 6.0000000 ...
sex       4.0000000 4.0000000 ...
cp        8.0000000 8.0000000 ...
trestbps  2.0000000 2.0000000 ...
chol      2.0000000 2.0000000 ...
fbs       0.0000000 0.0000000 ...
restecg   1.0000000 1.0000000 ...
thalach   5.0000000 5.0000000 ...
exang     3.0000000 3.0000000 ...
oldpeak   7.0000000 7.0000000 ...
slope     3.0000000 3.0000000 ...
ca        7.0000000 7.0000000 ...
thal      7.0000000 7.0000000 ...
target    0.0000000 0.0000000 ...
  meanRoughFix ...
age       6.0000 ...
sex       4.0000 ...
cp        8.0000 ...
trestbps  2.0000 ...
chol      2.0000 ...
fbs       0.0000 ...
restecg   1.0000 ...
thalach   5.0000 ...
exang     3.0000 ...
oldpeak   7.0000 ...
slope     3.0000 ...
ca        7.0000 ...
thal      7.0000 ...
target    0.0000 ...
  medianRoughFix ...
age       6.0000 ...
sex       4.0000 ...
cp        8.0000 ...
trestbps  2.0000 ...
chol      2.0000 ...
fbs       0.0000 ...
restecg   1.0000 ...
thalach   5.0000 ...
exang     3.0000 ...
oldpeak   7.0000 ...
slope     3.0000 ...
ca        7.0000 ...
thal      7.0000 ...
target    0.0000 ...
  hit ...
age   99 ...
sex   99 ...
cp    99 ...
trestbps 99 ...
chol  99 ...
fbs    0 ...
restecg 33 ...
thalach  99 ...
exang    99 ...
oldpeak  99 ...
slope    99 ...
ca       99 ...
thal     99 ...
target   99 ...
  decision
age  Confirmed
sex  Confirmed
cp   Confirmed
trestbps Confirmed
chol  Confirmed
fbs   Rejected
restecg Tentative
thalach  Confirmed
exang    Confirmed
oldpeak  Confirmed
slope    Confirmed
ca       Confirmed
thal     Confirmed
target   Confirmed
attr(,"maxRuns")
[1] 99
attr(,"times")
$boruta
[1] "2020-10-19 11:11:11 JST" "2020-10-19 12:12:12 JST"
$finalTerm
[1] "2020-10-19 12:12:12 JST"
$timeTaken
[1] "1.1 hours"
$formula
target ~ age + sex + cp + trestbps + chol + fbs + restecg + thalach +
    exang + oldpeak + slope + ca + thal
attr(,"class")
[1] "formula"
$pValue
[1] 0.01
$mustTerm
[1] 0
$extendedFormula
NULL
$maxBuffer
NULL
$maxRuns
[1] 99
attr(,"class")
[1] "Boruta"
```

```
plot(boruta.train, xlab = "", xaxt = "n")
lz<-lapply(1:ncol(boruta.train$ImpHistory),function(i)
boruta.train$ImpHistory[is.finite(boruta.train$ImpHistory[,i]),i])
names(lz) <- colnames(boruta.train$ImpHistory)
Labels <- sort(sapply(lz,median))
axis(3,labels=Labels,at=1:ncol(boruta.train$ImpHistory),cex.axis=0.7)
```



```
plot(boruta.train, xlab = "", xaxt = "n")
lz<-lapply(1:ncol(boruta.train$ImpHistory),function(i)
boruta.train$ImpHistory[is.finite(boruta.train$ImpHistory[,i]),i])
names(lz) <- colnames(boruta.train$ImpHistory)
Labels <- sort(sapply(lz,median))
axis(3,labels=Labels,at=1:ncol(boruta.train$ImpHistory),cex.axis=0.7)
```

```
In [28]: plt.plot(Tree)
         plt.xlabel('Tree')
         plt.ylabel('Accuracy')
         plt.title('Accuracy vs Tree')
         plt.show()
         plt.savefig('Accuracy vs Tree.jpg')
```

Text(0,0,'Accuracy vs Tree')

So we had done a few stage between 1 to 31 it shows that a perfect accuracy. Since I want to avoid overfitting I will use a 5 fold and between 6 the case, I had Tree perfect accuracy

```
In [29]: X_train, X_test, y_train, y_test = train_test_split(X, y, 5)
         clf = DecisionTreeClassifier(max_leaf_nodes=6, random_state=0)
         clf.fit(X_train, y_train)
         clf.score(X_test, y_test)
```

0.9333333333333333

Obtain a perfect accuracy

Now let's see how the tree looks like

```
In [30]: dot_data = StringIO()
         export_graphviz(clf, out_file=dot_data,
                         filled=True,
                         rounded=True,
                         special_characters=True,
                         feature_names = X.columns,
                         class_names=['0','1'])
         graph = pydotplus.graph_from_dot_data(dot_data.getvalue())
         Image(graph.create_png())
```

Out[30]:

Age
≤ 0.000
gini = 0.500
samples = 150
value = [75, 75]
class = 0
Age
≤ -0.000
gini = 0.500
samples = 75
value = [75, 0]
class = 0
Age
≤ 0.000
gini = 0.500
samples = 75
value = [0, 75]
class = 1