Decision Trees

Alex Weirth

4/9/2023

```
library(tidyverse)
library(caret)
library(rpart)
library(rpart.plot)
library(ipred)
library(xgboost)
library(randomForest)
```

The data:

Working again with the heart disease classification data set that I previously cleaned.

```
heart <- read.csv("/Users/alexweirth/Downloads/heart_clean.csv")
str(heart)
```

```
## 'data.frame':    918 obs. of  12 variables:
##  $ Age           : int  40 49 37 48 54 39 45 54 37 48 ...
##  $ Sex           : chr  "M" "F" "M" "F" ...
##  $ ChestPainType : chr  "ATA" "NAP" "ATA" "ASY" ...
##  $ RestingBP     : int  140 160 130 138 150 120 130 110 140 120 ...
##  $ Cholesterol   : int  289 180 283 214 195 339 237 208 207 284 ...
##  $ FastingBS     : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ RestingECG    : chr  "Normal" "Normal" "ST" "Normal" ...
##  $ MaxHR         : int  172 156 98 108 122 170 170 142 130 120 ...
##  $ ExerciseAngina: int  0 0 0 1 0 0 0 0 1 0 ...
##  $ Oldpeak       : num  0 1 0 1.5 0 0 0 0 1.5 0 ...
##  $ ST_Slope      : chr  "Up" "Flat" "Up" "Flat" ...
##  $ HeartDisease  : int  0 1 0 1 0 0 0 0 1 0 ...
```

Trees

The outcome variable HeartDisease has to be encoded as a factor.

```
heart$HeartDisease <- as.factor(heart$HeartDisease)
str(heart)
```

```
## 'data.frame':    918 obs. of  12 variables:
```

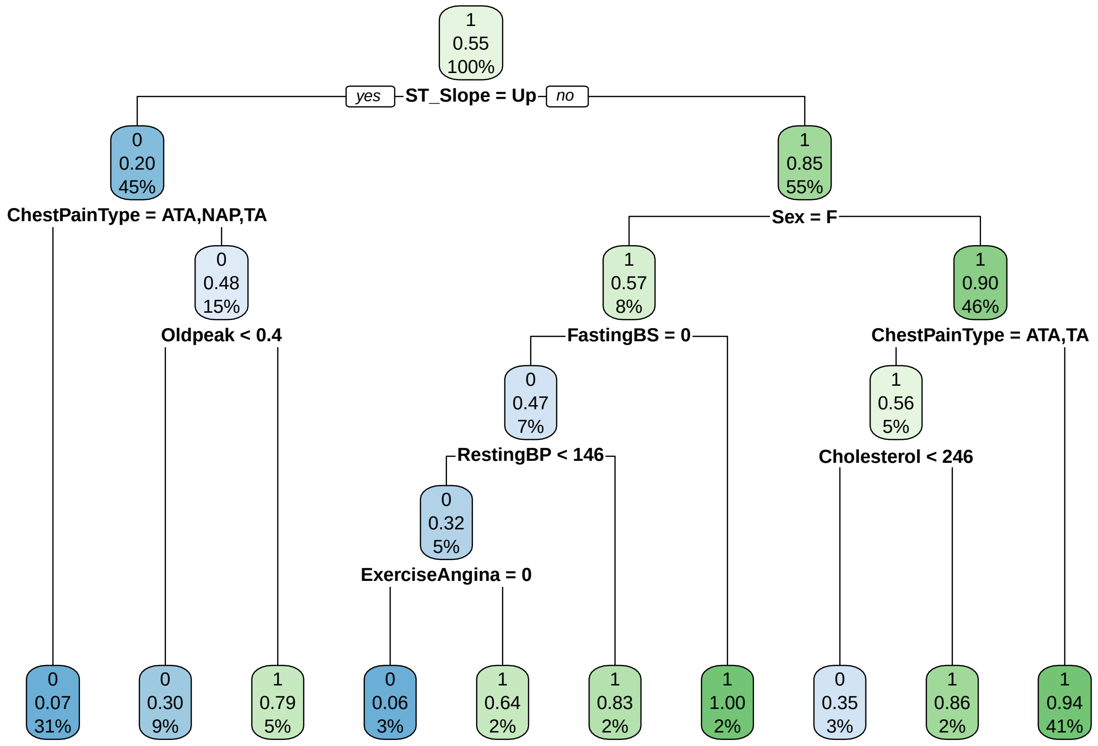
```
##  $ Age           : int  40 49 37 48 54 39 45 54 37 48 ...
##  $ Sex           : chr  "M" "F" "M" "F" ...
##  $ ChestPainType : chr  "ATA" "NAP" "ATA" "ASY" ...
##  $ RestingBP     : int  140 160 130 138 150 120 130 110 140 120 ...
##  $ Cholesterol   : int  289 180 283 214 195 339 237 208 207 284 ...
##  $ FastingBS     : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ RestingECG    : chr  "Normal" "Normal" "ST" "Normal" ...
##  $ MaxHR         : int  172 156 98 108 122 170 170 142 130 120 ...
##  $ ExerciseAngina: int  0 0 0 1 0 0 0 0 1 0 ...
##  $ Oldpeak       : num  0 1 0 1.5 0 0 0 0 1.5 0 ...
##  $ ST_Slope      : chr  "Up" "Flat" "Up" "Flat" ...
##  $ HeartDisease  : Factor w/ 2 levels "0","1": 1 2 1 2 1 1 1 1 2 1 ...
```

Splitting into stratified training/testing using caret package

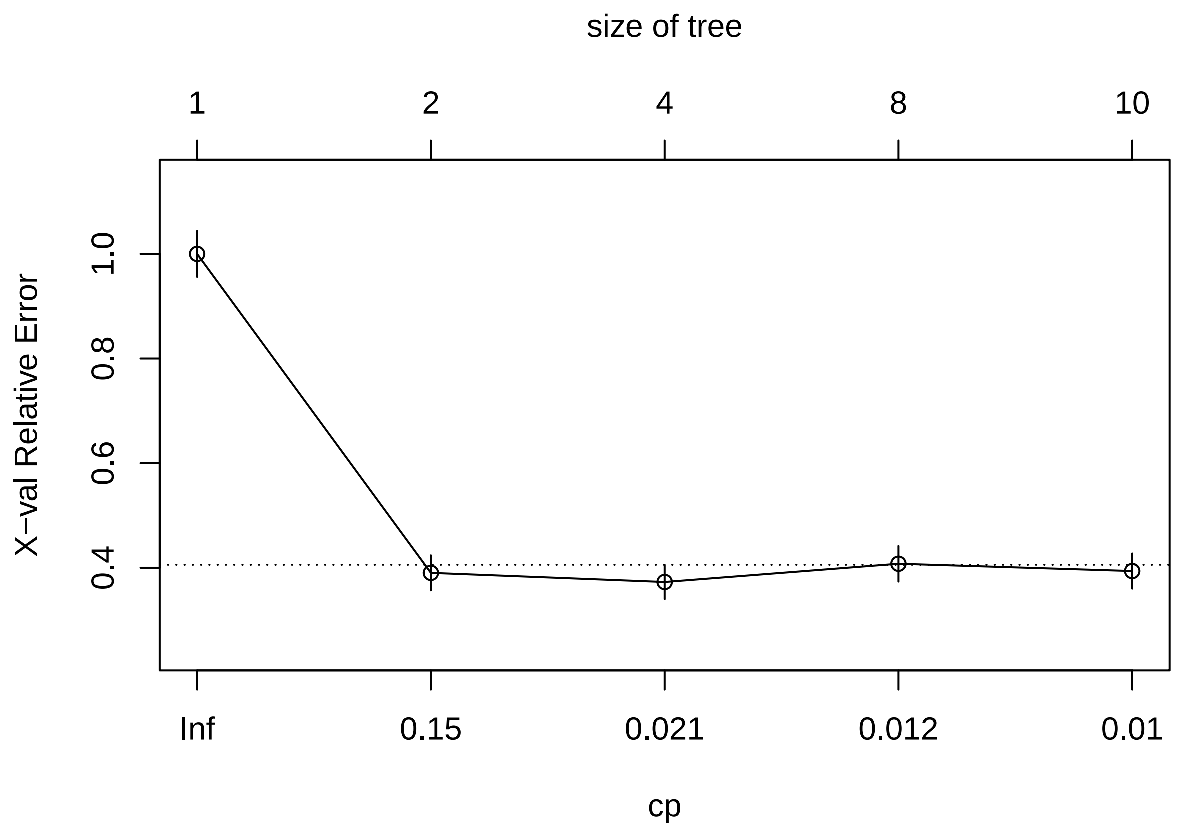
```r
set.seed(3)
caretSamp <- createDataPartition(heart$HeartDisease,
                                       p = 0.7,
                                       list = FALSE)

heart_train <- heart[caretSamp, ]
heart_test<- heart[-caretSamp, ]
```

Fitting Classification Tree:

```r
set.seed(3)

classTree<- rpart(HeartDisease ~., data = heart_train, method = "class")

## Plot Tree
rpart.plot(classTree)
```



```
### Plot CP
plotcp(classTree)
```



```r
printcp(classTree)
```

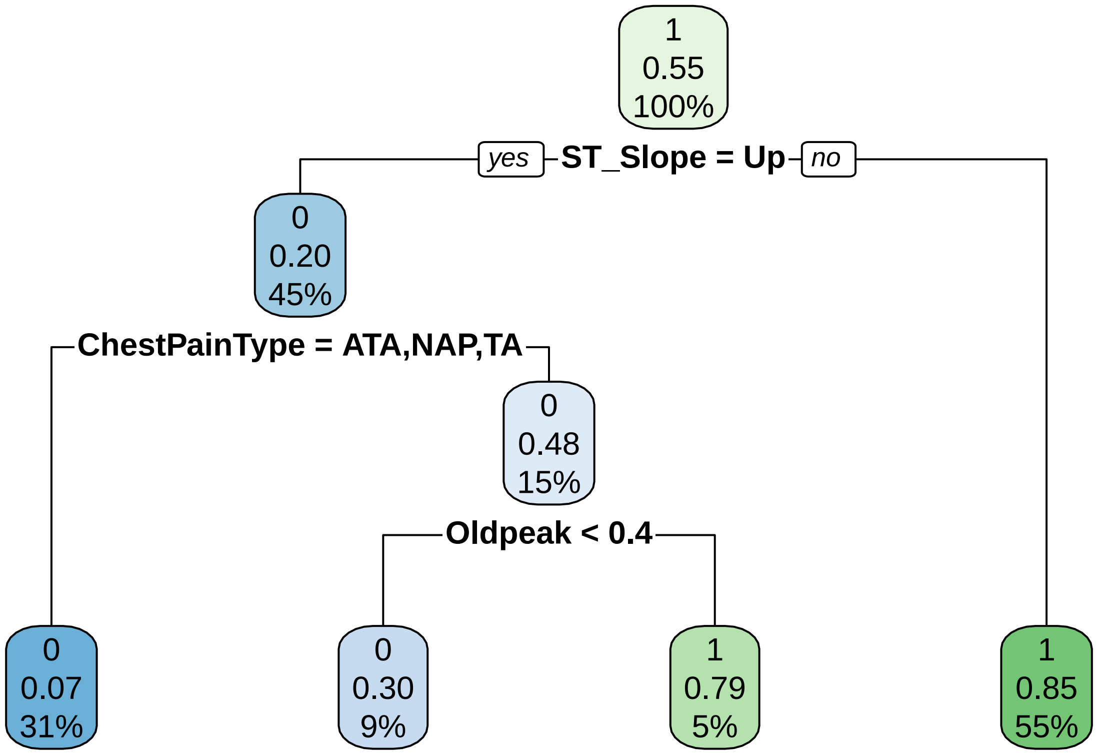
```
##
## Classification tree:
## rpart(formula = HeartDisease ~ ., data = heart_train, method = "class")
##
## Variables actually used in tree construction:
## [1] ChestPainType  Cholesterol    ExerciseAngina FastingBS      Oldpeak
## [6] RestingBP      Sex            ST_Slope
##
## Root node error: 287/643 = 0.44635
##
## n= 643
##
##          CP nsplit rel error  xerror     xstd
## 1 0.609756      0   1.00000 1.00000 0.043922
## 2 0.034843      1   0.39024 0.39024 0.033510
## 3 0.012776      3   0.32056 0.37282 0.032907
## 4 0.010453      7   0.26829 0.40767 0.034088
## 5 0.010000      9   0.24739 0.39373 0.033627
```

```r
## Best CP
minCP<-classTree$cptable[which.min(classTree$cptable[,"xerror"]),"CP"]
minCP
```

```
## [1] 0.01277584
```

Pruning the tree with CP that minimizes error

```r
prune_classTree <- prune(classTree, cp = minCP )
rpart.plot(prune_classTree )
```



Prediction

```r
## Predict Function
predTree1<-predict(prune_classTree, heart_test, type = "class")

## Confusion Matrix
cmTree1<-table(heart_test$HeartDisease, predTree1)
cmTree1
```

```
##    predTree1
##       0   1
##   0  79  44
##   1  11 141
```

```r
## Correct Rate
mean(heart_test$HeartDisease == predTree1)
```

```
## [1] 0.8
```

The pruned tree had an 80% correct rate. Also, false positives were much less common than the models false negatives.

Tree aggregation using bagging

```
set.seed(3)
heartBag <- bagging(HeartDisease ~ .,
                    data = heart_train,
                    nbagg = 150,
                    coob = TRUE,
                    control = rpart.control(minsplit = 2, cp = 0))

## PREDICT
predBag<-predict(heartBag, heart_test, type="class")

## CONFUSION MATRIX & CORRECT RATE
cmBag<-table(heart_test$HeartDisease, predBag)
cmBag
```

```
##    predBag
##       0   1
##   0  93  30
##   1  15 137
```

```
mean(heart_test$HeartDisease == predBag)
```

```
## [1] 0.8363636
```

Aggreagation using bagging increased my correct rate from 80% to 83.6%

Random Forest using Caret Library

```
set.seed(3)
caretRF <- train(HeartDisease ~.,
               data = heart_train,
               method = "rf",
               trControl = trainControl("cv", number = 10),
               importance = TRUE
)
```

Prediction

```
predCaretRF <- caretRF %>% predict(heart_test)

## Confusion Matrix
table(predCaretRF, heart_test$HeartDisease)
```

```
##
## predCaretRF   0   1
##           0  95  12
##           1  28 140
```

```
## Correct Rate
mean(predCaretRF == heart_test$HeartDisease)
```

```
## [1] 0.8545455
```

Correct rate is the best so far at 85.4%

Boosting using the Caret library

```
caretBoost <- train(HeartDisease ~.,
               data = heart_train,
               method="xgbTree",
               trControl = trainControl("cv", number = 10),
               verbosity = 0)
```

Prediction

```
predCaretBoost <- caretBoost %>% predict(heart_test)

## Confusion Matrix
table(predCaretBoost, heart_test$HeartDisease)
```

```
##
## predCaretBoost   0   1
##              0  99  14
##              1  24 138
```

```
## Correct Rate
mean(predCaretBoost == heart_test$HeartDisease)
```

```
## [1] 0.8618182
```

Boosting was able to achieve the best performance overall on the testing data with a 86.1% correct rate.