Classification Trees

Ariel Romero

library(tidyverse)

## -- Attaching packages -------------

## v ggplot2 3.1.0 v purrr 0.3.2   
## v tibble 2.1.1 v dplyr 0.8.0.1  
## v tidyr 0.8.3 v stringr 1.4.0   
## v readr 1.3.1 v forcats 0.4.0

## -- Conflicts ----------------------  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()

library(caret)

## Loading required package: lattice

##   
## Attaching package: 'caret'

## The following object is masked from 'package:purrr':  
##   
## lift

library(rpart)  
library(rattle)

## Rattle: A free graphical interface for data science with R.  
## Version 5.2.0 Copyright (c) 2006-2018 Togaware Pty Ltd.  
## Type 'rattle()' to shake, rattle, and roll your data.

library(RColorBrewer)

parole <- read\_csv("parole (1).csv")

## Parsed with column specification:  
## cols(  
## male = col\_double(),  
## race = col\_double(),  
## age = col\_double(),  
## state = col\_double(),  
## time.served = col\_double(),  
## max.sentence = col\_double(),  
## multiple.offenses = col\_double(),  
## crime = col\_double(),  
## violator = col\_double()  
## )

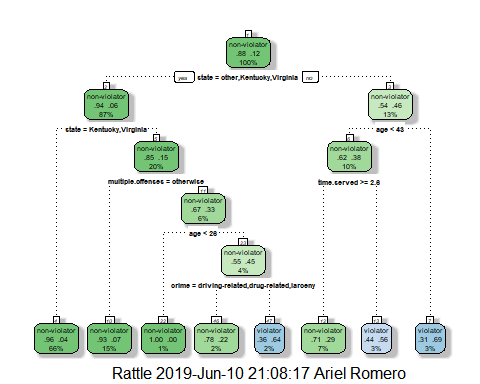
Blood <- read\_csv("Blood.csv")

## Parsed with column specification:  
## cols(  
## Mnths\_Since\_Last = col\_double(),  
## TotalDonations = col\_double(),  
## Total\_Donated = col\_double(),  
## Mnths\_Since\_First = col\_double(),  
## DonatedMarch = col\_double()  
## )

parole = parole %>% mutate(male = as\_factor(as.character(male))) %>%  
mutate(male = fct\_recode(male,  
"female" = "0",  
"male" = "1")) %>% mutate(race = as\_factor(as.character(race))) %>%  
mutate(race = fct\_recode(race,  
"white" = "1",  
"other" = "2")) %>% mutate(state = as\_factor(as.character(state))) %>%  
mutate(state = fct\_recode(state,  
"Kentucky" = "2",  
"Louisiana" = "3",  
"Virginia" = "4",  
"other" = "1")) %>% mutate(crime = as\_factor(as.character(crime))) %>%  
mutate(crime = fct\_recode(crime,  
"larceny" = "2",  
"drug-related" = "3",  
"driving-related" = "4",  
"other" = "1")) %>% mutate(multiple.offenses = as\_factor(as.character(multiple.offenses))) %>%  
mutate(multiple.offenses = fct\_recode(multiple.offenses,  
"otherwise" = "0",  
"multiple offenses" = "1")) %>% mutate(violator = as\_factor(as.character(violator))) %>%  
mutate(violator = fct\_recode(violator,  
"non-violator" = "0",  
"violator" = "1"))

set.seed(12345)  
train.rows = createDataPartition(y = parole$violator, p=0.7, list = FALSE)  
train = parole[train.rows,]   
test = parole[-train.rows,]

tree1 = rpart(violator ~., train, method = "class")  
fancyRpartPlot(tree1)

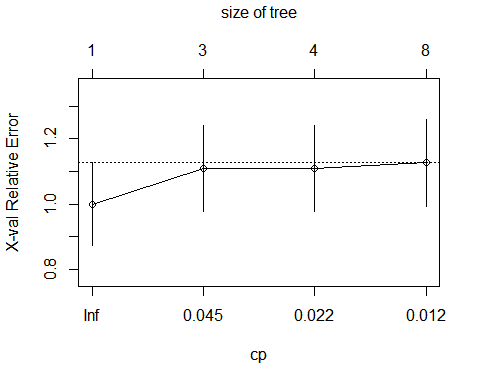


A 40 year old parolee from Louisiana who served a 5 year prison sentence would be classified as a non-violator. The first prompt was if the state equaled other, Kentucky, or Virignina, the answer is no. The second prompt asked if their age was less than 40, the answer is yes. The final prompt asked if their time served was greater than 2.6, the answer is also yes which classified him as a non-violator.

printcp(tree1)

##   
## Classification tree:  
## rpart(formula = violator ~ ., data = train, method = "class")  
##   
## Variables actually used in tree construction:  
## [1] age crime multiple.offenses state   
## [5] time.served   
##   
## Root node error: 55/473 = 0.11628  
##   
## n= 473   
##   
## CP nsplit rel error xerror xstd  
## 1 0.054545 0 1.00000 1.0000 0.12676  
## 2 0.036364 2 0.89091 1.1091 0.13253  
## 3 0.013636 3 0.85455 1.1091 0.13253  
## 4 0.010000 7 0.80000 1.1273 0.13345

plotcp(tree1)



0.054545 is the cp value that should be selected.

tree2 = prune(tree1,cp= tree1$cptable[which.min(tree1$cptable[,"xerror"]),"CP"])  
  
printcp(tree2)

##   
## Classification tree:  
## rpart(formula = violator ~ ., data = train, method = "class")  
##   
## Variables actually used in tree construction:  
## character(0)  
##   
## Root node error: 55/473 = 0.11628  
##   
## n= 473   
##   
## CP nsplit rel error xerror xstd  
## 1 0.054545 0 1 1 0.12676

treepred = predict(tree2,train,type = "class")  
head(treepred)

## [1] non-violator non-violator non-violator non-violator non-violator  
## [6] non-violator  
## Levels: non-violator violator

The class “non-violator” is the majority class.

treepredtrain = predict(tree1, train, type="class")  
  
confusionMatrix(treepredtrain, train$violator)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction non-violator violator  
## non-violator 402 28  
## violator 16 27  
##   
## Accuracy : 0.907   
## 95% CI : (0.8771, 0.9316)  
## No Information Rate : 0.8837   
## P-Value [Acc > NIR] : 0.06272   
##   
## Kappa : 0.5   
##   
## Mcnemar's Test P-Value : 0.09725   
##   
## Sensitivity : 0.9617   
## Specificity : 0.4909   
## Pos Pred Value : 0.9349   
## Neg Pred Value : 0.6279   
## Prevalence : 0.8837   
## Detection Rate : 0.8499   
## Detection Prevalence : 0.9091   
## Balanced Accuracy : 0.7263   
##   
## 'Positive' Class : non-violator   
##

The predicition for the train data, from the classification tree have the following characteristics: its accuracy is 90.7%, which I believe is good, the sensitivity is 96.17% and the specificity is 49.09%.

treepredtest = predict(tree1, test, type="class")  
  
confusionMatrix(treepredtest, test$violator)

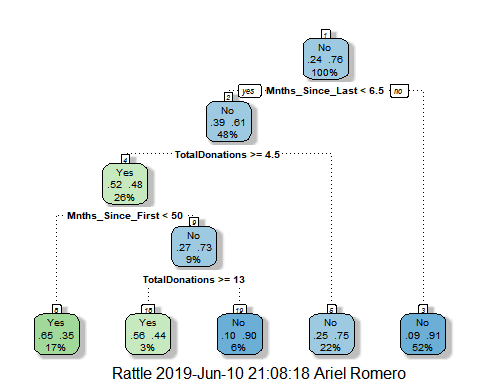
## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction non-violator violator  
## non-violator 170 19  
## violator 9 4  
##   
## Accuracy : 0.8614   
## 95% CI : (0.8059, 0.9059)  
## No Information Rate : 0.8861   
## P-Value [Acc > NIR] : 0.88631   
##   
## Kappa : 0.1525   
##   
## Mcnemar's Test P-Value : 0.08897   
##   
## Sensitivity : 0.9497   
## Specificity : 0.1739   
## Pos Pred Value : 0.8995   
## Neg Pred Value : 0.3077   
## Prevalence : 0.8861   
## Detection Rate : 0.8416   
## Detection Prevalence : 0.9356   
## Balanced Accuracy : 0.5618   
##   
## 'Positive' Class : non-violator   
##

On the testing data, the accuracy goes down a little to 86.14%, sensitivity stays pretty consistent at 94.97%, specifcity goes down considerably to 17.39%.

Blood = Blood %>% mutate(DonatedMarch=as\_factor(as.character(DonatedMarch))) %>% mutate(DonatedMarch = fct\_recode(DonatedMarch, "No" = "0", "Yes" = "1"))

set.seed(1234)  
train.rows = createDataPartition(y = Blood$DonatedMarch, p=0.7, list = FALSE)  
train2 = Blood[train.rows,]   
test2 = Blood[-train.rows,]

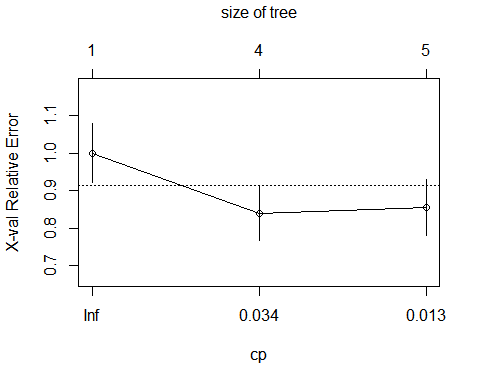
BloodClassTree = rpart(DonatedMarch ~., train2, method = "class")  
  
fancyRpartPlot(BloodClassTree)



printcp(BloodClassTree)

##   
## Classification tree:  
## rpart(formula = DonatedMarch ~ ., data = train2, method = "class")  
##   
## Variables actually used in tree construction:  
## [1] Mnths\_Since\_First Mnths\_Since\_Last TotalDonations   
##   
## Root node error: 125/524 = 0.23855  
##   
## n= 524   
##   
## CP nsplit rel error xerror xstd  
## 1 0.072 0 1.000 1.000 0.078049  
## 2 0.016 3 0.784 0.840 0.073304  
## 3 0.010 4 0.768 0.856 0.073822

plotcp(BloodClassTree)



BloodClassTree2= prune(BloodClassTree,cp= BloodClassTree$cptable[which.min(tree1$cptable[,"xerror"]),"CP"])  
  
printcp(BloodClassTree2)

##   
## Classification tree:  
## rpart(formula = DonatedMarch ~ ., data = train2, method = "class")  
##   
## Variables actually used in tree construction:  
## character(0)  
##   
## Root node error: 125/524 = 0.23855  
##   
## n= 524   
##   
## CP nsplit rel error xerror xstd  
## 1 0.072 0 1 1 0.078049

treepredtrain2 = predict(BloodClassTree2, train2, type="class")  
  
head(treepredtrain2)

## [1] No No No No No No  
## Levels: Yes No

confusionMatrix(treepredtrain2, train2$DonatedMarch)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 0 0  
## No 125 399  
##   
## Accuracy : 0.7615   
## 95% CI : (0.7226, 0.7973)  
## No Information Rate : 0.7615   
## P-Value [Acc > NIR] : 0.524   
##   
## Kappa : 0   
##   
## Mcnemar's Test P-Value : <2e-16   
##   
## Sensitivity : 0.0000   
## Specificity : 1.0000   
## Pos Pred Value : NaN   
## Neg Pred Value : 0.7615   
## Prevalence : 0.2385   
## Detection Rate : 0.0000   
## Detection Prevalence : 0.0000   
## Balanced Accuracy : 0.5000   
##   
## 'Positive' Class : Yes   
##

treepredtest2 = predict(BloodClassTree2, test2, type="class")  
  
head(treepredtest2)

## [1] No No No No No No  
## Levels: Yes No

confusionMatrix(treepredtest2, test2$DonatedMarch)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 0 0  
## No 53 171  
##   
## Accuracy : 0.7634   
## 95% CI : (0.7022, 0.8175)  
## No Information Rate : 0.7634   
## P-Value [Acc > NIR] : 0.5368   
##   
## Kappa : 0   
##   
## Mcnemar's Test P-Value : 9.148e-13   
##   
## Sensitivity : 0.0000   
## Specificity : 1.0000   
## Pos Pred Value : NaN   
## Neg Pred Value : 0.7634   
## Prevalence : 0.2366   
## Detection Rate : 0.0000   
## Detection Prevalence : 0.0000   
## Balanced Accuracy : 0.5000   
##   
## 'Positive' Class : Yes   
##

The accuracy on both predictions from the classification trees are around 76% which I would deem good. One thing I am concerned about is that on both predictions, the sensitivity is 0 and the specificty is 1. The accuaracy is significant against the naive rate in both models.