Classification Trees

# Classification Trees

## BAN502

### Danielle Harris

library(tidyverse)

## ── Attaching packages ────────────────────────────────────────── tidyverse 1.2.1 ──

## ✔ ggplot2 3.1.0 ✔ purrr 0.2.5  
## ✔ tibble 1.4.2 ✔ dplyr 0.7.7  
## ✔ tidyr 0.8.2 ✔ stringr 1.3.1  
## ✔ readr 1.1.1 ✔ forcats 0.3.0

## ── Conflicts ───────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ 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.csv")

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

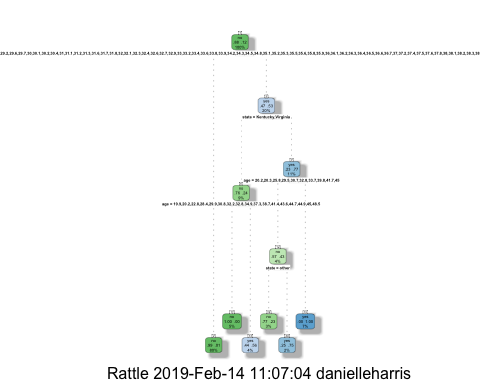
parole = parole %>% mutate(male= as.factor(male)) %>%   
 mutate (male= fct\_recode(male, "female"="0", "male"= "1")) %>%   
mutate(race= as.factor(race)) %>% mutate(race=fct\_recode(race, "white" = "1", "other"= "2")) %>%   
mutate(age= as.factor(age)) %>%  
mutate(state= as.factor(state)) %>% mutate(state= fct\_recode(state, "Kentucky"="2", "Louisiana"= "3", "Virginia"="4", "other"= "1")) %>%  
mutate(crime= as.factor(crime)) %>%  
mutate(crime= fct\_recode(crime, "larceny"= "2", "drug-related"="3", "driving-related"="4", "other"= "1")) %>%  
mutate(multiple.offenses= as.factor(multiple.offenses)) %>% mutate(multiple.offenses=fct\_recode(multiple.offenses, "yes"="1", "no"= "0")) %>%  
mutate(violator=as.factor(violator)) %>% mutate (violator= fct\_recode(violator, "yes"= "1", "no"="0"))

# Task 1

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

# Task 2

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



# Task 3

## Step 1: The age of the parolee is 40, which is on the first step of the classification tree. Because we would say “yes,” we move to the next blue step that says “yes.”

## Step 2: The state of the parolee we are trying to predict is neither Kentucky nor Virginia, so we would say “no,” moving us to the next green step.

## The age of the parolee is not an option on this step of the classification tree, so we would say “no,” moving us to thhe next green step of the classification tree.

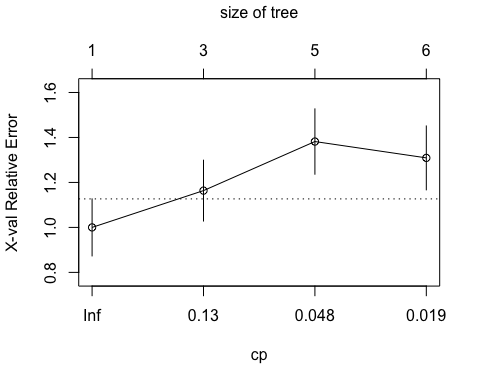
## Finally, the last step of this path on the classification tree leads to 5% chance that the parolee is predcicted to violate his parole.

# Task 4

printcp(tree1)

##   
## Classification tree:  
## rpart(formula = violator ~ ., data = train, method = "class")  
##   
## Variables actually used in tree construction:  
## [1] age state  
##   
## Root node error: 55/473 = 0.11628  
##   
## n= 473   
##   
## CP nsplit rel error xerror xstd  
## 1 0.254545 0 1.00000 1.0000 0.12676  
## 2 0.063636 2 0.49091 1.1636 0.13526  
## 3 0.036364 4 0.36364 1.3818 0.14521  
## 4 0.010000 5 0.32727 1.3091 0.14205

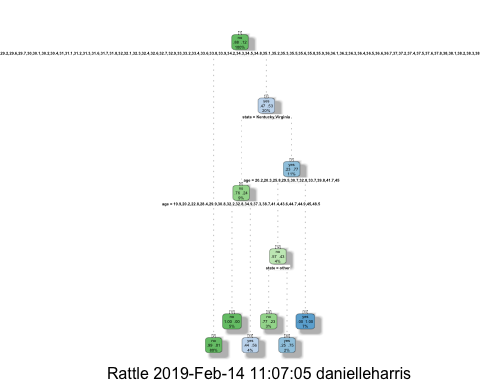
plotcp(tree1)



## We would use a complexity parameter of 0.01.

# Task 5

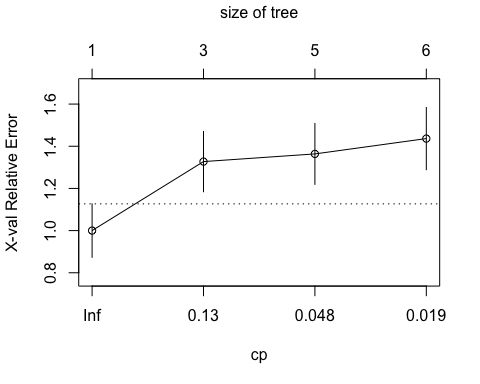
tree2 = rpart(violator ~., train, cp=0.01, method="class")  
fancyRpartPlot(tree2)



printcp(tree2)

##   
## Classification tree:  
## rpart(formula = violator ~ ., data = train, method = "class",   
## cp = 0.01)  
##   
## Variables actually used in tree construction:  
## [1] age state  
##   
## Root node error: 55/473 = 0.11628  
##   
## n= 473   
##   
## CP nsplit rel error xerror xstd  
## 1 0.254545 0 1.00000 1.0000 0.12676  
## 2 0.063636 2 0.49091 1.3273 0.14286  
## 3 0.036364 4 0.36364 1.3636 0.14444  
## 4 0.010000 5 0.32727 1.4364 0.14749

plotcp(tree2)



## Age is the majority class in the training set.

# Task 6

treepred = predict(tree1, train, type = "class")  
head(treepred)

## 1 2 3 4 5 6   
## no no no no no no   
## Levels: no yes

confusionMatrix(treepred,train$violator,positive="yes")

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction no yes  
## no 408 8  
## yes 10 47  
##   
## Accuracy : 0.9619   
## 95% CI : (0.9405, 0.9773)  
## No Information Rate : 0.8837   
## P-Value [Acc > NIR] : 1.265e-09   
##   
## Kappa : 0.8177   
## Mcnemar's Test P-Value : 0.8137   
##   
## Sensitivity : 0.85455   
## Specificity : 0.97608   
## Pos Pred Value : 0.82456   
## Neg Pred Value : 0.98077   
## Prevalence : 0.11628   
## Detection Rate : 0.09937   
## Detection Prevalence : 0.12051   
## Balanced Accuracy : 0.91531   
##   
## 'Positive' Class : yes   
##

treepred\_test = predict(tree1, test, type = "class")  
head(treepred\_test)

## 1 2 3 4 5 6   
## no yes no no no no   
## Levels: no yes

confusionMatrix(treepred\_test,test$violator,positive="yes")

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction no yes  
## no 166 22  
## yes 13 1  
##   
## Accuracy : 0.8267   
## 95% CI : (0.7674, 0.8762)  
## No Information Rate : 0.8861   
## P-Value [Acc > NIR] : 0.9956   
##   
## Kappa : -0.0351   
## Mcnemar's Test P-Value : 0.1763   
##   
## Sensitivity : 0.04348   
## Specificity : 0.92737   
## Pos Pred Value : 0.07143   
## Neg Pred Value : 0.88298   
## Prevalence : 0.11386   
## Detection Rate : 0.00495   
## Detection Prevalence : 0.06931   
## Balanced Accuracy : 0.48543   
##   
## 'Positive' Class : yes   
##

## The accuracy of the testing model decreases and the p-values of both the training and testing set are greater than 0.05. Therefore, the quality of the model is not as strong as it could be.

# Task 8

Blood= read\_csv("Blood.csv")

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

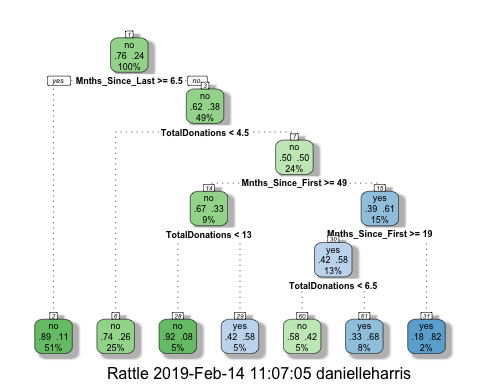
Blood = Blood %>% mutate(DonatedMarch= as.factor(DonatedMarch)) %>%   
 mutate (DonatedMarch= fct\_recode(DonatedMarch, "no"="0", "yes"= "1"))

# Task 9

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

# Task 10

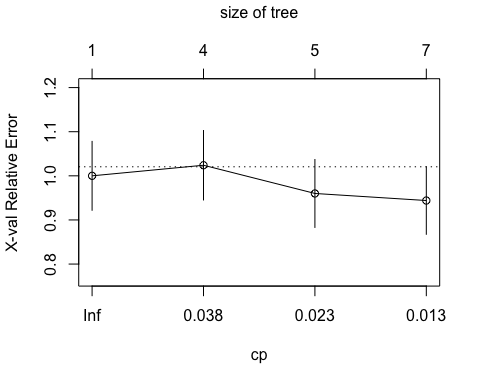
tree3 = rpart(DonatedMarch ~ ., train2, method="class")  
fancyRpartPlot(tree3)



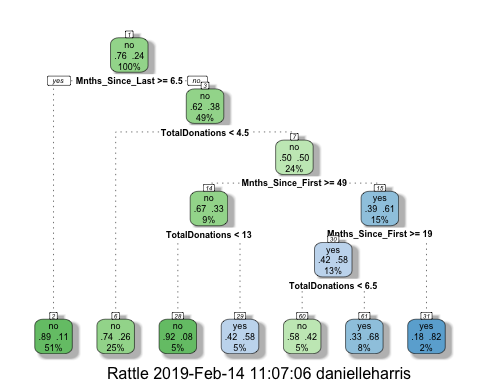
printcp(tree3)

##   
## 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 0.045333 0 1.000 1.000 0.078049  
## 2 0.032000 3 0.864 1.024 0.078682  
## 3 0.016000 4 0.832 0.960 0.076949  
## 4 0.010000 6 0.800 0.944 0.076494

plotcp(tree3)



tree4 = rpart(DonatedMarch ~ ., train2, cp=0.01, method="class")  
fancyRpartPlot(tree4)



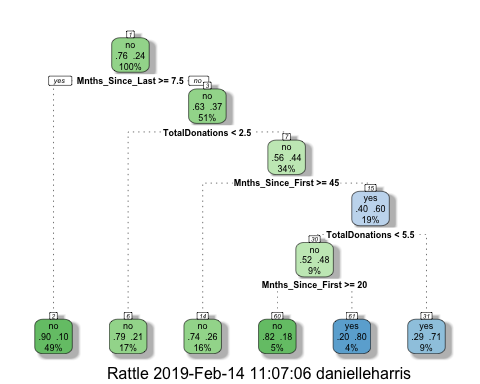
treepred2 = predict(tree4, train2, type = "class")  
head(treepred2)

## 1 2 3 4 5 6   
## yes yes yes no yes yes   
## Levels: no yes

confusionMatrix(treepred2,train2$DonatedMarch,positive="yes")

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction no yes  
## no 374 75  
## yes 25 50  
##   
## Accuracy : 0.8092   
## 95% CI : (0.7729, 0.8419)  
## No Information Rate : 0.7615   
## P-Value [Acc > NIR] : 0.005169   
##   
## Kappa : 0.3911   
## Mcnemar's Test P-Value : 9.584e-07   
##   
## Sensitivity : 0.40000   
## Specificity : 0.93734   
## Pos Pred Value : 0.66667   
## Neg Pred Value : 0.83296   
## Prevalence : 0.23855   
## Detection Rate : 0.09542   
## Detection Prevalence : 0.14313   
## Balanced Accuracy : 0.66867   
##   
## 'Positive' Class : yes   
##

tree5 = rpart(DonatedMarch ~ ., test2, cp=0.01, method="class")  
fancyRpartPlot(tree5)



treepred\_test2 = predict(tree5, test2, type = "class")  
head(treepred\_test2)

## 1 2 3 4 5 6   
## yes yes yes yes yes no   
## Levels: no yes

confusionMatrix(treepred\_test2,test2$DonatedMarch,positive="yes")

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction no yes  
## no 163 30  
## yes 8 23  
##   
## Accuracy : 0.8304   
## 95% CI : (0.7747, 0.8771)  
## No Information Rate : 0.7634   
## P-Value [Acc > NIR] : 0.0094414   
##   
## Kappa : 0.4519   
## Mcnemar's Test P-Value : 0.0006577   
##   
## Sensitivity : 0.4340   
## Specificity : 0.9532   
## Pos Pred Value : 0.7419   
## Neg Pred Value : 0.8446   
## Prevalence : 0.2366   
## Detection Rate : 0.1027   
## Detection Prevalence : 0.1384   
## Balanced Accuracy : 0.6936   
##   
## 'Positive' Class : yes   
##

# The accuracy of the model is 83% and the p-value is less than 0.05, making the model a strong quailty.