# Classification Trees

## Zac Coffey

Libraries

options(tidyverse.quiet = TRUE)  
library("tidyverse")  
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")

Read In Data

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()  
## )

Data frame Conversions

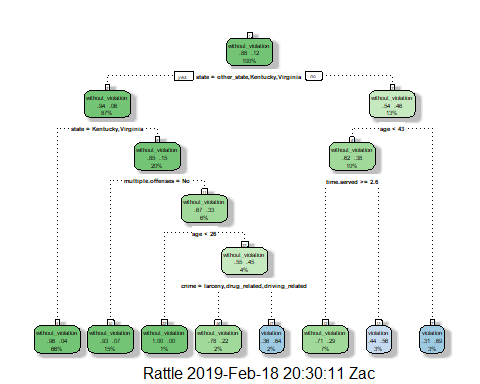
parole = parole %>% mutate(male = as.factor(male)) %>%  
 mutate(male = fct\_recode(male,  
 "female" = "0",  
 "male" = "1"))  
  
parole = parole %>% mutate(race = as.factor(race)) %>%  
 mutate(race = fct\_recode(race,  
 "white" = "1",  
 "otherwise" = "2"))  
  
parole = parole %>% mutate(state = as.factor(state)) %>%  
 mutate(state = fct\_recode(state,  
 "other\_state" = "1",  
   
 "Kentucky" = "2",  
 "Louisiana" = "3",  
 "Virginia" = "4"))  
  
parole = parole %>% mutate(crime = as.factor(crime)) %>%  
 mutate(crime = fct\_recode(crime,  
 "other\_crime" = "1",  
 "larceny" = "2",  
 "drug\_related" = "3",  
 "driving\_related" = "4"))  
  
parole = parole %>% mutate(multiple.offenses = as.factor(multiple.offenses)) %>%  
 mutate(multiple.offenses = fct\_recode(multiple.offenses,  
 "No" = "0",  
 "multiple\_offenses" = "1"))  
  
parole = parole %>% mutate(violator = as.factor(violator)) %>%  
 mutate(violator = fct\_recode(violator,  
 "without\_violation" = "0",  
 "violation" = "1"))

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

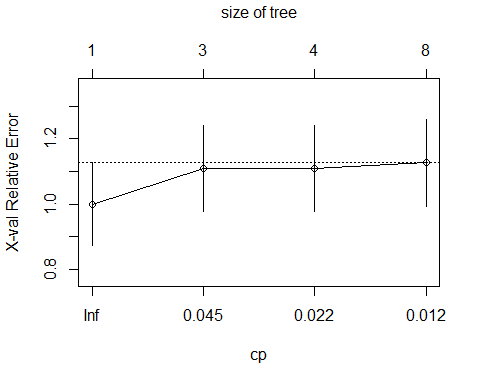
Tree1 = rpart(violator ~., train, method="class")  
fancyRpartPlot(Tree1)



Task 3 #I would classifiy an individual with those characteristics as someone who would not violate thier parole. The walk through of the classification tree. First, I saw that Louisiana wasn’t recongized so I went right to repersent no. Second, I saw that age 40 fit the criteria so I went left to repersent yes. Thrid, I saw time served 5 years fit the criteria so I went left to repersent yes. That brought me to my conclusion.

Task 4

plotcp(Tree1)



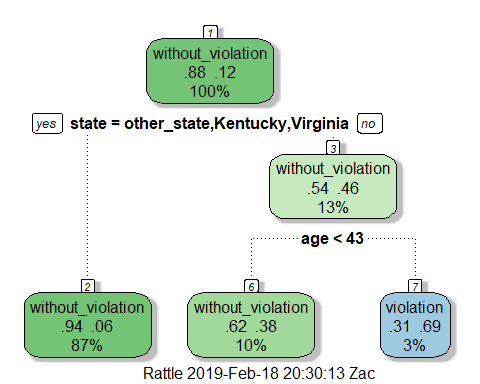
printcp(Tree1)

##   
## 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

#Based on the plotcp I would choose the cp value of 0.045 becasue it appears to have the lowest relative error. Based on the printcp I would choose 0.364 becasue it has the lowest relative error. Even though this relative value is idential to another result I chose the 0.364 because the other result is close to .01 ehich is bad becasue it could lead to overfitting.

Task 5

Tree2 = prune(Tree1, cp = 0.036364)  
fancyRpartPlot(Tree2)



#Based of the of cp value from the printcp(0.036364) I found the majority class. The majority class would be an individual who is from Kentucky, Virginia, or other\_state. This majority class has 87% of the total observations, and is a result of without violation of parole.

Task 6

Train\_Pred = predict(Tree1, train, type = "class")  
  
confusionMatrix(Train\_Pred, train$violator)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction without\_violation violation  
## without\_violation 402 28  
## violation 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 : without\_violation  
##

Task 7

Test\_Pred = predict(Tree1, test, type = "class")  
  
confusionMatrix(Test\_Pred, test$violator)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction without\_violation violation  
## without\_violation 170 19  
## violation 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 : without\_violation  
##

#Our Accuracy is 0.8614 and the No Information Rate is 0.8861. This result is not great, we want Our Accuracy to be higher than the No Information Rate. Also the p-value is 0.88631 meaning that there is not a singificant differnce in this tree.To conclude the quality of the model is low.

Task 8

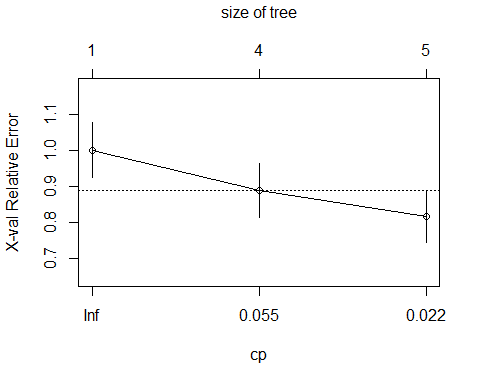
Blood <- read\_csv("Blood.csv")

## 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.rows2 = createDataPartition(y = Blood$DonatedMarch, p=0.7, list = FALSE)   
train2 = Blood[train.rows,]   
test2 = Blood[-train.rows,]  
  
Tree3 = rpart(DonatedMarch ~., train2, method="class")  
  
plotcp(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: 126/473 = 0.26638  
##   
## n= 473   
##   
## CP nsplit rel error xerror xstd  
## 1 0.063492 0 1.00000 1.00000 0.076304  
## 2 0.047619 3 0.80952 0.88889 0.073377  
## 3 0.010000 4 0.76190 0.81746 0.071239

#Based on the plotcp I would choose the cp value of 0.022 becasue it appears to have the lowest relative error. Based on the printcp I would choose 0.0476 becasue it has the lowest relative error.

Task 10

Tree4 = prune(Tree3, cp = 0.047619 )  
  
Train\_Pred2 = predict(Tree4, train2, type = "class")  
confusionMatrix(Train\_Pred2, train2$DonatedMarch)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction No Yes  
## No 312 61  
## Yes 35 65  
##   
## Accuracy : 0.797   
## 95% CI : (0.7579, 0.8324)  
## No Information Rate : 0.7336   
## P-Value [Acc > NIR] : 0.0008251   
##   
## Kappa : 0.4442   
## Mcnemar's Test P-Value : 0.0107244   
##   
## Sensitivity : 0.8991   
## Specificity : 0.5159   
## Pos Pred Value : 0.8365   
## Neg Pred Value : 0.6500   
## Prevalence : 0.7336   
## Detection Rate : 0.6596   
## Detection Prevalence : 0.7886   
## Balanced Accuracy : 0.7075   
##   
## 'Positive' Class : No   
##

Test\_Pred2 = predict(Tree4, test2, type = "class")  
confusionMatrix(Test\_Pred2, test2$DonatedMarch)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction No Yes  
## No 204 34  
## Yes 19 18  
##   
## Accuracy : 0.8073   
## 95% CI : (0.7556, 0.8522)  
## No Information Rate : 0.8109   
## P-Value [Acc > NIR] : 0.59717   
##   
## Kappa : 0.2934   
## Mcnemar's Test P-Value : 0.05447   
##   
## Sensitivity : 0.9148   
## Specificity : 0.3462   
## Pos Pred Value : 0.8571   
## Neg Pred Value : 0.4865   
## Prevalence : 0.8109   
## Detection Rate : 0.7418   
## Detection Prevalence : 0.8655   
## Balanced Accuracy : 0.6305   
##   
## 'Positive' Class : No   
##

#The results of the Train2 prediction. Our Accuracy is 0.797 and the No Information Rate is 0.7336, also the P-Value is signicant. The results of the Test2 prediction. Our Accuracy is 0.8073 and the No Information Rate is 0.8109, also the P-Value is not significant. These results are interesting to me because based on the results of Train2 I would conclude that the quality of the model is high. However based on the results of Test2 I would conclude that the quality of the model is low. I perosnally would conclude that this model's quality is weak because the Test2 results would need to be postive because the testing data proves if the dataset has a high enough quailty to be tested against other datasets.