# Ben Robison

## Mod 4 - Classification trees

library(tidyverse)

## -- Attaching packages ------------------------ tidyverse 1.2.1 --

## 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 --------------------------- tidyverse\_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)

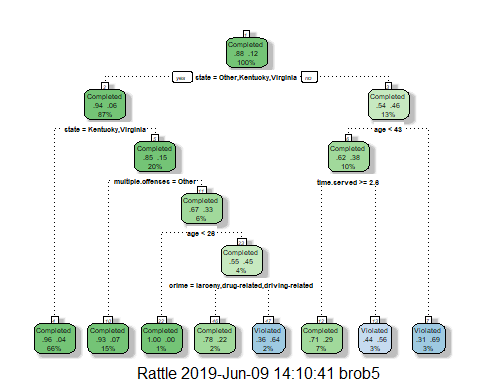
parole <- read\_csv("~/Ban502/Module 3/LogisticRegression/parole.csv")

## 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()  
## )

View(parole)  
parole = parole %>% mutate(male = as\_factor(male)) %>% mutate(male = fct\_recode(male, "male" = "1", "female" = "0")) %>% mutate(race = as.factor(race)) %>% mutate(race = fct\_recode(race, "White" = "1", "Otherwise" = "2")) %>% 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, "multiple offenses" = "1", "Other" = "0")) %>%  
mutate(violator = as\_factor(violator)) %>% mutate(violator = fct\_recode(violator, "Violated" = "1", "Completed" = "0")) %>%  
mutate(state = as\_factor(state)) %>% mutate(state = fct\_recode(state, "Kentucky" = "2", "Lousiana" = "3", "Virginia" = "4", "Other" = "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)

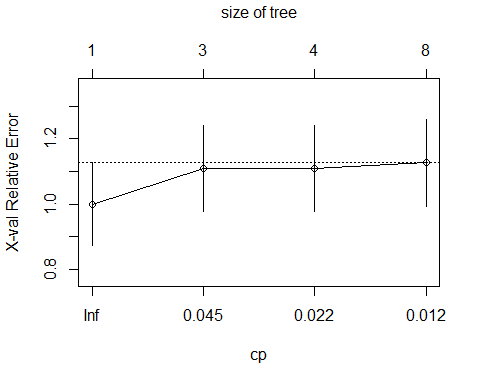


He would be classified as a violator. start to the right as lousiana is the only state not listed, he is less than 43 so go left, but he is serving greater than 2.6 years so go right to violated.

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)

 .054 would be the best CP.

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

Completed is the majority class for violator

treepred = predict(tree1, train, type = "class")  
confusionMatrix(treepred, train$violator, positive = "Completed")

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Completed Violated  
## Completed 402 28  
## Violated 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 : Completed   
##

treepred\_test = predict(tree1, test, type = "class")  
confusionMatrix(treepred\_test, test$violator, positive = "Completed")

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Completed Violated  
## Completed 170 19  
## Violated 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 : Completed   
##

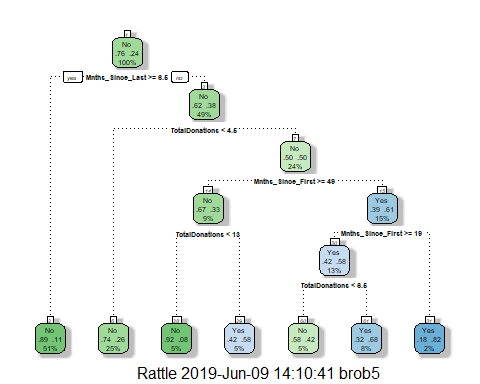
Good model, high accuracy and high spensivity but lower specifity.

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()  
## )

View(Blood)  
Blood <- Blood %>% mutate(DonatedMarch = as\_factor(DonatedMarch)) %>% mutate(DonatedMarch = fct\_recode(DonatedMarch, "Yes" = "1", "No" = "0"))

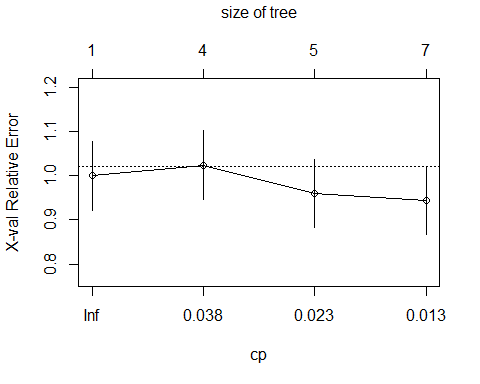
set.seed(1234)  
train.rows2 = createDataPartition(y = Blood$DonatedMarch, p=0.7, list = FALSE)   
train2 = Blood[train.rows2,]   
test2 = Blood[-train.rows2,]  
tree2 = rpart(DonatedMarch~.,train2, method="class")  
fancyRpartPlot(tree2)



printcp(tree2)

##   
## 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(tree2)

 .010 is best CP at lowest xerror 0.944

tree2\_prune = prune(tree2,cp= tree2$cptable[which.min(tree2$cptable[,"xerror"]),"CP"])  
treepred2 = predict(tree2, train2, type = "class")  
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   
##

treepred2\_test = predict(tree2, test2, type = "class")  
confusionMatrix(treepred2\_test, test2$DonatedMarch, positive = "Yes")

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction No Yes  
## No 160 32  
## Yes 11 21  
##   
## Accuracy : 0.808   
## 95% CI : (0.7503, 0.8575)  
## No Information Rate : 0.7634   
## P-Value [Acc > NIR] : 0.065180   
##   
## Kappa : 0.3845   
##   
## Mcnemar's Test P-Value : 0.002289   
##   
## Sensitivity : 0.39623   
## Specificity : 0.93567   
## Pos Pred Value : 0.65625   
## Neg Pred Value : 0.83333   
## Prevalence : 0.23661   
## Detection Rate : 0.09375   
## Detection Prevalence : 0.14286   
## Balanced Accuracy : 0.66595   
##   
## 'Positive' Class : Yes   
##

Moderate model, high accuracy, low sensitivy but high specifity.