library(tidyverse)

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

## ✔ ggplot2 3.2.1 ✔ purrr 0.3.3  
## ✔ tibble 2.1.3 ✔ dplyr 0.8.3  
## ✔ tidyr 1.0.0 ✔ stringr 1.4.0  
## ✔ readr 1.3.1 ✔ forcats 0.4.0

## ── 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 5.3.0 Copyright (c) 2006-2018 Togaware Pty Ltd.  
## Type 'rattle()' to shake, rattle, and roll your data.

library(RColorBrewer)  
library(e1071)

library(readxl)  
parole <- read\_excel("parole.xlsx")  
View(parole)

## Warning in system2("/usr/bin/otool", c("-L", shQuote(DSO)), stdout = TRUE):  
## running command ''/usr/bin/otool' -L '/Library/Frameworks/R.framework/  
## Resources/modules/R\_de.so'' had status 1

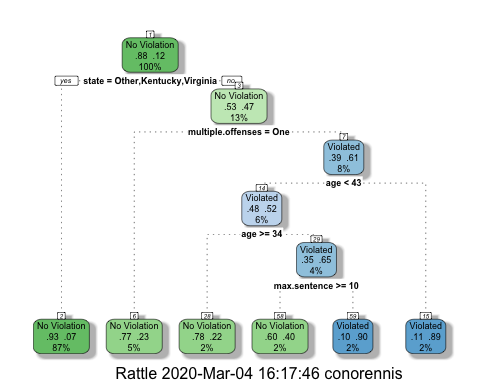
parole = parole %>% mutate(male = as\_factor(as.character(male))) %>%  
 mutate(male = fct\_recode(male,  
 "Male" = "1",  
 "Female" = "0"))  
parole = parole %>% mutate(race = as\_factor(as.character(race))) %>%  
 mutate(race = fct\_recode(race,  
 "White" = "1",  
 "Other" = "2"))  
parole = parole %>% mutate(state = as\_factor(as.character(state))) %>%  
 mutate(state = fct\_recode(state,  
 "Other" = "1",  
 "Kentucky" = "2",  
 "Louisiana" = "3",  
 "Virginia" = "4"))  
parole = parole %>% mutate(multiple.offenses = as\_factor(as.character(multiple.offenses))) %>%  
 mutate(multiple.offenses = fct\_recode(multiple.offenses,  
 "Multiple" = "1",  
 "One" = "0"))  
parole = parole %>% mutate(crime = as\_factor(as.character(crime))) %>%  
 mutate(crime = fct\_recode(crime,  
 "larceny" = "2",  
 "drug-related" = "3",  
 "driving-related" = "4",  
 "Other" = "1"))  
parole = parole %>% mutate(violator = as\_factor(as.character(violator))) %>%  
 mutate(violator = fct\_recode(violator,  
 "Violated" = "1",  
 "No Violation" = "0"))

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

summary(train)

## male race age state time.served   
## Male :394 White:271 Min. :18.40 Other : 99 Min. :0.000   
## Female: 79 Other:202 1st Qu.:25.00 Kentucky : 79 1st Qu.:3.200   
## Median :33.30 Louisiana: 60 Median :4.400   
## Mean :34.04 Virginia :235 Mean :4.174   
## 3rd Qu.:41.90 3rd Qu.:5.200   
## Max. :65.10 Max. :6.000   
## max.sentence multiple.offenses crime violator   
## Min. : 1.0 One :224 driving-related: 65 No Violation:418   
## 1st Qu.:12.0 Multiple:249 drug-related :110 Violated : 55   
## Median :13.0 Other :223   
## Mean :13.1 larceny : 75   
## 3rd Qu.:15.0   
## Max. :18.0

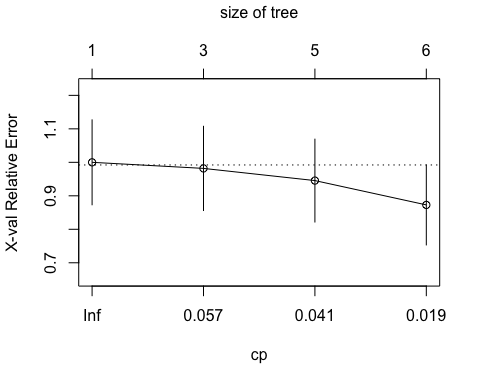
tree1 = rpart(violator ~., train, method = "class")  
fancyRpartPlot(tree1)

 I would class the 40 year-old parolee from Louisiana as having No Violations. I came to this by first selecting no as he is not from Kentucky, Viriginia, or Other and then yes for less than 13 year sentence and yes for under the age of 44. Which arrives to the No Violation.

printcp(tree1)

##   
## Classification tree:  
## rpart(formula = violator ~ ., data = train, method = "class")  
##   
## Variables actually used in tree construction:  
## [1] age max.sentence multiple.offenses state   
##   
## Root node error: 55/473 = 0.11628  
##   
## n= 473   
##   
## CP nsplit rel error xerror xstd  
## 1 0.072727 0 1.00000 1.00000 0.12676  
## 2 0.045455 2 0.85455 0.98182 0.12575  
## 3 0.036364 4 0.76364 0.94545 0.12369  
## 4 0.010000 5 0.72727 0.87273 0.11940

plotcp(tree1)

 Based on this information the 1st CP value is the best as it is 0.05 and 1 is optimal.

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

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

## 1 2 3 4 5   
## No Violation No Violation No Violation No Violation No Violation   
## 6   
## No Violation   
## Levels: No Violation Violated

confusionMatrix(treepred, train$violator,positive = "No Violation")

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction No Violation Violated  
## No Violation 416 38  
## Violated 2 17  
##   
## Accuracy : 0.9154   
## 95% CI : (0.8866, 0.9389)  
## No Information Rate : 0.8837   
## P-Value [Acc > NIR] : 0.01576   
##   
## Kappa : 0.4251   
##   
## Mcnemar's Test P-Value : 3.13e-08   
##   
## Sensitivity : 0.9952   
## Specificity : 0.3091   
## Pos Pred Value : 0.9163   
## Neg Pred Value : 0.8947   
## Prevalence : 0.8837   
## Detection Rate : 0.8795   
## Detection Prevalence : 0.9598   
## Balanced Accuracy : 0.6522   
##   
## 'Positive' Class : No Violation   
##

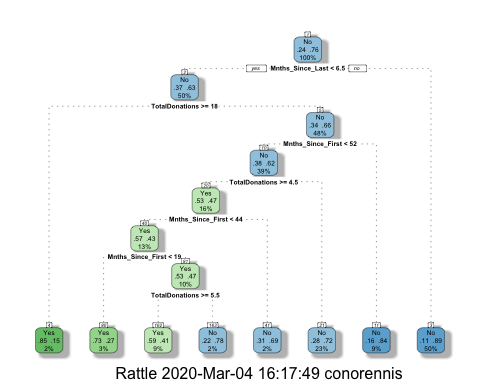
library(readxl)  
Blood <- read\_excel("Blood.xlsx")  
View(Blood)

## Warning in system2("/usr/bin/otool", c("-L", shQuote(DSO)), stdout = TRUE):  
## running command ''/usr/bin/otool' -L '/Library/Frameworks/R.framework/  
## Resources/modules/R\_de.so'' had status 1

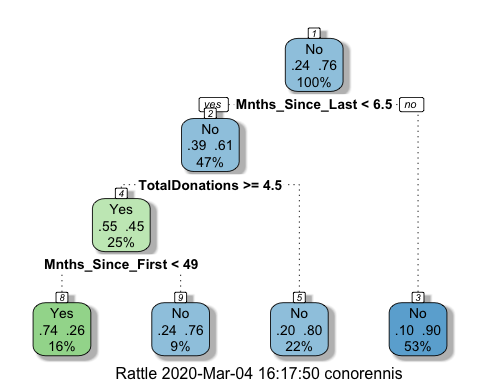
Blood = Blood %>% mutate(DonatedMarch = as\_factor(as.character(DonatedMarch))) %>%  
 mutate(DonatedMarch = fct\_recode(DonatedMarch,  
 "No" = "0",  
 "Yes" = "1"))

train.rows2 = createDataPartition(y = Blood$DonatedMarch, p=0.7, list = FALSE)  
train2 = Blood[train.rows2,]  
test2 = Blood[-train.rows2,]  
set.seed(12345)

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



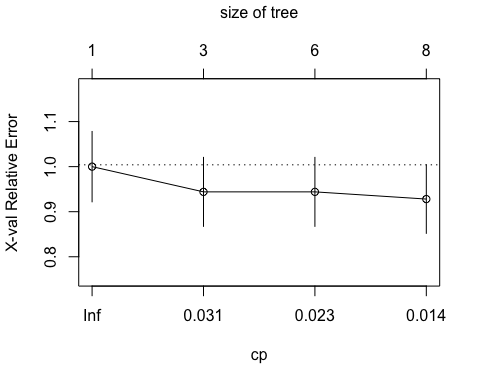
tree5 = rpart(DonatedMarch ~., test2, method = "class")  
fancyRpartPlot(tree5)



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 0.036000 0 1.000 1.000 0.078049  
## 2 0.026667 2 0.928 0.944 0.076494  
## 3 0.020000 5 0.848 0.944 0.076494  
## 4 0.010000 7 0.808 0.928 0.076030

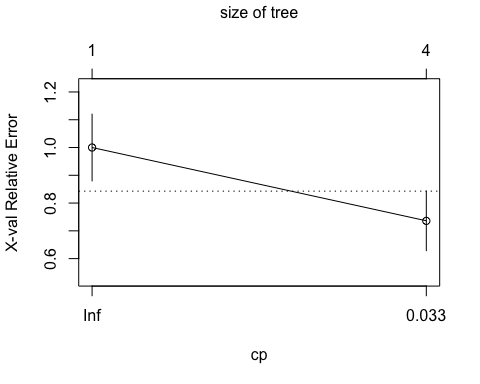
plotcp(tree3)



printcp(tree5)

##   
## Classification tree:  
## rpart(formula = DonatedMarch ~ ., data = test2, method = "class")  
##   
## Variables actually used in tree construction:  
## [1] Mnths\_Since\_First Mnths\_Since\_Last TotalDonations   
##   
## Root node error: 53/224 = 0.23661  
##   
## n= 224   
##   
## CP nsplit rel error xerror xstd  
## 1 0.10692 0 1.00000 1.00000 0.12002  
## 2 0.01000 3 0.67925 0.73585 0.10708

plotcp(tree5)



tree4 = prune(tree3, cp= tree3$cptable[which.min(tree3$cptable[, "xerror"]),"CP"])  
tree6 = prune(tree5, cp= tree5$cptable[which.min(tree5$cptable[, "xerror"]),"CP"])