### Classification Trees

### BAN502

### Module 4: Assignment 1

### Mayberry, Brandon

Libraries

library(tidyverse)

## Warning: package 'tidyverse' was built under R version 3.5.2

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

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

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

library(caret)

## Warning: package 'caret' was built under R version 3.5.2

## Loading required package: lattice

##   
## Attaching package: 'caret'

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

library(rpart)

## Warning: package 'rpart' was built under R version 3.5.2

library(rattle)

## Warning: package 'rattle' was built under R version 3.5.2

## 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)

## Warning: package 'RColorBrewer' was built under R version 3.5.2

library(e1071)

## Warning: package 'e1071' was built under R version 3.5.2

Read in dataset

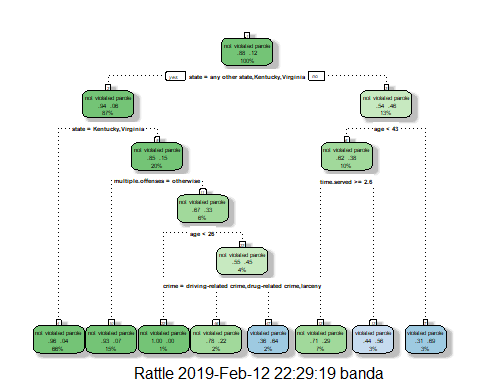
parole = read.csv("parole.csv")  
parole = parole %>% mutate(male = as\_factor(as.character(male))) %>%  
 mutate(male = fct\_recode(male,  
 "male" = "1",  
 "female" = "0")) %>%  
 mutate(race = as\_factor(as.character(race))) %>%  
 mutate(race = fct\_recode(race,  
 "white" = "1",  
 "otherwise" = "2")) %>%  
 mutate(state = as\_factor(as.character(state))) %>%  
 mutate(state = fct\_recode(state,  
 "any other state" = "1",  
 "Kentucky" = "2",  
 "Louisiana" = "3",  
 "Virginia" = "4")) %>%  
 mutate(crime = as\_factor(as.character(crime))) %>%  
 mutate(crime = fct\_recode(crime,  
 "any other crime" = "1",  
 "larceny" = "2",  
 "drug-related crime" = "3",  
 "driving-related crime" = "4")) %>%  
 mutate(multiple.offenses = as\_factor(as.character(multiple.offenses))) %>%  
 mutate(multiple.offenses = fct\_recode(multiple.offenses,  
 "multiple offenses" = "1",  
 "otherwise" ="0")) %>%  
 mutate(violator = as\_factor(as.character(violator))) %>%  
 mutate(violator = fct\_recode(violator,  
 "violated parole" = "1",  
 "not violated parole" = "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

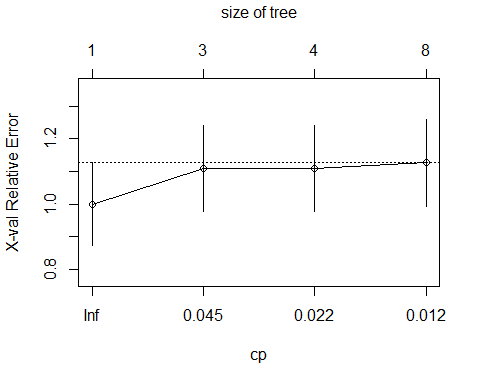
A 40 year old parolee from Louisiana who served a 5yr prison sentence would not violate his parole. Starting at the top of the tree, since the parolee is from Louisiana, we would choose “no”. Then the next branch we would choose < 43 and then time served would be >=2.6 which leads us to the conclusion that he would not violate his parole.

Task 4

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)



.054545 is the cp value that should be selected.

Task 5

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

## male race age state   
## male :384 white :269 Min. :18.40 any other state: 96   
## female: 89 otherwise:204 1st Qu.:25.10 Kentucky : 75   
## Median :33.20 Louisiana : 63   
## Mean :34.07 Virginia :239   
## 3rd Qu.:42.00   
## Max. :67.00   
## time.served max.sentence multiple.offenses  
## Min. :0.000 Min. : 1.00 otherwise :205   
## 1st Qu.:3.100 1st Qu.:12.00 multiple offenses:268   
## Median :4.300 Median :12.00   
## Mean :4.136 Mean :13.01   
## 3rd Qu.:5.200 3rd Qu.:15.00   
## Max. :6.000 Max. :18.00   
## crime violator   
## driving-related crime: 75 not violated parole:418   
## drug-related crime :103 violated parole : 55   
## any other crime :219   
## larceny : 76   
##   
##

The majority class are parolees who did not violate their parole.

Task 6

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

## 1 3 4   
## not violated parole not violated parole not violated parole   
## 5 6 7   
## not violated parole not violated parole not violated parole   
## Levels: not violated parole violated parole

confusionMatrix(treepred,train$violator,positive="violated parole")

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction not violated parole violated parole  
## not violated parole 402 28  
## violated parole 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 : 0.49091   
## Specificity : 0.96172   
## Pos Pred Value : 0.62791   
## Neg Pred Value : 0.93488   
## Prevalence : 0.11628   
## Detection Rate : 0.05708   
## Detection Prevalence : 0.09091   
## Balanced Accuracy : 0.72632   
##   
## 'Positive' Class : violated parole   
##

Task 7

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

## 2 11 13   
## not violated parole violated parole not violated parole   
## 14 17 18   
## not violated parole not violated parole not violated parole   
## Levels: not violated parole violated parole

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

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction not violated parole violated parole  
## not violated parole 170 19  
## violated parole 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 : 0.17391   
## Specificity : 0.94972   
## Pos Pred Value : 0.30769   
## Neg Pred Value : 0.89947   
## Prevalence : 0.11386   
## Detection Rate : 0.01980   
## Detection Prevalence : 0.06436   
## Balanced Accuracy : 0.56182   
##   
## 'Positive' Class : violated parole   
##

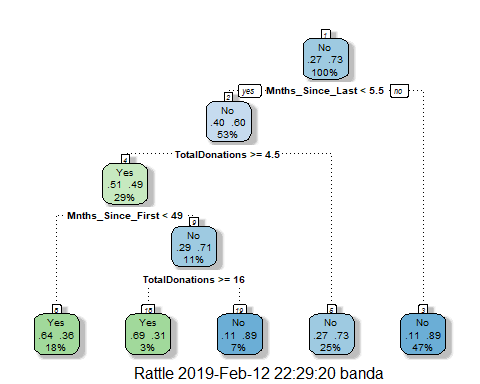
Our model is 86.14% accurate for the testing set which is slightly lower than the naive prediction of 88.61% if we had just said that all of the parolees did not violate their parole. The classification tree does not increase accuracy.

Task 8

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

Task 9

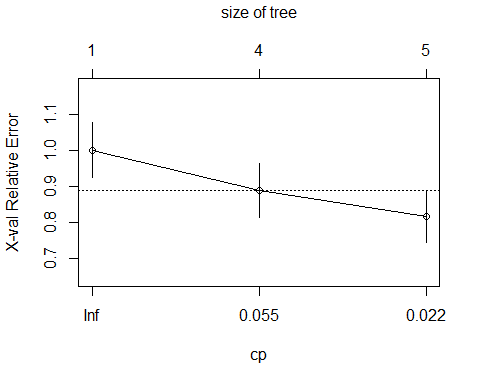
set.seed(1234)  
train.rows2 = createDataPartition(y = Blood$DonatedMarch, p=0.7, list = FALSE)  
train2 = Blood[train.rows,]   
test2 = Blood[-train.rows,]  
tree3 = rpart(DonatedMarch ~., train2, method="class")  
fancyRpartPlot(tree3)



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: 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

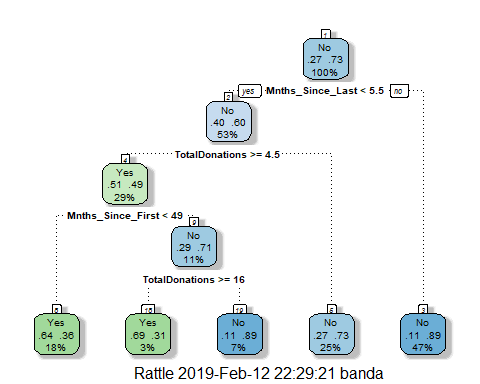
plotcp(tree3)



The complexity parameter selection for this model is .01

Task 10

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



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

## 1 3 4 5 6 7   
## Yes Yes Yes Yes No Yes   
## Levels: Yes No

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

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 65 35  
## No 61 312  
##   
## 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 : 0.5159   
## Specificity : 0.8991   
## Pos Pred Value : 0.6500   
## Neg Pred Value : 0.8365   
## Prevalence : 0.2664   
## Detection Rate : 0.1374   
## Detection Prevalence : 0.2114   
## Balanced Accuracy : 0.7075   
##   
## 'Positive' Class : Yes   
##

treepred\_test2 = predict(tree4, test2, type = "class")  
head(treepred\_test2)

## 2 11 13 14 17 18   
## Yes Yes Yes Yes Yes No   
## Levels: Yes No

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

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 18 19  
## No 34 204  
##   
## 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 : 0.34615   
## Specificity : 0.91480   
## Pos Pred Value : 0.48649   
## Neg Pred Value : 0.85714   
## Prevalence : 0.18909   
## Detection Rate : 0.06545   
## Detection Prevalence : 0.13455   
## Balanced Accuracy : 0.63048   
##   
## 'Positive' Class : Yes   
##

The training model is 79.7% which is higher than the naive prediciton of 73.36% on the training data. The testing model is 80.73% which is lower than the naive prediction of 81.09% on the testing data. So on the testing data the classification tree model does not add to its accuracy.