options(tidyverse.quiet=TRUE)  
library(tidyverse)  
library(caret)  
library(rpart)  
library(rattle)  
library(RColorBrewer)

parole<-read\_csv("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()  
## )

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",  
 "otherwise"="2"))

parole=parole%>% mutate(state=as\_factor(as.character(state)))%>%  
 mutate(state=fct\_recode(state,  
 "Other"="1",  
 "Kentucky"="2",  
 "Louisiana"="3",  
 "Virginia"="4"))

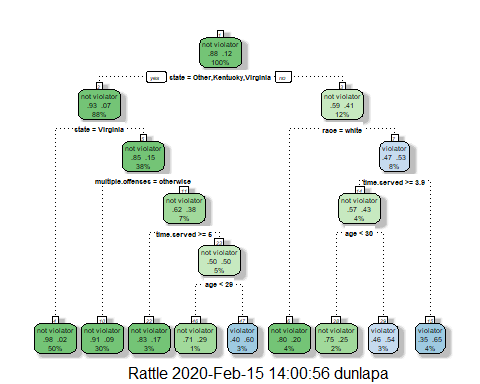
parole=parole%>% mutate(crime=as\_factor(as.character(crime)))%>%  
 mutate(crime=fct\_recode(crime,  
 "Other crime"="1",  
 "larceny"="2",  
 "drug-related"="3",  
 "driving-related"="4"))

parole=parole%>% mutate(multiple.offenses=as\_factor(as.character(multiple.offenses)))%>%  
 mutate(multiple.offenses=fct\_recode(multiple.offenses,  
 "multiple offenses"="1",  
 "otherwise"="0"))

parole=parole%>% mutate(violator=as\_factor(as.character(violator)))%>%  
 mutate(violator=fct\_recode(violator,  
 "violator"="1",  
 "not violator"="0"))

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

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



40 year old parolee from Louisiana who served a 5 year prison sentence? If white: Not a violator - Because he is from Louisiana we go right, because he is white, we go left, which makes him a non-violaotor. Other: Violator Because he is from Lousianan we go right, because he his not white we go right againbecause he served 5 years we go left, because he is over 30 it lead to Violator.

printcp(traintree)

##   
## Classification tree:  
## rpart(formula = violator ~ ., data = train, method = "class")  
##   
## Variables actually used in tree construction:  
## [1] age multiple.offenses race state   
## [5] time.served   
##   
## Root node error: 55/473 = 0.11628  
##   
## n= 473   
##   
## CP nsplit rel error xerror xstd  
## 1 0.030303 0 1.00000 1.0000 0.12676  
## 2 0.018182 3 0.90909 1.1091 0.13253  
## 3 0.013636 4 0.89091 1.2182 0.13788  
## 4 0.010000 8 0.83636 1.2000 0.13702

traintree2 = prune(traintree,cp= traintree$cptable[which.min(traintree$cptable[,"xerror"]),"CP"])

The majority class is non-violators.

traintreepred=predict(traintree,train,type="class")  
head(traintreepred)

## 1 2 3 4 5 6   
## not violator not violator not violator not violator not violator not violator   
## Levels: not violator violator

confusionMatrix(traintreepred,train$violator,positive="violator")

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction not violator violator  
## not violator 400 28  
## violator 18 27  
##   
## Accuracy : 0.9027   
## 95% CI : (0.8724, 0.9279)  
## No Information Rate : 0.8837   
## P-Value [Acc > NIR] : 0.1095   
##   
## Kappa : 0.4862   
##   
## Mcnemar's Test P-Value : 0.1845   
##   
## Sensitivity : 0.49091   
## Specificity : 0.95694   
## Pos Pred Value : 0.60000   
## Neg Pred Value : 0.93458   
## Prevalence : 0.11628   
## Detection Rate : 0.05708   
## Detection Prevalence : 0.09514   
## Balanced Accuracy : 0.72392   
##   
## 'Positive' Class : violator   
##

testpred=predict(traintree,test,type="class")  
head(testpred)

## 1 2 3 4 5 6   
## not violator not violator not violator not violator not violator not violator   
## Levels: not violator violator

confusionMatrix(testpred,test$violator,positive="violator")

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction not violator violator  
## not violator 171 13  
## violator 8 10  
##   
## Accuracy : 0.896   
## 95% CI : (0.8455, 0.9345)  
## No Information Rate : 0.8861   
## P-Value [Acc > NIR] : 0.3797   
##   
## Kappa : 0.4309   
##   
## Mcnemar's Test P-Value : 0.3827   
##   
## Sensitivity : 0.43478   
## Specificity : 0.95531   
## Pos Pred Value : 0.55556   
## Neg Pred Value : 0.92935   
## Prevalence : 0.11386   
## Detection Rate : 0.04950   
## Detection Prevalence : 0.08911   
## Balanced Accuracy : 0.69504   
##   
## 'Positive' Class : violator   
##

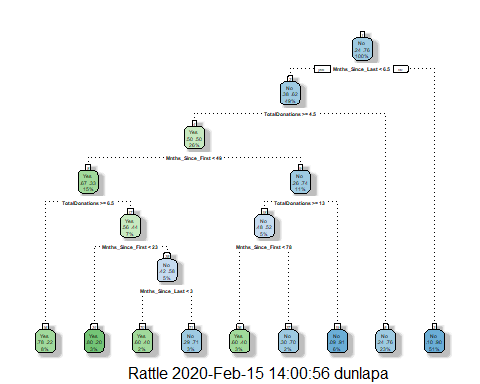
Blood<-read\_csv("Blood.csv")

## 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()  
## )

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

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

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



printcp(traintreeBlood)

##   
## 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 0.072 0 1.000 1.000 0.078049  
## 2 0.016 3 0.784 0.880 0.074580  
## 3 0.012 6 0.736 0.912 0.075556  
## 4 0.010 8 0.712 0.928 0.076030

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

traintreeBloodpred=predict(traintreeBlood,train2,type="class")  
head(traintreeBloodpred)

## 1 2 3 4 5 6   
## Yes Yes Yes No No Yes   
## Levels: Yes No

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

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 53 26  
## No 72 373  
##   
## Accuracy : 0.813   
## 95% CI : (0.7769, 0.8455)  
## No Information Rate : 0.7615   
## P-Value [Acc > NIR] : 0.002713   
##   
## Kappa : 0.4107   
##   
## Mcnemar's Test P-Value : 5.476e-06   
##   
## Sensitivity : 0.4240   
## Specificity : 0.9348   
## Pos Pred Value : 0.6709   
## Neg Pred Value : 0.8382   
## Prevalence : 0.2385   
## Detection Rate : 0.1011   
## Detection Prevalence : 0.1508   
## Balanced Accuracy : 0.6794   
##   
## 'Positive' Class : Yes   
##

testpredBlood=predict(traintreeBlood,test2,type="class")  
head(testpredBlood)

## 1 2 3 4 5 6   
## No Yes Yes No No Yes   
## Levels: Yes No

confusionMatrix(testpredBlood,test2$DonatedMarch,positive="Yes")

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 18 20  
## No 35 151  
##   
## Accuracy : 0.7545   
## 95% CI : (0.6927, 0.8094)  
## No Information Rate : 0.7634   
## P-Value [Acc > NIR] : 0.65710   
##   
## Kappa : 0.2468   
##   
## Mcnemar's Test P-Value : 0.05906   
##   
## Sensitivity : 0.33962   
## Specificity : 0.88304   
## Pos Pred Value : 0.47368   
## Neg Pred Value : 0.81183   
## Prevalence : 0.23661   
## Detection Rate : 0.08036   
## Detection Prevalence : 0.16964   
## Balanced Accuracy : 0.61133   
##   
## 'Positive' Class : Yes   
##

It looks like accuracy decreases from 81.3% to 75.45% between the training and testing data when I run these predictions. Sensitivity and specificity also go down.