# Logistic regression (Classification)

library(tidyverse)

## -- 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(MASS)

##   
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':  
##   
## select

library(ROCR)

## Loading required package: gplots

##   
## Attaching package: 'gplots'

## The following object is masked from 'package:stats':  
##   
## lowess

library(caret)

## Loading required package: lattice

##   
## Attaching package: 'caret'

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

Conversions

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()  
## )

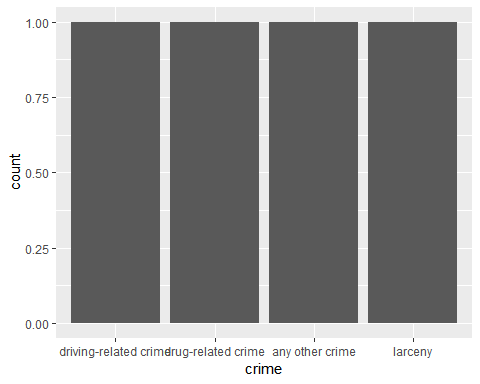
Parole = Parole %>% mutate(male = as\_factor(as.character(male))) %>%  
mutate(male = fct\_recode(male,  
"male" = "1",  
"female" = "0"))  
  
Parole = Parole %>% mutate(race = as\_factor(as.character(race))) %>%  
mutate(race = fct\_recode(race,  
"white" = "1",  
"otherwise" = "2"))  
  
Parole = Parole %>% mutate(state = as\_factor(as.character(state))) %>%  
mutate(state = fct\_recode(state,  
"any other state" = "1",  
"Kentucky" = "2",  
"Louisiana" = "3",  
"Virginia" = "4"))  
  
Parole = Parole %>% 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"))  
  
Parole = Parole %>% mutate(multiple.offenses = as\_factor(as.character(multiple.offenses))) %>%  
mutate(multiple.offenses = fct\_recode(multiple.offenses,  
"incarcerated" = "1",  
"otherwise" = "0"))  
  
Parole = Parole %>% mutate(violator = as\_factor(as.character(violator))) %>%  
mutate(violator = fct\_recode(violator,  
"parolee violated parole" = "1",  
"no violation" = "0"))

# Task 1 Split data into training and testing

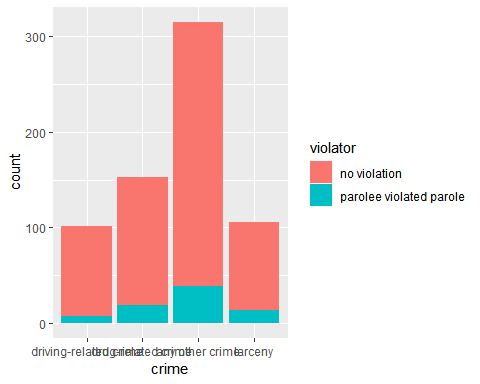
set.seed(12345)  
train.row = createDataPartition(y= Parole$violator, p=0.7, list = FALSE)   
train = Parole[train.row,]  
test = Parole[-train.row,]

# Task 2 Predict whether a Parolee will violate

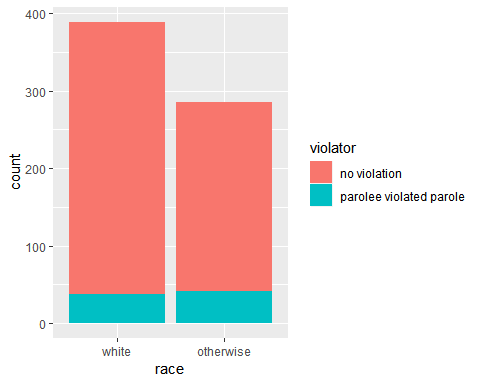
ggplot(Parole, aes(x= crime)) + geom\_bar(position = "fill")



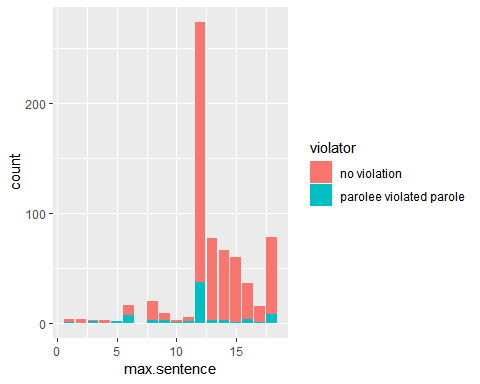
ggplot(Parole, aes(x= crime, fill = violator)) + geom\_bar()



ggplot(Parole, aes(x= race, fill = violator)) + geom\_bar()



ggplot(Parole, aes(x= max.sentence, fill = violator)) + geom\_bar()



To decide whether a parolee with violate their parole I decided to test 3 different factors. I choose crime, race, and max.sentence. In my opinion the more intense crime someone commited will probably lead to bad behavior while on parole. The max sentence someone was given could also increase chances of parole because after so many years people would end up in a bad situation again. I chose Race because this one is always looked at when it comes to crime report data. This one can identify if crime is widely spread or more at one race.

# Task 3 Identify a variable

t1 = table(Parole$crime, Parole$violator)  
prop.table(t1, margin = 2)

##   
## no violation parolee violated parole  
## driving-related crime 0.15745394 0.08974359  
## drug-related crime 0.22445561 0.24358974  
## any other crime 0.46231156 0.50000000  
## larceny 0.15577889 0.16666667

mod1 = glm(crime ~ violator , train, family = "binomial")  
summary(mod1)

##   
## Call:  
## glm(formula = crime ~ violator, family = "binomial", data = train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.2896 0.3886 0.6102 0.6102 0.6102   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 1.5866 0.1303 12.181 <2e-16 \*\*\*  
## violatorparolee violated parole 0.9589 0.5353 1.791 0.0733 .   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 413.67 on 472 degrees of freedom  
## Residual deviance: 409.60 on 471 degrees of freedom  
## AIC: 413.6  
##   
## Number of Fisher Scoring iterations: 5

The factor of crime comitted versus violation of parole was not as strong as I thought. It appears those who comitted what I would consider higher crimes drug and driving actually violated or not violated about the same. With drug related crimes about 22.4% did not violate, where as only 24.35% violated. From this it does not appear crime affects the violator.

# Task 4

allmod = glm(crime ~., Parole, family = "binomial")   
summary(allmod)

##   
## Call:  
## glm(formula = crime ~ ., family = "binomial", data = Parole)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.5963 0.2291 0.4707 0.6217 1.1839   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 2.78384 0.82774 3.363 0.000770 \*\*\*  
## malefemale 0.44869 0.33004 1.360 0.173982   
## raceotherwise 0.52304 0.25916 2.018 0.043567 \*   
## age -0.03718 0.01075 -3.458 0.000544 \*\*\*  
## stateKentucky 1.91733 0.55847 3.433 0.000597 \*\*\*  
## stateLouisiana 1.04085 0.56810 1.832 0.066926 .   
## stateVirginia 0.34412 0.32641 1.054 0.291762   
## time.served 0.17064 0.09499 1.796 0.072432 .   
## max.sentence -0.07651 0.04407 -1.736 0.082558 .   
## multiple.offensesincarcerated -0.32540 0.28470 -1.143 0.253057   
## violatorparolee violated parole 0.47961 0.47097 1.018 0.308506   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 569.79 on 674 degrees of freedom  
## Residual deviance: 511.51 on 664 degrees of freedom  
## AIC: 533.51  
##   
## Number of Fisher Scoring iterations: 6

emptymod = glm(crime~1, Parole, family = "binomial")   
summary(emptymod)

##   
## Call:  
## glm(formula = crime ~ 1, family = "binomial", data = Parole)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.9491 0.5694 0.5694 0.5694 0.5694   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 1.7375 0.1079 16.1 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 569.79 on 674 degrees of freedom  
## Residual deviance: 569.79 on 674 degrees of freedom  
## AIC: 571.79  
##   
## Number of Fisher Scoring iterations: 4

Forward Stepwise

forwardmod = stepAIC(emptymod, direction = "forward", scope=list(upper=allmod,lower=emptymod), trace = TRUE)

## Start: AIC=571.79  
## crime ~ 1  
##   
## Df Deviance AIC  
## + state 3 536.44 544.44  
## + age 1 558.21 562.21  
## + max.sentence 1 559.31 563.31  
## + race 1 565.11 569.11  
## + male 1 566.41 570.41  
## + violator 1 567.00 571.00  
## + time.served 1 567.24 571.24  
## <none> 569.79 571.79  
## + multiple.offenses 1 568.19 572.19  
##   
## Step: AIC=544.44  
## crime ~ state  
##   
## Df Deviance AIC  
## + age 1 526.18 536.18  
## + race 1 531.01 541.01  
## + max.sentence 1 533.62 543.62  
## + time.served 1 534.32 544.32  
## <none> 536.44 544.44  
## + male 1 534.82 544.82  
## + violator 1 535.90 545.90  
## + multiple.offenses 1 536.11 546.11  
##   
## Step: AIC=536.18  
## crime ~ state + age  
##   
## Df Deviance AIC  
## + race 1 520.67 532.67  
## + time.served 1 523.52 535.52  
## + male 1 523.84 535.84  
## + max.sentence 1 523.86 535.86  
## <none> 526.18 536.18  
## + multiple.offenses 1 525.35 537.35  
## + violator 1 525.59 537.59  
##   
## Step: AIC=532.67  
## crime ~ state + age + race  
##   
## Df Deviance AIC  
## + time.served 1 518.11 532.11  
## + max.sentence 1 518.49 532.49  
## + male 1 518.54 532.54  
## <none> 520.67 532.67  
## + multiple.offenses 1 520.18 534.18  
## + violator 1 520.25 534.25  
##   
## Step: AIC=532.11  
## crime ~ state + age + race + time.served  
##   
## Df Deviance AIC  
## + max.sentence 1 515.22 531.22  
## + male 1 516.06 532.06  
## <none> 518.11 532.11  
## + violator 1 517.62 533.62  
## + multiple.offenses 1 517.65 533.65  
##   
## Step: AIC=531.22  
## crime ~ state + age + race + time.served + max.sentence  
##   
## Df Deviance AIC  
## <none> 515.22 531.22  
## + male 1 513.49 531.49  
## + multiple.offenses 1 514.40 532.40  
## + violator 1 514.70 532.70

summary(forwardmod)

##   
## Call:  
## glm(formula = crime ~ state + age + race + time.served + max.sentence,   
## family = "binomial", data = Parole)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.6591 0.2423 0.4840 0.6211 1.1171   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 2.67259 0.79451 3.364 0.000769 \*\*\*  
## stateKentucky 1.97416 0.55510 3.556 0.000376 \*\*\*  
## stateLouisiana 1.05542 0.54680 1.930 0.053585 .   
## stateVirginia 0.10535 0.27328 0.386 0.699859   
## age -0.03451 0.01070 -3.225 0.001259 \*\*   
## raceotherwise 0.58203 0.25688 2.266 0.023464 \*   
## time.served 0.17264 0.09506 1.816 0.069370 .   
## max.sentence -0.07374 0.04350 -1.695 0.090033 .   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 569.79 on 674 degrees of freedom  
## Residual deviance: 515.22 on 667 degrees of freedom  
## AIC: 531.22  
##   
## Number of Fisher Scoring iterations: 6

In the forward stepwise model the variables that are significant are the state of Louisiana and Kentucky, age, and other races.

Backward Stepwise

backmod = stepAIC(allmod, direction = "backward", trace = TRUE)

## Start: AIC=533.51  
## crime ~ male + race + age + state + time.served + max.sentence +   
## multiple.offenses + violator  
##   
## Df Deviance AIC  
## - violator 1 512.61 532.61  
## - multiple.offenses 1 512.84 532.84  
## - male 1 513.49 533.49  
## <none> 511.51 533.51  
## - max.sentence 1 514.55 534.55  
## - time.served 1 514.73 534.73  
## - race 1 515.65 535.65  
## - age 1 523.61 543.61  
## - state 3 529.69 545.69  
##   
## Step: AIC=532.61  
## crime ~ male + race + age + state + time.served + max.sentence +   
## multiple.offenses  
##   
## Df Deviance AIC  
## - multiple.offenses 1 513.49 531.49  
## - male 1 514.40 532.40  
## <none> 512.61 532.61  
## - max.sentence 1 515.53 533.53  
## - time.served 1 515.72 533.72  
## - race 1 517.18 535.18  
## - age 1 524.38 542.38  
## - state 3 532.11 546.11  
##   
## Step: AIC=531.49  
## crime ~ male + race + age + state + time.served + max.sentence  
##   
## Df Deviance AIC  
## - male 1 515.22 531.22  
## <none> 513.49 531.49  
## - max.sentence 1 516.06 532.06  
## - time.served 1 516.62 532.62  
## - race 1 518.49 534.49  
## - age 1 524.67 540.67  
## - state 3 535.79 547.79  
##   
## Step: AIC=531.22  
## crime ~ race + age + state + time.served + max.sentence  
##   
## Df Deviance AIC  
## <none> 515.22 531.22  
## - max.sentence 1 518.11 532.11  
## - time.served 1 518.49 532.49  
## - race 1 520.45 534.45  
## - age 1 525.73 539.73  
## - state 3 538.32 548.32

summary(backmod)

##   
## Call:  
## glm(formula = crime ~ race + age + state + time.served + max.sentence,   
## family = "binomial", data = Parole)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.6591 0.2423 0.4840 0.6211 1.1171   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 2.67259 0.79451 3.364 0.000769 \*\*\*  
## raceotherwise 0.58203 0.25688 2.266 0.023464 \*   
## age -0.03451 0.01070 -3.225 0.001259 \*\*   
## stateKentucky 1.97416 0.55510 3.556 0.000376 \*\*\*  
## stateLouisiana 1.05542 0.54680 1.930 0.053585 .   
## stateVirginia 0.10535 0.27328 0.386 0.699859   
## time.served 0.17264 0.09506 1.816 0.069370 .   
## max.sentence -0.07374 0.04350 -1.695 0.090033 .   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 569.79 on 674 degrees of freedom  
## Residual deviance: 515.22 on 667 degrees of freedom  
## AIC: 531.22  
##   
## Number of Fisher Scoring iterations: 6

In the backward stepwise model the variables that are significant are raceotherwise, age, and the state of kentucky and Lousiana.

In these model the predictions seem accurate, the state of Vriginia is not a significant factor towards whether someone will violate their parole compared to the state of Kentucky and Louisana. The time served on parole was not a significant facotr, which is shocking because I would have thought that would play a factor into if someone would violate the Parole.

# Task 5- State, multiple offenses, race

allmod5 = glm(violator ~ state + multiple.offenses + race, train, family = "binomial")  
  
#Sorry for all the code, I tested a lot of code when struggling with this task.   
mod2 = glm(state ~ violator , train, family = "binomial")  
summary(mod2)

##   
## Call:  
## glm(formula = state ~ violator, family = "binomial", data = train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.8049 0.6609 0.6609 0.6609 0.7665   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 1.4104 0.1232 11.451 <2e-16 \*\*\*  
## violatorparolee violated parole -0.3359 0.3332 -1.008 0.313   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 477.24 on 472 degrees of freedom  
## Residual deviance: 476.26 on 471 degrees of freedom  
## AIC: 480.26  
##   
## Number of Fisher Scoring iterations: 4

mod3 = glm(multiple.offenses ~ violator , train, family = "binomial")  
summary(mod3)

##   
## Call:  
## glm(formula = multiple.offenses ~ violator, family = "binomial",   
## data = train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.6542 -1.2516 0.7665 1.1050 1.1050   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 0.17268 0.09819 1.759 0.07864 .   
## violatorparolee violated parole 0.90184 0.32475 2.777 0.00549 \*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 647.30 on 472 degrees of freedom  
## Residual deviance: 638.77 on 471 degrees of freedom  
## AIC: 642.77  
##   
## Number of Fisher Scoring iterations: 4

mod4 = glm(race ~ violator , train, family = "binomial")  
summary(mod4)

##   
## Call:  
## glm(formula = race ~ violator, family = "binomial", data = train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.288 -1.034 -1.034 1.328 1.328   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -0.34797 0.09931 -3.504 0.000458 \*\*\*  
## violatorparolee violated parole 0.60390 0.28946 2.086 0.036951 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 646.76 on 472 degrees of freedom  
## Residual deviance: 642.36 on 471 degrees of freedom  
## AIC: 646.36  
##   
## Number of Fisher Scoring iterations: 4

tstate = table(Parole$state, Parole$violator)  
prop.table(tstate, margin = 2)

##   
## no violation parolee violated parole  
## any other state 0.20603015 0.25641026  
## Kentucky 0.17755444 0.17948718  
## Louisiana 0.07537688 0.47435897  
## Virginia 0.54103853 0.08974359

tmultiple = table(Parole$multiple.offenses, Parole$violator)  
prop.table(tmultiple, margin = 2)

##   
## no violation parolee violated parole  
## otherwise 0.4824121 0.3205128  
## incarcerated 0.5175879 0.6794872

tRACE1 = table(Parole$race, Parole$violator)  
prop.table(tRACE1, margin = 2)

##   
## no violation parolee violated parole  
## white 0.5896147 0.4743590  
## otherwise 0.4103853 0.5256410

From this logisitic regression model we can see that state was not a significant factors on whether parolee’s violated their parole. Those parolee’s who were multiple offenders did have a significant value and this led to becoming a violator of parole. Race shows us that both the white race and all other races were significant values meaning they were likely to break their paroles.

# Task 6

data1 = data.frame(state = "Louisiana", multiple.offenses = "incarcerated", race = "white")  
predict(allmod5, data1, type="response")

## 1   
## 0.408682

data2 = data.frame(state = "Kentucky", multiple.offenses = "otherwise", race = "otherwise")  
predict(allmod5, data2, type = "response")

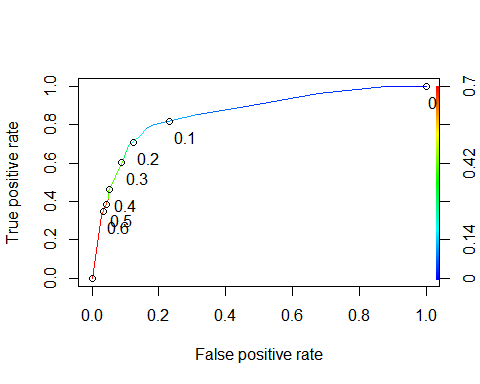
## 1   
## 0.1153326

# Task 7

predictions = predict(allmod5, train, type="response")   
head(predictions, 10)

## 1 2 3 4 5 6   
## 0.07187555 0.17425270 0.07187555 0.17425270 0.17425270 0.07187555   
## 7 8 9 10   
## 0.07187555 0.07187555 0.17425270 0.07187555

ROCRpred = prediction(predictions, train$violator)   
  
###You shouldn't need to ever change the next two lines:  
ROCRperf = performance(ROCRpred, "tpr", "fpr")  
plot(ROCRperf, colorize=TRUE, print.cutoffs.at=seq(0,1,by=0.1), text.adj=c(-0.2,1.7))



# Task 8- Accuracy, Sensitivity, and Specificty

as.numeric(performance(ROCRpred, "auc")@y.values)

## [1] 0.8586124

opt.cut = function(perf, pred){  
 cut.ind = mapply(FUN=function(x, y, p){  
 d = (x - 0)^2 + (y-1)^2  
 ind = which(d == min(d))  
 c(sensitivity = y[[ind]], specificity = 1-x[[ind]],   
 cutoff = p[[ind]])  
 }, perf@x.values, perf@y.values, pred@cutoffs)  
}  
print(opt.cut(ROCRperf, ROCRpred))

## [,1]  
## sensitivity 0.7818182  
## specificity 0.8373206  
## cutoff 0.1161882

I minored in Sociology in undergrad and one of the things constantyl talked about was when telling someone they are identified as something, they will most liekly turn into that thing. In this example if you incorrectly classify a parolee they would begin to believe they are a violator and would be more likely to violate parole because they are constantly told this. If someone told person A they were not likely to violate parole, but person B is the majority of the time person B is liekly to violate because they were told they would.

# Task 9

terror = table(train$violator, predictions > 0.1161882)  
terror

##   
## FALSE TRUE  
## no violation 357 61  
## parolee violated parole 14 41

Calculate accuracy

(terror[1,1]+terror[2,2])/nrow(train)

## [1] 0.8414376

Above was the calculation with 0.1161882 threshold. We will now do trial and error on a 0.2, .3, .4, .5, and .6 threshold.

t.2 = table(train$violator, predictions > 0.2)  
t.2

##   
## FALSE TRUE  
## no violation 367 51  
## parolee violated parole 16 39

(t.2[1,1]+t.2[2,2])/nrow(train)

## [1] 0.858351

t.3 = table(train$violator, predictions > 0.3)  
t.3

##   
## FALSE TRUE  
## no violation 397 21  
## parolee violated parole 30 25

(t.3[1,1]+t.3[2,2])/nrow(train)

## [1] 0.8921776

t.4 = table(train$violator, predictions > 0.4)  
t.4

##   
## FALSE TRUE  
## no violation 397 21  
## parolee violated parole 30 25

(t.4[1,1]+t.4[2,2])/nrow(train)

## [1] 0.8921776

t.5 = table(train$violator, predictions > 0.5)  
t.5

##   
## FALSE TRUE  
## no violation 406 12  
## parolee violated parole 37 18

(t.5[1,1]+t.5[2,2])/nrow(train)

## [1] 0.8964059

t.6 = table(train$violator, predictions > 0.6)  
t.6

##   
## FALSE TRUE  
## no violation 406 12  
## parolee violated parole 37 18

(t.6[1,1]+t.6[2,2])/nrow(train)

## [1] 0.8964059

Here we can see .3 threshold and above kept producing the same scores. 0.3 will be the threshold to best maximize the accuracy of the traing set.

# Task 10

#Accuracy  
(t.3[1,1]+t.3[2,2])/nrow(test)

## [1] 2.089109