# Logistic Regression (Classification)

## Brian Fortier

### Module 3 Assignment 2

Loading Packages:

#install.packages("tidyverse")  
#install.packages("MASS")  
#install.packages("caret")  
#install.packages("ROCR")  
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(caret)

## Loading required package: lattice

##   
## Attaching package: 'caret'

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

library(ROCR)

## Loading required package: gplots

##   
## Attaching package: 'gplots'

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

Loading Parole and Conversions of Variables:

parole <- read.csv("parole.csv")  
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,  
 "Kentucky" = "2",  
 "Louisiana" = "3",  
 "Virginia" = "4",  
 "Other" = "1"))  
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(multiple.offenses = as\_factor(as.character(multiple.offenses))) %>%  
 mutate(multiple.offenses = fct\_recode(multiple.offenses,  
 "multiple" = "1",  
 "other" = "0"))  
parole = parole %>% mutate(violator = as\_factor(as.character(violator))) %>%  
 mutate(violator = fct\_recode(violator,  
 "violator" = "1",  
 "other" = "0"))  
parole = parole %>% drop\_na()  
glimpse(parole)

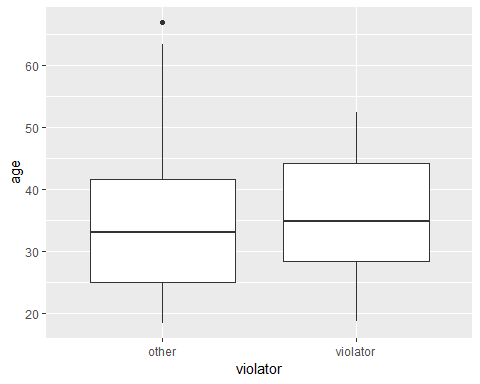
## Observations: 675  
## Variables: 9  
## $ male <fct> male, female, male, male, male, male, male, ...  
## $ race <fct> white, white, otherwise, white, otherwise, o...  
## $ age <dbl> 33.2, 39.7, 29.5, 22.4, 21.6, 46.7, 31.0, 24...  
## $ state <fct> Other, Other, Other, Other, Other, Other, Ot...  
## $ time.served <dbl> 5.5, 5.4, 5.6, 5.7, 5.4, 6.0, 6.0, 4.8, 4.5,...  
## $ max.sentence <int> 18, 12, 12, 18, 12, 18, 18, 12, 13, 12, 12, ...  
## $ multiple.offenses <fct> other, other, other, other, other, other, ot...  
## $ crime <fct> driving-related, drug-related, drug-related,...  
## $ violator <fct> other, other, other, other, other, other, ot...

Splitting Parole Data into Training/Testing Sets:

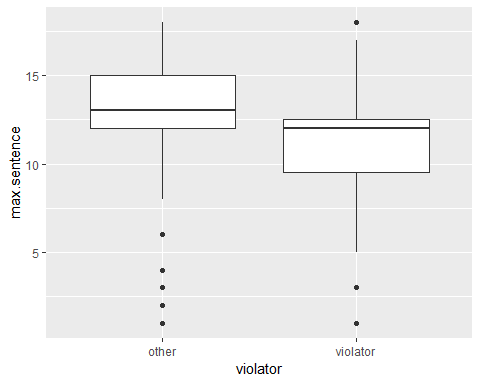
set.seed(12345)  
train.rows = createDataPartition(y = parole$violator, p=0.7, list=FALSE)  
train = parole[train.rows,]  
test = parole[-train.rows,]

Visualizing Training Set to Observe Variables:

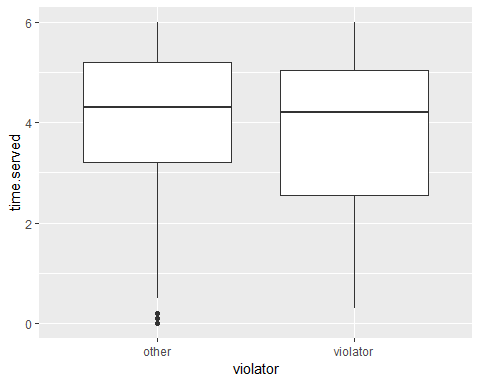
ggplot(train,aes(x=violator, y=age)) +   
 geom\_boxplot()



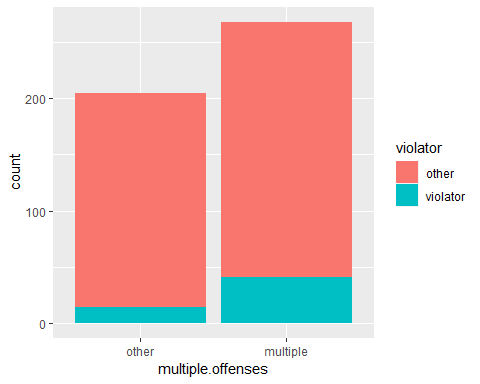
ggplot(train,aes(x=violator, y=max.sentence)) +   
 geom\_boxplot()



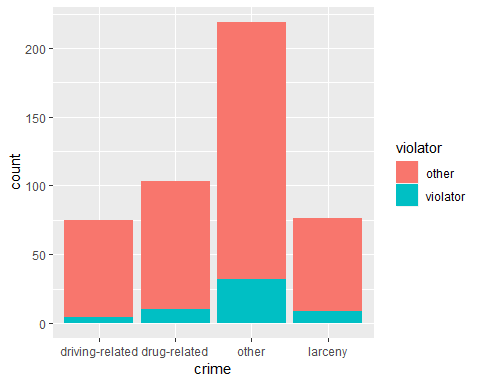
ggplot(train,aes(x=violator, y=time.served)) +   
 geom\_boxplot()



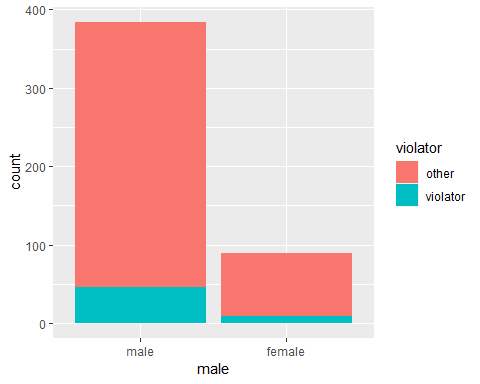
ggplot(train,aes(x=multiple.offenses, fill=violator)) +   
 geom\_bar()



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



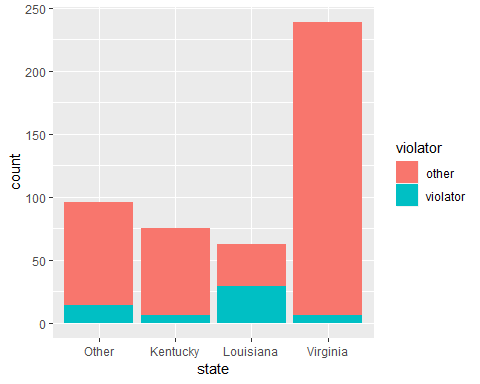
ggplot(train,aes(x=male, fill=violator)) +   
 geom\_bar()



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



ggplot(train,aes(x=state, fill=violator)) +   
 geom\_bar()



After looking through these variables, the multiple.offenses variable seems to have the most significant correlation to parole violation. This would show that parolee’s with multiple violations have a higher chance of creating another. Other significant variables would be state and race.

Logistic Regression Model with Variable: multiple.offenses:

mod1 = glm(violator ~ multiple.offenses, train, family = "binomial")  
summary(mod1)

##   
## Call:  
## glm(formula = violator ~ multiple.offenses, family = "binomial",   
## data = train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -0.5763 -0.5763 -0.3761 -0.3761 2.3169   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -2.6132 0.2769 -9.438 < 2e-16 \*\*\*  
## multiple.offensesmultiple 0.9018 0.3247 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: 340.04 on 472 degrees of freedom  
## Residual deviance: 331.50 on 471 degrees of freedom  
## AIC: 335.5  
##   
## Number of Fisher Scoring iterations: 5

We see a positive slope coefficient of about 1, so this will lead to higher count of violator with more multiple.offenses. The p-value is also <0.05 so we do see a significant value here. We also have a smaller AIC of 335.5 when compared to other variables.

Regression Setup:

allmod = glm(violator ~., train, family = "binomial")  
summary(allmod)

##   
## Call:  
## glm(formula = violator ~ ., family = "binomial", data = train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.9635 -0.3638 -0.2354 -0.1449 2.9869   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -5.33220 1.39750 -3.816 0.000136 \*\*\*  
## malefemale -0.53377 0.49107 -1.087 0.277051   
## raceotherwise 1.06698 0.41324 2.582 0.009824 \*\*   
## age 0.03361 0.01696 1.982 0.047493 \*   
## stateKentucky -0.30132 0.56939 -0.529 0.596665   
## stateLouisiana 0.87804 0.52428 1.675 0.093984 .   
## stateVirginia -3.46523 0.63742 -5.436 5.44e-08 \*\*\*  
## time.served -0.03009 0.12159 -0.247 0.804537   
## max.sentence 0.08458 0.05644 1.499 0.133963   
## multiple.offensesmultiple 1.72841 0.41857 4.129 3.64e-05 \*\*\*  
## crimedrug-related 0.11232 0.71712 0.157 0.875535   
## crimeother 0.87795 0.62271 1.410 0.158571   
## crimelarceny 1.06304 0.73146 1.453 0.146139   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 340.04 on 472 degrees of freedom  
## Residual deviance: 230.16 on 460 degrees of freedom  
## AIC: 256.16  
##   
## Number of Fisher Scoring iterations: 6

emptymod = glm(violator~1, train, family = "binomial")  
summary(emptymod)

##   
## Call:  
## glm(formula = violator ~ 1, family = "binomial", data = train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -0.4972 -0.4972 -0.4972 -0.4972 2.0745   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -2.0281 0.1434 -14.14 <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: 340.04 on 472 degrees of freedom  
## Residual deviance: 340.04 on 472 degrees of freedom  
## AIC: 342.04  
##   
## Number of Fisher Scoring iterations: 4

Backward Stepwise:

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

## Start: AIC=256.16  
## violator ~ male + race + age + state + time.served + max.sentence +   
## multiple.offenses + crime  
##   
## Df Deviance AIC  
## - time.served 1 230.22 254.22  
## - crime 3 235.30 255.30  
## - male 1 231.41 255.41  
## <none> 230.16 256.16  
## - max.sentence 1 232.46 256.46  
## - age 1 234.09 258.09  
## - race 1 236.97 260.97  
## - multiple.offenses 1 248.67 272.67  
## - state 3 304.40 324.40  
##   
## Step: AIC=254.22  
## violator ~ male + race + age + state + max.sentence + multiple.offenses +   
## crime  
##   
## Df Deviance AIC  
## - crime 3 235.38 253.38  
## - male 1 231.56 253.56  
## <none> 230.22 254.22  
## - max.sentence 1 232.50 254.50  
## - age 1 234.09 256.09  
## - race 1 236.97 258.98  
## - multiple.offenses 1 249.39 271.39  
## - state 3 304.94 322.95  
##   
## Step: AIC=253.38  
## violator ~ male + race + age + state + max.sentence + multiple.offenses  
##   
## Df Deviance AIC  
## - male 1 236.28 252.28  
## <none> 235.38 253.38  
## - max.sentence 1 237.41 253.41  
## - age 1 238.26 254.26  
## - race 1 242.32 258.32  
## - multiple.offenses 1 255.31 271.31  
## - state 3 309.30 321.30  
##   
## Step: AIC=252.28  
## violator ~ race + age + state + max.sentence + multiple.offenses  
##   
## Df Deviance AIC  
## <none> 236.28 252.28  
## - max.sentence 1 238.31 252.31  
## - age 1 238.81 252.81  
## - race 1 243.44 257.44  
## - multiple.offenses 1 256.39 270.39  
## - state 3 309.81 319.80

summary(backmod)

##   
## Call:  
## glm(formula = violator ~ race + age + state + max.sentence +   
## multiple.offenses, family = "binomial", data = train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.7414 -0.3643 -0.2668 -0.1502 2.7714   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -4.60426 1.13428 -4.059 4.92e-05 \*\*\*  
## raceotherwise 1.07386 0.40527 2.650 0.00806 \*\*   
## age 0.02636 0.01660 1.588 0.11224   
## stateKentucky -0.41360 0.54930 -0.753 0.45147   
## stateLouisiana 0.86000 0.51900 1.657 0.09751 .   
## stateVirginia -3.34208 0.62057 -5.386 7.22e-08 \*\*\*  
## max.sentence 0.07733 0.05475 1.412 0.15788   
## multiple.offensesmultiple 1.77974 0.41476 4.291 1.78e-05 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 340.04 on 472 degrees of freedom  
## Residual deviance: 236.28 on 465 degrees of freedom  
## AIC: 252.28  
##   
## Number of Fisher Scoring iterations: 6

Forward Stepwise:

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

## Start: AIC=342.04  
## violator ~ 1  
##   
## Df Deviance AIC  
## + state 3 264.58 272.58  
## + max.sentence 1 321.79 325.79  
## + multiple.offenses 1 331.50 335.50  
## + race 1 335.64 339.64  
## + time.served 1 336.02 340.02  
## <none> 340.04 342.04  
## + age 1 338.27 342.27  
## + crime 3 334.34 342.34  
## + male 1 339.78 343.78  
##   
## Step: AIC=272.58  
## violator ~ state  
##   
## Df Deviance AIC  
## + multiple.offenses 1 246.88 256.88  
## + race 1 259.14 269.14  
## + age 1 262.48 272.48  
## <none> 264.58 272.58  
## + crime 3 259.43 273.43  
## + male 1 263.58 273.58  
## + time.served 1 264.29 274.29  
## + max.sentence 1 264.49 274.49  
##   
## Step: AIC=256.88  
## violator ~ state + multiple.offenses  
##   
## Df Deviance AIC  
## + race 1 240.42 252.42  
## <none> 246.88 256.88  
## + age 1 245.01 257.01  
## + max.sentence 1 245.58 257.58  
## + male 1 246.13 258.13  
## + time.served 1 246.88 258.88  
## + crime 3 242.93 258.93  
##   
## Step: AIC=252.42  
## violator ~ state + multiple.offenses + race  
##   
## Df Deviance AIC  
## + age 1 238.31 252.31  
## <none> 240.42 252.42  
## + max.sentence 1 238.81 252.81  
## + male 1 239.85 253.85  
## + time.served 1 240.37 254.37  
## + crime 3 236.69 254.69  
##   
## Step: AIC=252.31  
## violator ~ state + multiple.offenses + race + age  
##   
## Df Deviance AIC  
## + max.sentence 1 236.28 252.28  
## <none> 238.31 252.31  
## + male 1 237.41 253.41  
## + crime 3 233.88 253.88  
## + time.served 1 238.18 254.18  
##   
## Step: AIC=252.28  
## violator ~ state + multiple.offenses + race + age + max.sentence  
##   
## Df Deviance AIC  
## <none> 236.28 252.28  
## + male 1 235.38 253.38  
## + crime 3 231.56 253.56  
## + time.served 1 236.12 254.12

summary(forwardmod)

##   
## Call:  
## glm(formula = violator ~ state + multiple.offenses + race + age +   
## max.sentence, family = "binomial", data = train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.7414 -0.3643 -0.2668 -0.1502 2.7714   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -4.60426 1.13428 -4.059 4.92e-05 \*\*\*  
## stateKentucky -0.41360 0.54930 -0.753 0.45147   
## stateLouisiana 0.86000 0.51900 1.657 0.09751 .   
## stateVirginia -3.34208 0.62057 -5.386 7.22e-08 \*\*\*  
## multiple.offensesmultiple 1.77974 0.41476 4.291 1.78e-05 \*\*\*  
## raceotherwise 1.07386 0.40527 2.650 0.00806 \*\*   
## age 0.02636 0.01660 1.588 0.11224   
## max.sentence 0.07733 0.05475 1.412 0.15788   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 340.04 on 472 degrees of freedom  
## Residual deviance: 236.28 on 465 degrees of freedom  
## AIC: 252.28  
##   
## Number of Fisher Scoring iterations: 6

We see our backward and forward stepwise model returned the same results. We have our significant variables that include: state, race, age, max.sentence, and multiple.offenses. All of the coefficiencts match what we would expect and we have an AIC of 252.28. With this lower AIC value we would think this is a good model and it makes sense intuitively.

Logistic Regression Model with Variables: state, race, multiple.offenses

mod2 = glm(violator ~ state + multiple.offenses + race, train, family = "binomial")  
summary(mod2)

##   
## Call:  
## glm(formula = violator ~ state + multiple.offenses + race, family = "binomial",   
## data = train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.4553 -0.3862 -0.2931 -0.1787 2.8791   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -2.5582 0.3709 -6.898 5.28e-12 \*\*\*  
## stateKentucky -0.4816 0.5417 -0.889 0.3740   
## stateLouisiana 0.5292 0.4769 1.110 0.2672   
## stateVirginia -3.2301 0.6028 -5.358 8.39e-08 \*\*\*  
## multiple.offensesmultiple 1.6596 0.3985 4.165 3.12e-05 \*\*\*  
## raceotherwise 1.0024 0.3966 2.528 0.0115 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 340.04 on 472 degrees of freedom  
## Residual deviance: 240.42 on 467 degrees of freedom  
## AIC: 252.42  
##   
## Number of Fisher Scoring iterations: 6

We can observe this model with an AIC = 252.42 to have a lower AIC than our previous logistic regression which bodes well for this model. The coefficients are behaving as one would expect and the significant variables follow along with our previous visualizations.

Predictions for 2 Parolees:

parolee1 = data.frame(state = "Kentucky", multiple.offenses = "multiple", race = "white")  
predict(mod2, parolee1, type = "response")

## 1   
## 0.2009733

parolee2 = data.frame(state = "Louisiana", multiple.offenses = "other", race = "otherwise")  
predict(mod2, parolee2, type = "response")

## 1   
## 0.2637466

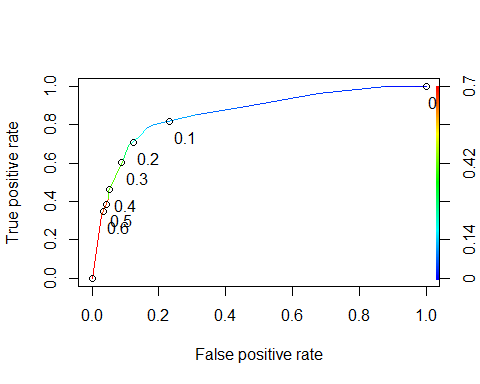
After running these two predictions, we see Parolee 2 to have a higher chance of violating parole than Parolee 1.

Creating ROC Curve and Evaluating Performance:

predictions = predict(mod2, type="response")  
head(predictions)

## 1 3 4 5 6 7   
## 0.07187555 0.17425270 0.07187555 0.17425270 0.17425270 0.07187555

ROCpred = prediction(predictions, train$violator)  
  
  
ROCperf = performance(ROCpred, "tpr", "fpr")  
plot(ROCperf, colorize=TRUE, print.cutoffs.at=seq(0,1,by=0.1), text.adj=c(-0.2,1.7))



Area Under Curve:

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

## [1] 0.8586124

Accuracy, Sensitivity, and Specificity:

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(ROCperf, ROCpred))

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

Above we see our 3 values for sensitivity, specificity, and the cutoff for our graph.

Confusion Matrix to Analyze Accuracy:

t1 = table(train$violator, predictions > 0.1161882)  
t1

##   
## FALSE TRUE  
## other 357 61  
## violator 14 41

We have correctly classified 357 non violators as False and 41 violators as True but we missed the 61 and 14 others. These numbers come from our original observed cutoff value.

Calculating Accuracy:

(t1[1,1]+t1[2,2])/nrow(train)

## [1] 0.8414376

Our model accuracy initially is at .84. Now we can manipulate our cutoff point to find a better accuracy. Incorrectly classifying a parolee can give us either a bad representation of a parolee who would not normally be a risk for a parole violation or it wouldn’t show us a parolee who should be high risk potential and we miss out on this parole violation.

Threshold = 0.65 cutoff

t1 = table(train$violator, predictions > 0.65)  
t1

##   
## FALSE TRUE  
## other 406 12  
## violator 37 18

(t1[1,1]+t1[2,2])/nrow(train)

## [1] 0.8964059

Using a higher cutoff point gets us closer and closer to 0.9 until we reach a range not in our graphical representation. Thus for this model we have our highest accuract at 0.8964 at a cutoff of 0.65.