# Module 3: Assignment 2 – Classification with Logistic Regression

## Kellie McLiverty

### Assignment Needs & Data Importation

Libraries needed for Assignemnt

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)

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

##   
## Attaching package: 'MASS'

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

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(ROCR)

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

## Loading required package: gplots

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

##   
## Attaching package: 'gplots'

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

library(leaps)

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

library(e1071)

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

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()  
## )

### Converting data to factors

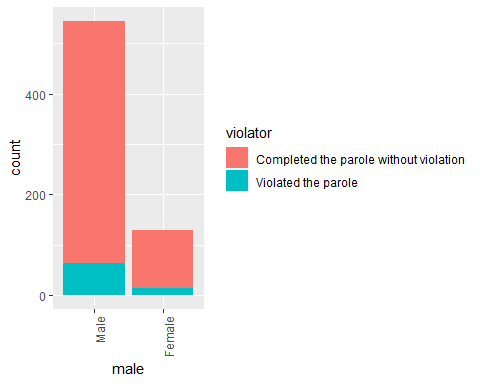
#converting data into male or female  
parole = parole %>% mutate(male = as\_factor(as.character(male))) %>%  
mutate(male = fct\_recode(male,  
"Female" = "0",  
"Male" = "1"  
))  
  
#converting race into White or otherwise  
parole = parole %>% mutate(race = as\_factor(as.character(race))) %>%  
mutate(race = fct\_recode(race,  
"White" = "1",  
"Otherwise" = "2"  
))  
  
#converting states  
parole = parole %>% mutate(state = as\_factor(as.character(state))) %>%  
mutate(state = fct\_recode(state,  
"Any Other State" = "1",  
"Kentucky" = "2",  
"Louisiana" = "3",  
"Virginia" = "4"  
))  
  
#converting Crimes  
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"  
))  
  
#converting Multiple Offenses  
parole = parole %>% mutate(multiple.offenses = as\_factor(as.character(multiple.offenses))) %>%  
mutate(multiple.offenses = fct\_recode(multiple.offenses,  
"Otherwise" = "0",  
"Incarcerated for multiple offenses" = "1"  
))  
  
#converting parole  
parole = parole %>% mutate(violator = as\_factor(as.character(violator))) %>%  
mutate(violator = fct\_recode(violator,  
"Completed the parole without violation" = "0",  
"Violated the parole" = "1"  
))

For this assignment, we’ll start by splitting the data into training and testing, using a random number seed of 1234.

set.seed(12345) #sets random number seed for cross validation  
train.rows = createDataPartition(y = parole$violator, p=0.7, list= FALSE)  
train = parole[train.rows,]  
test = parole[-train.rows,]

Visuals to see relationships in data

#Looking at the relationship of parole violators to sex  
ggplot(parole, aes(x=male, fill = violator)) + geom\_bar() +theme(axis.text.x = element\_text(angle = 90, hjust = 1))

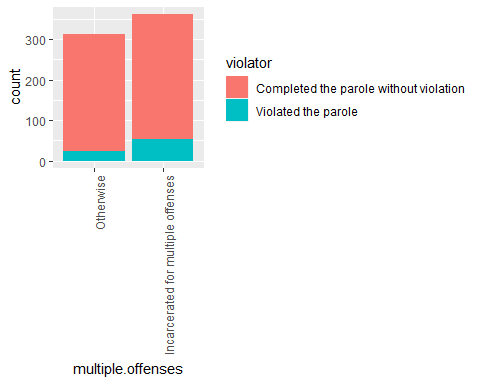


#table view  
tb = table(parole$violator, parole$male) #creates table object  
prop.table(tb, margin = 2) #crosstab with proportions

##   
## Male Female  
## Completed the parole without violation 0.8825688 0.8923077  
## Violated the parole 0.1174312 0.1076923

Looking at the tabluar data, there is not a significant difference in the percentage between males and females violating parole simply based on sex.

#Looking at the relationship of parole violators to multiple offenses  
ggplot(parole, aes(x=multiple.offenses, fill = violator)) + geom\_bar() + theme(axis.text.x = element\_text(angle = 90, hjust = 1))

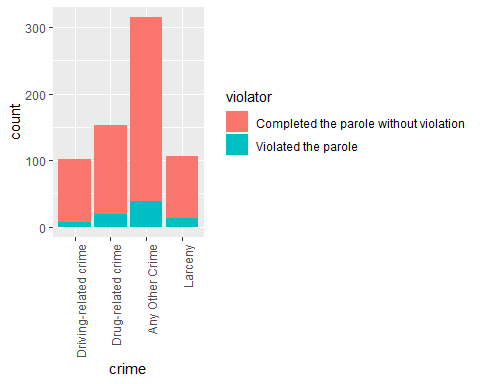


#table view  
tb1 = table(parole$violator, parole$multiple.offenses) #creates table object  
prop.table(tb1, margin = 2) #crosstab with proportions

##   
## Otherwise  
## Completed the parole without violation 0.9201278  
## Violated the parole 0.0798722  
##   
## Incarcerated for multiple offenses  
## Completed the parole without violation 0.8535912  
## Violated the parole 0.1464088

The number of offenses a parolee has does appear to have an impact on their liklihood to violate parole. Those incarcerated for multiple offenses are more likely to violate parole, and may be the most significant indicator of parole violation.

#Looking at the relationship of parole violators to type of crime   
ggplot(parole, aes(x=crime, fill = violator)) + geom\_bar() + theme(axis.text.x = element\_text(angle = 90, hjust = 1))

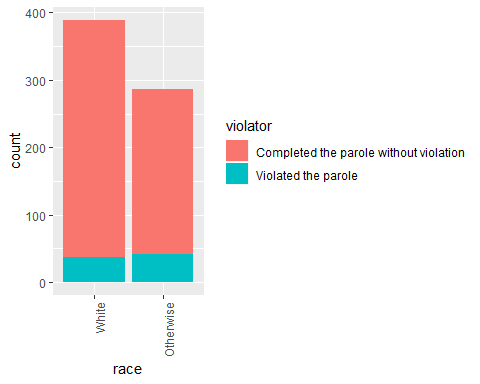


#table view  
tb2 = table(parole$violator, parole$crime) #creates table object  
prop.table(tb2, margin = 2) #crosstab with proportions

##   
## Driving-related crime  
## Completed the parole without violation 0.93069307  
## Violated the parole 0.06930693  
##   
## Drug-related crime  
## Completed the parole without violation 0.87581699  
## Violated the parole 0.12418301  
##   
## Any Other Crime Larceny  
## Completed the parole without violation 0.87619048 0.87735849  
## Violated the parole 0.12380952 0.12264151

Looking at this data, the type of crime does seems to have a slight impact on the number of people who violate parole, but possibly not as much as the number of offenses.

#Looking at the relationship of parole violators to race   
ggplot(parole, aes(x=race, fill = violator)) + geom\_bar() + theme(axis.text.x = element\_text(angle = 90, hjust = 1))

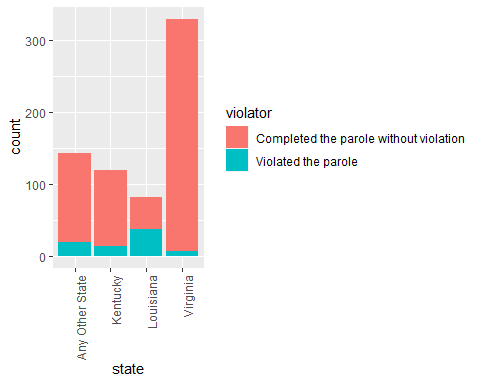


#table view  
tb3 = table(parole$violator, parole$race) #creates table object  
prop.table(tb3, margin = 2) #crosstab with proportions

##   
## White Otherwise  
## Completed the parole without violation 0.90488432 0.85664336  
## Violated the parole 0.09511568 0.14335664

Though it appears that the number of parolees that violated their parole is about equal between whites and other races, this could be a significant of predictor of parole violation but maybe not as significant as other factors.

#Looking at the relationship of parole violators to state   
ggplot(parole, aes(x=state, fill = violator)) + geom\_bar() + theme(axis.text.x = element\_text(angle = 90, hjust = 1))



#table view  
tb4 = table(parole$violator, parole$state) #creates table object  
prop.table(tb4, margin = 2) #crosstab with proportions

##   
## Any Other State Kentucky  
## Completed the parole without violation 0.86013986 0.88333333  
## Violated the parole 0.13986014 0.11666667  
##   
## Louisiana Virginia  
## Completed the parole without violation 0.54878049 0.97878788  
## Violated the parole 0.45121951 0.02121212

Looking at this data alone, I would say the state could be a significant predictor if a person will violate their parole. It is interesting to note that Lousiana had a higher rate a parole violation than the other states, this may be an indication of significance to me.

Next, we’ll build a model with offenses.

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

##   
## Call:  
## glm(formula = violator ~ multiple.offenses, family = "binomial",   
## data = parole)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -0.5627 -0.5627 -0.4080 -0.4080 2.2483   
##   
## Coefficients:  
## Estimate Std. Error  
## (Intercept) -2.4441 0.2085  
## multiple.offensesIncarcerated for multiple offenses 0.6810 0.2561  
## z value Pr(>|z|)   
## (Intercept) -11.722 < 2e-16 \*\*\*  
## multiple.offensesIncarcerated for multiple offenses 2.659 0.00783 \*\*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 483.27 on 674 degrees of freedom  
## Residual deviance: 475.81 on 673 degrees of freedom  
## AIC: 479.81  
##   
## Number of Fisher Scoring iterations: 5

From this regression model, we can see that multiple offenses is a highly significant predictor of parole violation. This model has a low AIC value of 479.81 and significant p value for otherwise & multiple incarcerations.

With this in mind, we’ll create the best fitting model to predict “violator”.

allmod = glm(violator ~., parole, family = "binomial") #creates the full model  
summary(allmod)

##   
## Call:  
## glm(formula = violator ~ ., family = "binomial", data = parole)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.6633 -0.4123 -0.2574 -0.1589 2.8738   
##   
## Coefficients:  
## Estimate Std. Error  
## (Intercept) -3.182313 1.041740  
## maleFemale -0.270624 0.370506  
## raceOtherwise 0.757252 0.324581  
## age 0.006554 0.013724  
## stateKentucky 0.208399 0.417528  
## stateLouisiana 0.893812 0.447042  
## stateVirginia -3.280842 0.526952  
## time.served -0.076548 0.099531  
## max.sentence 0.053293 0.043824  
## multiple.offensesIncarcerated for multiple offenses 1.531547 0.325794  
## crimeDrug-related crime -0.123182 0.542514  
## crimeAny Other Crime 0.157812 0.484705  
## crimeLarceny 0.494705 0.584656  
## z value Pr(>|z|)   
## (Intercept) -3.055 0.00225 \*\*   
## maleFemale -0.730 0.46513   
## raceOtherwise 2.333 0.01965 \*   
## age 0.478 0.63297   
## stateKentucky 0.499 0.61769   
## stateLouisiana 1.999 0.04557 \*   
## stateVirginia -6.226 4.78e-10 \*\*\*  
## time.served -0.769 0.44184   
## max.sentence 1.216 0.22396   
## multiple.offensesIncarcerated for multiple offenses 4.701 2.59e-06 \*\*\*  
## crimeDrug-related crime -0.227 0.82038   
## crimeAny Other Crime 0.326 0.74474   
## crimeLarceny 0.846 0.39747   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 483.27 on 674 degrees of freedom  
## Residual deviance: 348.68 on 662 degrees of freedom  
## AIC: 374.68  
##   
## Number of Fisher Scoring iterations: 6

emptymod = glm(violator~1, parole, family = "binomial") #creates the empty model  
summary(emptymod)

##   
## Call:  
## glm(formula = violator ~ 1, family = "binomial", data = parole)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -0.4956 -0.4956 -0.4956 -0.4956 2.0775   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -2.0352 0.1204 -16.9 <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: 483.27 on 674 degrees of freedom  
## Residual deviance: 483.27 on 674 degrees of freedom  
## AIC: 485.27  
##   
## Number of Fisher Scoring iterations: 4

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

## Start: AIC=374.68  
## violator ~ male + race + age + state + time.served + max.sentence +   
## multiple.offenses + crime  
##   
## Df Deviance AIC  
## - crime 3 350.44 370.44  
## - age 1 348.91 372.91  
## - male 1 349.23 373.23  
## - time.served 1 349.27 373.27  
## - max.sentence 1 350.17 374.17  
## <none> 348.68 374.68  
## - race 1 354.07 378.07  
## - multiple.offenses 1 371.79 395.79  
## - state 3 450.74 470.74  
##   
## Step: AIC=370.44  
## violator ~ male + race + age + state + time.served + max.sentence +   
## multiple.offenses  
##   
## Df Deviance AIC  
## - age 1 350.60 368.60  
## - male 1 350.77 368.77  
## - time.served 1 351.06 369.06  
## - max.sentence 1 351.91 369.91  
## <none> 350.44 370.44  
## - race 1 355.64 373.64  
## - multiple.offenses 1 374.06 392.06  
## - state 3 453.72 467.72  
##   
## Step: AIC=368.6  
## violator ~ male + race + state + time.served + max.sentence +   
## multiple.offenses  
##   
## Df Deviance AIC  
## - male 1 350.93 366.93  
## - time.served 1 351.17 367.17  
## - max.sentence 1 352.03 368.03  
## <none> 350.60 368.60  
## - race 1 355.79 371.79  
## - multiple.offenses 1 374.18 390.18  
## - state 3 453.86 465.86  
##   
## Step: AIC=366.93  
## violator ~ race + state + time.served + max.sentence + multiple.offenses  
##   
## Df Deviance AIC  
## - time.served 1 351.62 365.62  
## - max.sentence 1 352.43 366.43  
## <none> 350.93 366.93  
## - race 1 356.19 370.19  
## - multiple.offenses 1 374.52 388.52  
## - state 3 453.97 463.97  
##   
## Step: AIC=365.62  
## violator ~ race + state + max.sentence + multiple.offenses  
##   
## Df Deviance AIC  
## - max.sentence 1 353.26 365.26  
## <none> 351.62 365.62  
## - race 1 356.73 368.73  
## - multiple.offenses 1 376.00 388.00  
## - state 3 455.73 463.73  
##   
## Step: AIC=365.26  
## violator ~ race + state + multiple.offenses  
##   
## Df Deviance AIC  
## <none> 353.26 365.26  
## - race 1 358.69 368.69  
## - multiple.offenses 1 376.71 386.71  
## - state 3 473.73 479.73

summary(backmod)

##   
## Call:  
## glm(formula = violator ~ race + state + multiple.offenses, family = "binomial",   
## data = parole)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.4012 -0.4051 -0.2604 -0.1801 2.8739   
##   
## Coefficients:  
## Estimate Std. Error  
## (Intercept) -2.50359 0.30055  
## raceOtherwise 0.74594 0.31828  
## stateKentucky 0.04449 0.39449  
## stateLouisiana 0.75016 0.39147  
## stateVirginia -3.12945 0.51147  
## multiple.offensesIncarcerated for multiple offenses 1.51964 0.32027  
## z value Pr(>|z|)   
## (Intercept) -8.330 < 2e-16 \*\*\*  
## raceOtherwise 2.344 0.0191 \*   
## stateKentucky 0.113 0.9102   
## stateLouisiana 1.916 0.0553 .   
## stateVirginia -6.119 9.44e-10 \*\*\*  
## multiple.offensesIncarcerated for multiple offenses 4.745 2.09e-06 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 483.27 on 674 degrees of freedom  
## Residual deviance: 353.26 on 669 degrees of freedom  
## AIC: 365.26  
##   
## Number of Fisher Scoring iterations: 6

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

## Start: AIC=485.27  
## violator ~ 1  
##   
## Df Deviance AIC  
## + state 3 382.89 390.89  
## + max.sentence 1 465.68 469.68  
## + multiple.offenses 1 475.81 479.81  
## + time.served 1 477.05 481.05  
## + race 1 479.56 483.56  
## <none> 483.27 485.27  
## + male 1 483.17 487.17  
## + age 1 483.25 487.25  
## + crime 3 480.48 488.48  
##   
## Step: AIC=390.89  
## violator ~ state  
##   
## Df Deviance AIC  
## + multiple.offenses 1 358.69 368.69  
## + race 1 376.71 386.71  
## <none> 382.89 390.89  
## + time.served 1 381.65 391.65  
## + max.sentence 1 381.93 391.93  
## + male 1 382.16 392.16  
## + age 1 382.87 392.87  
## + crime 3 380.87 394.87  
##   
## Step: AIC=368.69  
## violator ~ state + multiple.offenses  
##   
## Df Deviance AIC  
## + race 1 353.26 365.26  
## <none> 358.69 368.69  
## + max.sentence 1 356.73 368.73  
## + time.served 1 358.02 370.02  
## + male 1 358.04 370.04  
## + age 1 358.64 370.64  
## + crime 3 357.47 373.47  
##   
## Step: AIC=365.26  
## violator ~ state + multiple.offenses + race  
##   
## Df Deviance AIC  
## <none> 353.26 365.26  
## + max.sentence 1 351.62 365.62  
## + time.served 1 352.43 366.43  
## + male 1 352.71 366.71  
## + age 1 353.20 367.20  
## + crime 3 351.81 369.81

summary(forwardmod)

##   
## Call:  
## glm(formula = violator ~ state + multiple.offenses + race, family = "binomial",   
## data = parole)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.4012 -0.4051 -0.2604 -0.1801 2.8739   
##   
## Coefficients:  
## Estimate Std. Error  
## (Intercept) -2.50359 0.30055  
## stateKentucky 0.04449 0.39449  
## stateLouisiana 0.75016 0.39147  
## stateVirginia -3.12945 0.51147  
## multiple.offensesIncarcerated for multiple offenses 1.51964 0.32027  
## raceOtherwise 0.74594 0.31828  
## z value Pr(>|z|)   
## (Intercept) -8.330 < 2e-16 \*\*\*  
## stateKentucky 0.113 0.9102   
## stateLouisiana 1.916 0.0553 .   
## stateVirginia -6.119 9.44e-10 \*\*\*  
## multiple.offensesIncarcerated for multiple offenses 4.745 2.09e-06 \*\*\*  
## raceOtherwise 2.344 0.0191 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 483.27 on 674 degrees of freedom  
## Residual deviance: 353.26 on 669 degrees of freedom  
## AIC: 365.26  
##   
## Number of Fisher Scoring iterations: 6

Looking at the coefficents and AIC values [AIC=365.26] of both of these models, it shows that both models are the same, so I will choose to use the Forward stepwise from here. It also confirms and earlier assumption that State and Race (otherwise) are also significant predictors of parole violation.

This model of training data shows that the state in which a parolee is in, if they are a multiple offender, and if they are a race other than white all have a significant impact on their likelihood of violating parole. However, when looking at the significance of the model regarding state, it is important to note that only the state of Kentucky has no significant impact on a parolee’s likelihood to violate parole.

Based on the graphs from above these seem like intuitive predictors for parole violations, so we can predict a non-white parolee from Lousiana who has multiple offenses is more likely to violate parole than their counterpart in Virginia. But let’s continue to look at the data to be sure.

Next, we’ll build a logistic regression model for violator based on these three variables.

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

##   
## Call:  
## glm(formula = violator ~ multiple.offenses + race + state, 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  
## (Intercept) -2.5582 0.3709  
## multiple.offensesIncarcerated for multiple offenses 1.6596 0.3985  
## raceOtherwise 1.0024 0.3966  
## stateKentucky -0.4816 0.5417  
## stateLouisiana 0.5292 0.4769  
## stateVirginia -3.2301 0.6028  
## z value Pr(>|z|)   
## (Intercept) -6.898 5.28e-12 \*\*\*  
## multiple.offensesIncarcerated for multiple offenses 4.165 3.12e-05 \*\*\*  
## raceOtherwise 2.528 0.0115 \*   
## stateKentucky -0.889 0.3740   
## stateLouisiana 1.110 0.2672   
## stateVirginia -5.358 8.39e-08 \*\*\*  
## ---  
## 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

When working with only these variables in our training dataset, we can see that our AIC level decreased to 252.42, which is a good indicator for the quality of this model. However, if we look at the significance scores of this dataset, we can see that the state of Louisiana has lost it’s significance as reported in the stepwise models. Race and Multiple Offenses still remains very significant factors for parole violation.

Now let’s test if my earlier predictions were correct making predictions our mod2 that uses the training data. For a parolee who was incarcerated in Louisiana with multiple offenses and is white race, we can predict they would have a 40.87% chance of violating parole.

#parolee number 1  
newdata = data.frame(state = "Louisiana", multiple.offenses = "Incarcerated for multiple offenses", race = "White")  
predict(mod2, newdata, type="response")

## 1   
## 0.408682

Another parolee from Kentucky with no multiple offenses who is of any other race we can predict they would have a only a 11.53% chance of violating parole.

#parolee number 2  
newdata = data.frame(state = "Kentucky", multiple.offenses = "Otherwise", race = "Otherwise")  
predict(mod2, newdata, type="response")

## 1   
## 0.1153326

Without a threshold to tell us exactly where our cut off is for predicting parole violations, we can assume that the first parolee will violate parole, since his/her chances are over 50%. But that’s not always an accurate presumption for our data. To get a better idea, we need to create a threshold for our predictions.

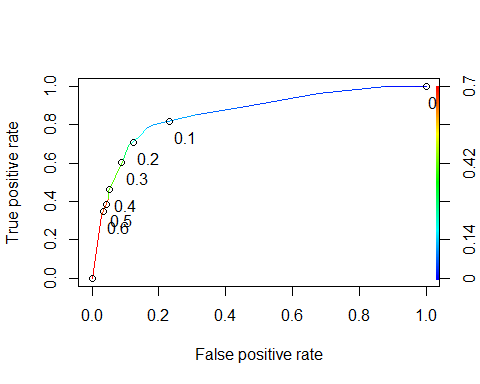
To begin, let’s apply ROCR to create a ROC curve to help us find our probability threshold.

predictions = predict(mod2, type="response") #develop predicted probabilities  
head(predictions)

## 1 2 3 4 5 6   
## 0.07187555 0.17425270 0.07187555 0.17425270 0.17425270 0.07187555

Threshold selection

#Determining the threshold in graph  
ROCRpred = prediction(predictions, train$violator)   
  
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))



#Determine threshold to balance sensitivity and specificity  
#DO NOT modify this code  
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

Examining this data in the threshld for our mod2, we can see that we have a cutoff value of .1161882, but is it really accurate? To know for sure, we’ll have to test its accuracy next.

Test thresholds to evaluate accuracy

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

##   
## FALSE TRUE  
## Completed the parole without violation 357 61  
## Violated the parole 14 41

Reviewing our confusion matrix, we correctly classified 357 completed parole without violation and 41 that violated parole. We missed 61 who were incorrecly classified as completing parole without violation and 14 that did not violate parole.

Calculate accuracy

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

## [1] 0.8414376

Looking at the accuracy of our model at 0.8414376, we can say if we choose to balance sensitivity, and specificity we will have accuracy of 0.84. There are some very negative implications in incorrecly classifying a parolee that could end up putting them back in jail, when they did in fact not violate their parole.

Knowing this, we can apply trial and error to maximize accuracy

t1 = table(train$violator,predictions > 0.5) #let's use 0.5 for our threshold  
t1

##   
## FALSE TRUE  
## Completed the parole without violation 406 12  
## Violated the parole 37 18

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

## [1] 0.8964059

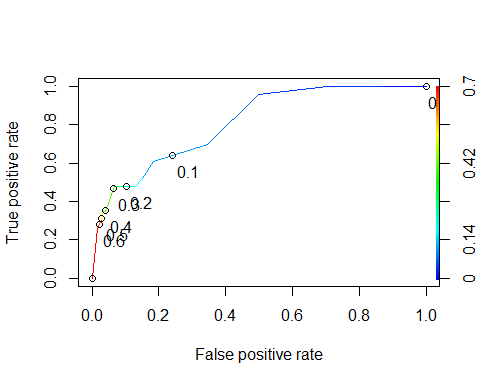
At a threshold of 0.5, our accuracy increases slightly to 0.8964059, which is roughly 0.9. A threshold of 0.6 delivered the same accuracy.  
Finally, we want to test this threshold on a naive prediction for the testing set.

#develop predicted probabilities  
newdata = data.frame(test)  
  
predictions = predict(mod2, newdata, type="response")   
head(predictions)

## 1 2 3 4 5 6   
## 0.07187555 0.52593528 0.07187555 0.07187555 0.07187555 0.07187555

Threshold selection

#Determining the threshold in graph  
ROCRpred = prediction(predictions, test$violator)   
  
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))



#Determine threshold to balance sensitivity and specificity  
#DO NOT modify this code  
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 0.6086957  
## specificity 0.8156425  
## cutoff 0.1153326

This returned us a cutoff value of 0.1153326. Finally, we apply the threshold from our training data to the test data.

t2 = table(test$violator,predictions > 0.5)  
t2

##   
## FALSE TRUE  
## Completed the parole without violation 174 5  
## Violated the parole 16 7

(t2[1,1]+t2[2,2])/nrow(test)

## [1] 0.8960396

For the testing data model, at a threshold of 0.5, we can expect to have accuracy of 0.8960396. Which is pretty accurare. Out of the 202 obervations in the dataset, we correctly classified 174 who did not violate parole and 7 that did violate parole.

In this dataset, we can see fewer incorrectly classified parolees with 16 incorrecly classified as parole violators and 5 as not parole violators.