## Week3 Assignment 2

### DEY, Sankha

#### Logistic Regression (Classification)

# message=FALSE done before last knit  
library(tidyverse)  
library(MASS)  
library(caret)  
library(ROCR)

parole <- 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))) %>% #Convert Male  
mutate(male = fct\_recode(male,  
"Male" = "1",  
"Female" = "0")) %>%  
 mutate(race = as\_factor(as.character(race))) %>% #Convert Race  
mutate(race = fct\_recode(race,  
"White" = "1",  
"Others" = "2")) %>%  
 mutate(state = as\_factor(as.character(state))) %>% #Convert State  
mutate(state = fct\_recode(state,  
"Kentucky" = "2",  
"Louisiana" = "3",  
"Virginia" = "4",  
"Others" = "1")) %>%  
 mutate(crime = as\_factor(as.character(crime))) %>% #Convert Crime  
mutate(crime = fct\_recode(crime,  
"Larceny" = "2",  
"Drug" = "3",  
"Driving" = "4",  
"Others" = "1")) %>%  
 mutate(multiple.offenses = as\_factor(as.character(multiple.offenses))) %>% #Convert Offences  
mutate(multiple.offenses = fct\_recode(multiple.offenses,  
"Multiple" = "1",  
"Others" = "0")) %>%  
 mutate(violator = as\_factor(as.character(violator))) %>% #Convert Violators  
mutate(violator = fct\_recode(violator,  
"Violation" = "1",  
"No Violation" = "0"))  
str(parole)

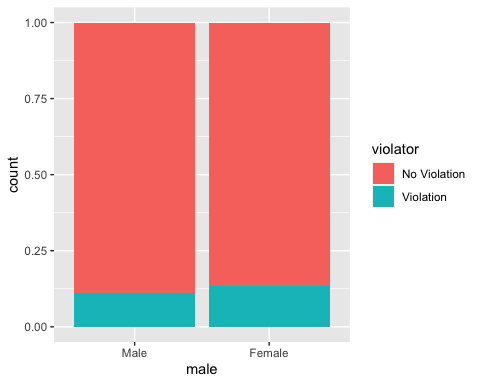
## Classes 'spec\_tbl\_df', 'tbl\_df', 'tbl' and 'data.frame': 675 obs. of 9 variables:  
## $ male : Factor w/ 2 levels "Male","Female": 1 2 1 1 1 1 1 2 2 1 ...  
## $ race : Factor w/ 2 levels "White","Others": 1 1 2 1 2 2 1 1 1 2 ...  
## $ age : num 33.2 39.7 29.5 22.4 21.6 46.7 31 24.6 32.6 29.1 ...  
## $ state : Factor w/ 4 levels "Others","Kentucky",..: 1 1 1 1 1 1 1 1 1 1 ...  
## $ time.served : num 5.5 5.4 5.6 5.7 5.4 6 6 4.8 4.5 4.7 ...  
## $ max.sentence : num 18 12 12 18 12 18 18 12 13 12 ...  
## $ multiple.offenses: Factor w/ 2 levels "Others","Multiple": 1 1 1 1 1 1 1 1 1 1 ...  
## $ crime : Factor w/ 4 levels "Driving","Drug",..: 1 2 2 3 3 1 2 3 2 4 ...  
## $ violator : Factor w/ 2 levels "No Violation",..: 1 1 1 1 1 1 1 1 1 1 ...

#### Task 1

set.seed(12345) #set random number seed for cross validation  
trainrows = createDataPartition(y = parole$violator, p=0.7, list = FALSE) #70% in training  
train = slice(parole,trainrows)  
test = slice(parole,-trainrows)

#### Task 2

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

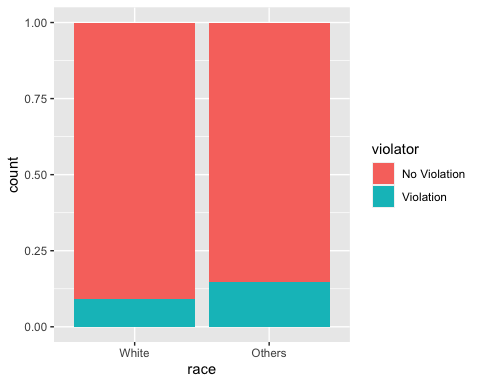


t1 = table(train$violator,train$male)  
prop.table(t1, margin = 2)

##   
## Male Female  
## No Violation 0.8880000 0.8673469  
## Violation 0.1120000 0.1326531

Female percentage is slightly higher on deciding parole vaiolator. But no apparent significant difference.

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



t2 = table(train$violator,train$race)  
prop.table(t2, margin = 2)

##   
## White Others  
## No Violation 0.90774908 0.85148515  
## Violation 0.09225092 0.14851485

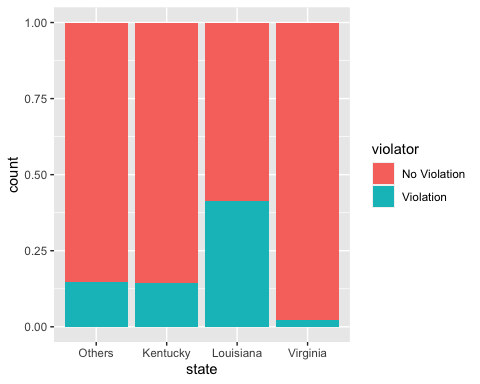
Other race (~15%) is higher than Whites (~9%) in deciding the potential violator.

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



Higher age seems to have more violators. But hard to decide with outliers.

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

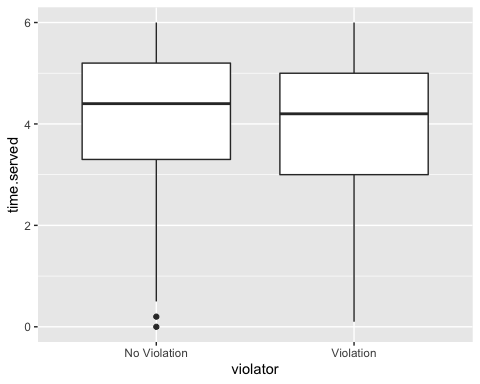


t3 = table(train$violator,train$state)  
prop.table(t3, margin = 2)

##   
## Others Kentucky Louisiana Virginia  
## No Violation 0.85263158 0.85542169 0.58620690 0.97890295  
## Violation 0.14736842 0.14457831 0.41379310 0.02109705

Louisiana has more violators (~41%) than other states. Virgina has less violators (only ~2%). Definitely state is a major contribuotr in predicting violators.

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



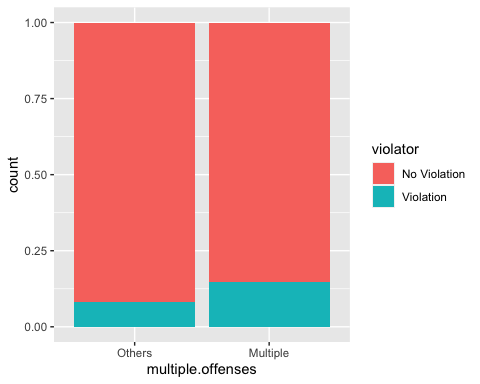
Less time served has stronger correlation with violation. No apparent significant difference.

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



No apparent significant difference.

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

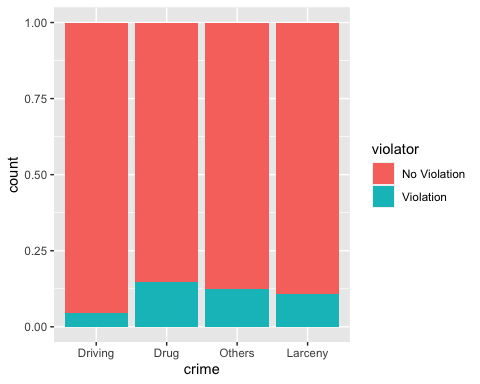


t4 = table(train$violator,train$multiple.offenses)  
prop.table(t4, margin = 2)

##   
## Others Multiple  
## No Violation 0.91981132 0.85440613  
## Violation 0.08018868 0.14559387

Multiple offence seems strongly linked with violation.

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



t5 = table(train$violator,train$crime)  
prop.table(t5, margin = 2)

##   
## Driving Drug Others Larceny  
## No Violation 0.95384615 0.85436893 0.87445887 0.89189189  
## Violation 0.04615385 0.14563107 0.12554113 0.10810811

Specific crime types have more violations. Drug related crimes have more violators than others.

#### Task 3

State will be the most predictive of violator.

mod21 = glm(violator ~ state , train, family = "binomial")  
summary(mod21)

##   
## Call:  
## glm(formula = violator ~ state, family = "binomial", data = train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.0335 -0.5589 -0.2065 -0.2065 2.7780   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -1.75539 0.28944 -6.065 1.32e-09 \*\*\*  
## stateKentucky -0.02238 0.42567 -0.053 0.958067   
## stateLouisiana 1.40709 0.39351 3.576 0.000349 \*\*\*  
## stateVirginia -2.08191 0.53672 -3.879 0.000105 \*\*\*  
## ---  
## 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: 275.18 on 469 degrees of freedom  
## AIC: 283.18  
##   
## Number of Fisher Scoring iterations: 6

AIC of this model (a measure of model quality) is 283.18. We can use this value to compare this model to others. Smaller AIC is better. Also, slope of stateLouisiana is positively correlated with violators, which makes sense.

#### Task 4

allmod32 = glm(violator ~., train, family = "binomial")   
summary(allmod32)

##   
## Call:  
## glm(formula = violator ~ ., family = "binomial", data = train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.6055 -0.3932 -0.2643 -0.1384 2.9470   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -3.750397 1.318165 -2.845 0.00444 \*\*   
## maleFemale 0.137577 0.411340 0.334 0.73803   
## raceOthers 1.143719 0.403890 2.832 0.00463 \*\*   
## age 0.005279 0.016910 0.312 0.75490   
## stateKentucky 0.124282 0.492370 0.252 0.80072   
## stateLouisiana 0.217202 0.556154 0.391 0.69614   
## stateVirginia -3.801561 0.666733 -5.702 1.19e-08 \*\*\*  
## time.served -0.109344 0.118901 -0.920 0.35777   
## max.sentence 0.065956 0.054593 1.208 0.22700   
## multiple.offensesMultiple 1.711032 0.396463 4.316 1.59e-05 \*\*\*  
## crimeDrug 0.516479 0.739095 0.699 0.48468   
## crimeOthers 0.727043 0.690775 1.053 0.29257   
## crimeLarceny 1.119953 0.797552 1.404 0.16025   
## ---  
## 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: 242.09 on 460 degrees of freedom  
## AIC: 268.09  
##   
## Number of Fisher Scoring iterations: 6

emptymod32 = glm(violator ~1, train, family = "binomial")   
summary(emptymod32)

##   
## 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

#backward  
backmod32 = stepAIC(allmod32, direction = "backward", trace = TRUE)

## Start: AIC=268.09  
## violator ~ male + race + age + state + time.served + max.sentence +   
## multiple.offenses + crime  
##   
## Df Deviance AIC  
## - crime 3 244.47 264.47  
## - age 1 242.18 266.18  
## - male 1 242.20 266.20  
## - time.served 1 242.93 266.93  
## - max.sentence 1 243.57 267.57  
## <none> 242.09 268.09  
## - race 1 250.24 274.24  
## - multiple.offenses 1 261.96 285.96  
## - state 3 316.24 336.24  
##   
## Step: AIC=264.47  
## violator ~ male + race + age + state + time.served + max.sentence +   
## multiple.offenses  
##   
## Df Deviance AIC  
## - age 1 244.48 262.48  
## - male 1 244.85 262.85  
## - time.served 1 245.04 263.04  
## - max.sentence 1 246.00 264.00  
## <none> 244.47 264.47  
## - race 1 252.62 270.62  
## - multiple.offenses 1 265.46 283.46  
## - state 3 321.69 335.69  
##   
## Step: AIC=262.48  
## violator ~ male + race + state + time.served + max.sentence +   
## multiple.offenses  
##   
## Df Deviance AIC  
## - male 1 244.86 260.86  
## - time.served 1 245.04 261.04  
## - max.sentence 1 246.01 262.01  
## <none> 244.48 262.48  
## - race 1 252.65 268.65  
## - multiple.offenses 1 265.52 281.52  
## - state 3 322.14 334.14  
##   
## Step: AIC=260.86  
## violator ~ race + state + time.served + max.sentence + multiple.offenses  
##   
## Df Deviance AIC  
## - time.served 1 245.31 259.31  
## - max.sentence 1 246.33 260.33  
## <none> 244.86 260.86  
## - race 1 252.80 266.80  
## - multiple.offenses 1 265.93 279.93  
## - state 3 322.54 332.54  
##   
## Step: AIC=259.31  
## violator ~ race + state + max.sentence + multiple.offenses  
##   
## Df Deviance AIC  
## - max.sentence 1 246.98 258.98  
## <none> 245.31 259.31  
## - race 1 253.11 265.11  
## - multiple.offenses 1 266.89 278.89  
## - state 3 323.88 331.88  
##   
## Step: AIC=258.98  
## violator ~ race + state + multiple.offenses  
##   
## Df Deviance AIC  
## <none> 246.98 258.98  
## - race 1 254.96 264.96  
## - multiple.offenses 1 267.66 277.66  
## - state 3 332.93 338.93

summary(backmod32)

##   
## Call:  
## glm(formula = violator ~ race + state + multiple.offenses, family = "binomial",   
## data = train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.3609 -0.4094 -0.2705 -0.1575 2.9653   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -2.51087 0.36354 -6.907 4.96e-12 \*\*\*  
## raceOthers 1.09382 0.38974 2.807 0.00501 \*\*   
## stateKentucky 0.07372 0.46051 0.160 0.87282   
## stateLouisiana 0.10381 0.50018 0.208 0.83559   
## stateVirginia -3.60795 0.63788 -5.656 1.55e-08 \*\*\*  
## multiple.offensesMultiple 1.73482 0.39421 4.401 1.08e-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: 246.98 on 467 degrees of freedom  
## AIC: 258.98  
##   
## Number of Fisher Scoring iterations: 6

Forward stepwise

#forward  
forwardmod32 = stepAIC(emptymod32, direction = "forward", scope=list(upper=allmod32,lower=emptymod32), trace = TRUE)

## Start: AIC=342.04  
## violator ~ 1  
##   
## Df Deviance AIC  
## + state 3 275.18 283.18  
## + max.sentence 1 331.01 335.01  
## + multiple.offenses 1 335.02 339.02  
## + race 1 336.51 340.51  
## + time.served 1 336.61 340.61  
## <none> 340.04 342.04  
## + crime 3 335.07 343.07  
## + male 1 339.72 343.72  
## + age 1 339.95 343.95  
##   
## Step: AIC=283.18  
## violator ~ state  
##   
## Df Deviance AIC  
## + multiple.offenses 1 254.96 264.96  
## + race 1 267.66 277.66  
## <none> 275.18 283.18  
## + max.sentence 1 274.27 284.27  
## + time.served 1 274.44 284.44  
## + age 1 275.11 285.11  
## + male 1 275.13 285.13  
## + crime 3 271.72 285.72  
##   
## Step: AIC=264.96  
## violator ~ state + multiple.offenses  
##   
## Df Deviance AIC  
## + race 1 246.98 258.98  
## <none> 254.96 264.96  
## + max.sentence 1 253.11 265.11  
## + time.served 1 254.47 266.47  
## + male 1 254.91 266.91  
## + age 1 254.94 266.94  
## + crime 3 252.75 268.75  
##   
## Step: AIC=258.98  
## violator ~ state + multiple.offenses + race  
##   
## Df Deviance AIC  
## <none> 246.98 258.98  
## + max.sentence 1 245.31 259.31  
## + time.served 1 246.33 260.33  
## + male 1 246.78 260.78  
## + age 1 246.98 260.98  
## + crime 3 244.78 262.79

summary(forwardmod32)

##   
## Call:  
## glm(formula = violator ~ state + multiple.offenses + race, family = "binomial",   
## data = train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.3609 -0.4094 -0.2705 -0.1575 2.9653   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -2.51087 0.36354 -6.907 4.96e-12 \*\*\*  
## stateKentucky 0.07372 0.46051 0.160 0.87282   
## stateLouisiana 0.10381 0.50018 0.208 0.83559   
## stateVirginia -3.60795 0.63788 -5.656 1.55e-08 \*\*\*  
## multiple.offensesMultiple 1.73482 0.39421 4.401 1.08e-05 \*\*\*  
## raceOthers 1.09382 0.38974 2.807 0.00501 \*\*   
## ---  
## 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: 246.98 on 467 degrees of freedom  
## AIC: 258.98  
##   
## Number of Fisher Scoring iterations: 6

AIC for both forward and backward model is same = 258.98. Both models are the same. In both models, **state, multiple offences and race** are the signficant predicotrs for violators. Slope of stateLouisiana is higher than any other state, which is true. In the training set, we saw that Louisiana has ~45% cases of violators. However, stateVirginia is negatively correlated with violators. This doesn’t seem right. This might be an example of multicollinearity. Multiple offence and Other Race have positive slopes indicating they are strongly correlated with violators. Overall, this model seems to be good.

#### Task 5

mod22 = glm(violator ~ state+multiple.offenses+race , train, family = "binomial")  
summary(mod22)

##   
## Call:  
## glm(formula = violator ~ state + multiple.offenses + race, family = "binomial",   
## data = train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.3609 -0.4094 -0.2705 -0.1575 2.9653   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -2.51087 0.36354 -6.907 4.96e-12 \*\*\*  
## stateKentucky 0.07372 0.46051 0.160 0.87282   
## stateLouisiana 0.10381 0.50018 0.208 0.83559   
## stateVirginia -3.60795 0.63788 -5.656 1.55e-08 \*\*\*  
## multiple.offensesMultiple 1.73482 0.39421 4.401 1.08e-05 \*\*\*  
## raceOthers 1.09382 0.38974 2.807 0.00501 \*\*   
## ---  
## 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: 246.98 on 467 degrees of freedom  
## AIC: 258.98  
##   
## Number of Fisher Scoring iterations: 6

AIC value is same with forward and backward regresseion model. All models are the same. State, multiple.offenses and race have p-value less than 0.05. All three variables are significnt.

#### Task 6:

What is the predicted probability of parole violation of the two following parolees?

newdata1 = data.frame(state = "Louisiana", multiple.offenses = "Multiple", race = "White")  
predict(forwardmod32, newdata1, type="response")

## 1   
## 0.3379961

**Parolee1**: Louisiana with multiple offenses and white race = 33.8% probabilty

newdata2 = data.frame(state = "Kentucky", multiple.offenses = "Others", race = "Others")  
predict(forwardmod32, newdata2, type="response")

## 1   
## 0.2069629

**Parolee2**: Kentucky with no multiple offenses and other race = 20.7% probabilty

#### Task 7:

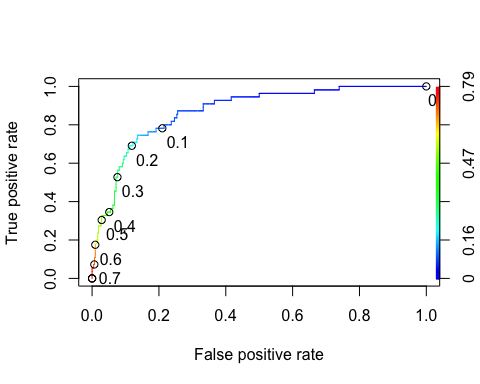
Develop an ROC curve and determine the probability threshold that best balances specificity and sensitivity (on the training set). Hint: In the predict function, use type = “response” and do not use the [,2] that we used in the logistic regression threshold lecture. We only had to include that code in that lecture because we used k-fold cross validation  
Develop predictions

predictions = predict(allmod32, type="response") #develop predicted probabilities  
head(predictions)

## 1 2 3 4 5 6   
## 0.04791065 0.14737696 0.13835686 0.07316025 0.07661207 0.25816041

Threshold selection

#Change this next line to the names of your predictions and the response variable in the training data frame  
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:

What is the accuracy, sensitivity, and specificity of the model on the training set given the cutoff from Task 7? What are the implications of incorrectly classifying a parolee?

#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 0.87272727  
## specificity 0.74401914  
## cutoff 0.08008453

Calculate accuracy/sensitivity/specificity

#confusion matrix  
t1 = table(train$violator,predictions > 0.08008453)  
t1

##   
## FALSE TRUE  
## No Violation 311 107  
## Violation 8 47

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

## [1] 0.756871

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

## [1] 0.8545455

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

## [1] 0.7440191

Accuracy -> 0.756871  
sensitivity -> 0.8545455

specificity -> 0.7440191

There might be multiple implications of incorrectly classifying a parolee. Let’s take an example of false-negative case, if a parolee is expected to be ‘non violator’,but he/she ends up being a potential violator. If such parolee is relased then there might be a potential risk to society and the associated cost would be much higher. In parole example, Sensitivity should be really high to detect the actual positive cases.

#### Task 9

Can apply trial and error to maximize accuracy (here trying 0.5 as threshold)

t1 = table(train$violator,predictions > 0.5)  
t1

##   
## FALSE TRUE  
## No Violation 406 12  
## Violation 39 16

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

## [1] 0.8921776

Threshold = 0.6

t1 = table(train$violator,predictions > 0.6)  
t1

##   
## FALSE TRUE  
## No Violation 414 4  
## Violation 46 9

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

## [1] 0.8942918

Threshold = 0.65

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

##   
## FALSE TRUE  
## No Violation 414 4  
## Violation 48 7

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

## [1] 0.8900634

So, accuracy is maximized when probability threshold >0.6. This dataset is a good example of imbalanced data too.

#### Task 10

# allmode for test data set  
allmodtest = glm(violator ~., test, family = "binomial")   
summary(allmodtest)

##   
## Call:  
## glm(formula = violator ~ ., family = "binomial", data = test)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.8012 -0.3548 -0.2056 -0.0971 2.5914   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -2.257250 2.144185 -1.053 0.29246   
## maleFemale -2.668315 1.224925 -2.178 0.02938 \*   
## raceOthers 0.258152 0.656863 0.393 0.69431   
## age 0.008419 0.027447 0.307 0.75905   
## stateKentucky 0.047867 0.982199 0.049 0.96113   
## stateLouisiana 2.564561 0.887951 2.888 0.00387 \*\*  
## stateVirginia -2.801571 0.949110 -2.952 0.00316 \*\*  
## time.served 0.067759 0.229225 0.296 0.76754   
## max.sentence 0.007561 0.089277 0.085 0.93251   
## multiple.offensesMultiple 1.448327 0.657212 2.204 0.02754 \*   
## crimeDrug -1.726431 1.060781 -1.628 0.10363   
## crimeOthers -1.152515 0.807116 -1.428 0.15331   
## crimeLarceny -0.994446 1.105671 -0.899 0.36844   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 143.223 on 201 degrees of freedom  
## Residual deviance: 89.119 on 189 degrees of freedom  
## AIC: 115.12  
##   
## Number of Fisher Scoring iterations: 6

#develop predicted probabilities for test data set  
predictionstest = predict(allmodtest, type="response")   
  
# Predictiontest for >0.6 (derived from train data set)  
tp = table(test$violator,predictionstest > 0.6)  
tp

##   
## FALSE TRUE  
## No Violation 177 2  
## Violation 14 9

(tp[1,1]+t1[2,2])/nrow(test)

## [1] 0.9108911

With probability threshold >0.6 (from train data set), accuracy of the model in test data set is 91.08%.