# Logistic Regression Classification

# Mod 3 - Assignment 2

# Cooper, Sarah

library(tidyverse)

library(MASS)

library(caret)

library(ROCR)

library(e1071)

parole <- read\_csv("C:/Users/Sarah/Downloads/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()  
## )

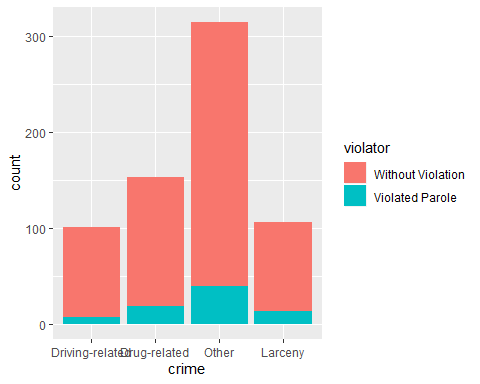
parole = parole %>% mutate(male = as\_factor(as.character(male))) %>%  
mutate(male = fct\_recode(male,  
"Male" = "1",  
"Female" = "0"))  
  
parole = parole %>% mutate(race = as\_factor(as.character(race))) %>%  
mutate(race = fct\_recode(race,  
"White" = "1",  
"Otherwise" = "2"))  
  
parole = parole %>% mutate(state = as\_factor(as.character(state))) %>%  
mutate(state = fct\_recode(state,  
"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 Offenses" = "1",  
"Otherwise" = "0",))  
  
parole = parole %>% mutate(violator = as\_factor(as.character(violator))) %>%  
mutate(violator = fct\_recode(violator,  
"Violated Parole" = "1",  
"Without Violation" = "0",))

# Task 1

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

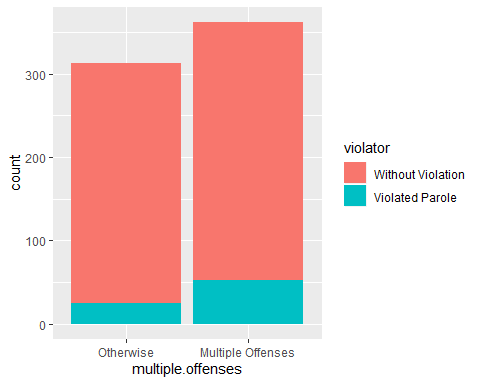
# Task 2

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



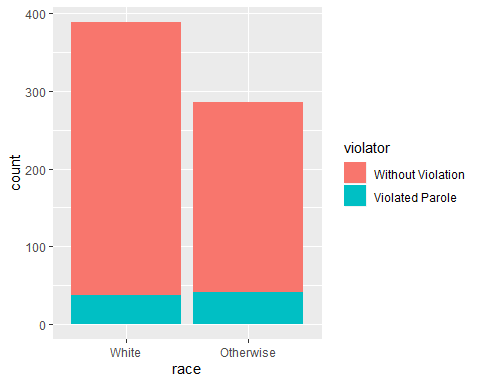
**My initial thought process in determining whether or not a parolee will violate their parole is to see what type of crime landed them in prison to begin with. I’m curious if the past offense can help us predict future behavior.**

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



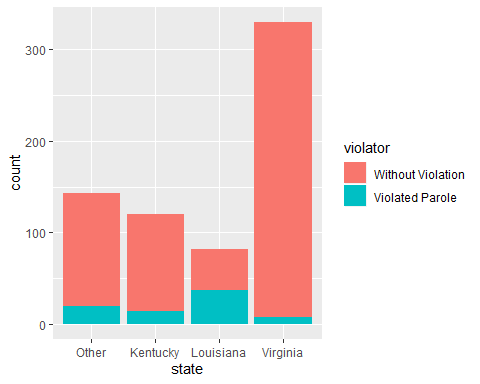
**I also wonder if the number of previous offenses may determine violation of parole or not.**

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



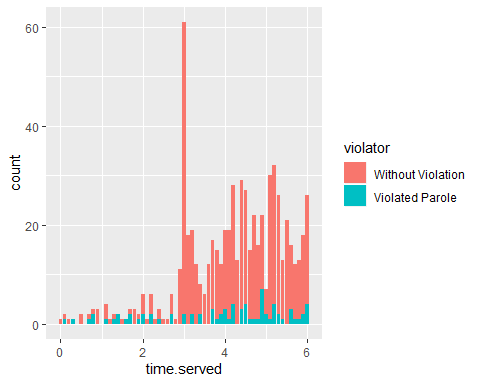
**Race most certainly doesn’t have an effect on determining parole.**

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



**There does appear to be a slight bump in Louisiana parole violations.**

ggplot(parole, aes(x=time.served, fill = violator)) + geom\_bar()



**My last inclination is that the more time served, the less likely the released inmate will violate parole. This graph doesn’t conclusively support that hunch.**

# Task 3

**According to the five graphs we ran above, the most likely indicator of parole violation could be state.**

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

##   
## 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

**Using the logistic regression model above, again, the state of Louisiana shows the most probability of a released inmate violating parole. The smaller AIC value of 283.18 indicates that this model is reliable.**

# Task 4

allmod = glm(violator ~ crime + multiple.offenses + max.sentence + time.served + state + age + race + male, train, family = "binomial")  
summary(allmod)

##   
## Call:  
## glm(formula = violator ~ crime + multiple.offenses + max.sentence +   
## time.served + state + age + race + male, 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 \*\*   
## crimeDrug-related 0.516479 0.739095 0.699 0.48468   
## crimeOther 0.727043 0.690775 1.053 0.29257   
## crimeLarceny 1.119953 0.797552 1.404 0.16025   
## multiple.offensesMultiple Offenses 1.711032 0.396463 4.316 1.59e-05 \*\*\*  
## max.sentence 0.065956 0.054593 1.208 0.22700   
## time.served -0.109344 0.118901 -0.920 0.35777   
## 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 \*\*\*  
## age 0.005279 0.016910 0.312 0.75490   
## raceOtherwise 1.143719 0.403890 2.832 0.00463 \*\*   
## maleFemale 0.137577 0.411340 0.334 0.73803   
## ---  
## 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

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

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

## Start: AIC=268.09  
## violator ~ crime + multiple.offenses + max.sentence + time.served +   
## state + age + race + male  
##   
## 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 ~ multiple.offenses + max.sentence + time.served + state +   
## age + race + male  
##   
## 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 ~ multiple.offenses + max.sentence + time.served + state +   
## race + male  
##   
## 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 ~ multiple.offenses + max.sentence + time.served + state +   
## race  
##   
## 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 ~ multiple.offenses + max.sentence + state + race  
##   
## 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 ~ multiple.offenses + state + race  
##   
## 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(backmod)

##   
## Call:  
## glm(formula = violator ~ multiple.offenses + state + 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 \*\*\*  
## multiple.offensesMultiple Offenses 1.73482 0.39421 4.401 1.08e-05 \*\*\*  
## 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 \*\*\*  
## raceOtherwise 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

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

##   
## 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 Offenses 1.73482 0.39421 4.401 1.08e-05 \*\*\*  
## raceOtherwise 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

**According to the AIC on the backward and forward stepwise model’s they are each considered to be well performing and reliable. There isn’t enough difference in the results of either model to distinguish one as better than the other. Both models found that the state of Virginia and “other” are the least likely to have parole offenders. Both models also landed on the same AIC of 271.29.**

# Task 5

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

##   
## 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 Offenses 1.73482 0.39421 4.401 1.08e-05 \*\*\*  
## raceOtherwise 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

**The state of Virginia is significant in that it is very unlikely that a released inmate commit parole violation. The race variable is also significant such that it predicts other races than “white” are more likely to violate parole. The AIC value of 268.09 states we can rely on this particular model for reliability.**

# Task 6

newdata = data.frame(state = "Louisiana", multiple.offenses = "Multiple Offenses", race = "White")  
predict(logRM, newdata, type = "response")

## 1   
## 0.3379961

newdata = data.frame(state = "Kentucky", multiple.offenses = "Otherwise", race = "Otherwise")  
predict(logRM, newdata, type = "response")

## 1   
## 0.2069629

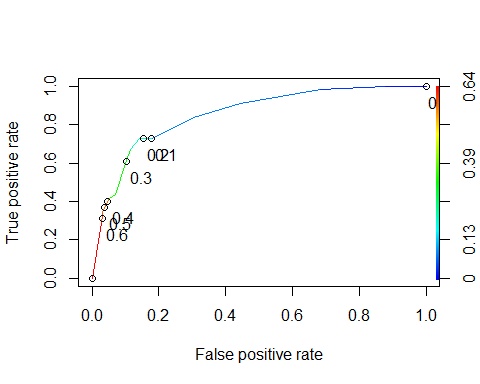
**The predicted probability of parole violation for the white Louisiana resident with multiple offenses is 53%, while the “other” race Kentucky resident with “otherwise” offenses is 13%.**

# Task 7

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

## 1 2 3 4 5 6   
## 0.07509978 0.19512504 0.19512504 0.07509978 0.07509978 0.19512504

ROCRpred = prediction(predictions, train$violator,label.ordering = c("Without Violation", "Violated Parole"))  
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))



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 0.7272727  
## specificity 0.8588517  
## cutoff 0.2069629

# Task 8

sensitivity 0.7272727

specificity 0.8588517

cutoff 0.2069629

**The implication of classifying a parolee as a violator of parole is a bit too inflated for comfort at 0.20.**

# Task 9

t1 = table(train$violator,predictions > 0.2069629)  
t1

##   
## FALSE TRUE  
## Without Violation 359 59  
## Violated Parole 15 40

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

##   
## FALSE TRUE  
## Without Violation 405 13  
## Violated Parole 36 19

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

## [1] 0.6281481

**It appears as though a threshold of 0.5 is going to be the value that gets us the maximum accuracy rating of 0.63. The formula does not accept a value over 0.7.**

# Task 10

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

## [1] 2.09901