# BAN502

## Kyle Capponcelli

### MOD 3 Assignment 2: Logistic Regression (Classification)

library("tidyverse", quietly = TRUE)

## -- Attaching packages ------------------------------------------ tidyverse 1.3.0 --

## v ggplot2 3.2.1 v purrr 0.3.3  
## v tibble 2.1.3 v dplyr 0.8.3  
## v tidyr 1.0.0 v stringr 1.4.0  
## v readr 1.3.1 v forcats 0.4.0

## -- 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

library("leaps")  
library("e1071")

CSV Load-in

library(readr)  
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()  
## )

View(parole)

Data Conversion

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", "other" = "2"))   
   
parole = parole %>% mutate(state = as.factor(as.character(state))) %>%  
 mutate(state = fct\_recode(state, "Kentucky" = "2", "Louisiana" = "3", "Virginia" = "4", "OtherState" = "1"))  
   
parole = parole %>% mutate(crime = as.factor(as.character(crime))) %>%  
 mutate(crime = fct\_recode(crime, "larceny" = "2", "drug-related" = "3", "driving-related" = "4", "OtherCrime" = "1"))   
  
parole = parole %>% mutate(multiple.offenses = as.factor(as.character(multiple.offenses))) %>%  
 mutate(multiple.offenses = fct\_recode(multiple.offenses, "MultipleOffense" = "1", "Other" = "0"))  
  
parole = parole %>% mutate(violator = as.factor(as.character(violator))) %>%  
 mutate(violator = fct\_recode(violator, "violated" = "1", "completed" = "0"))  
  
str(parole)

## Classes 'spec\_tbl\_df', 'tbl\_df', 'tbl' and 'data.frame': 675 obs. of 9 variables:  
## $ male : Factor w/ 2 levels "female","male": 2 1 2 2 2 2 2 1 1 2 ...  
## $ race : Factor w/ 2 levels "white","other": 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 "OtherState","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 "Other","MultipleOffense": 1 1 1 1 1 1 1 1 1 1 ...  
## $ crime : Factor w/ 4 levels "OtherCrime","larceny",..: 4 3 3 1 1 4 3 1 3 2 ...  
## $ violator : Factor w/ 2 levels "completed","violated": 1 1 1 1 1 1 1 1 1 1 ...

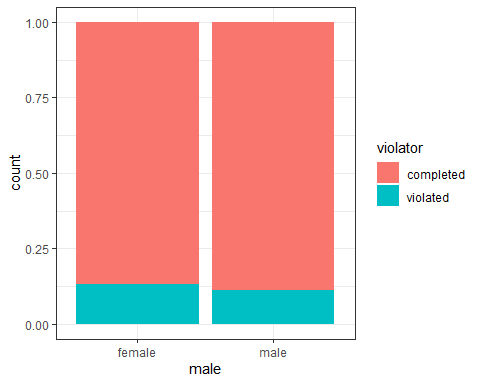
#### Task 1

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

#### Task 2

Male -> Violator

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



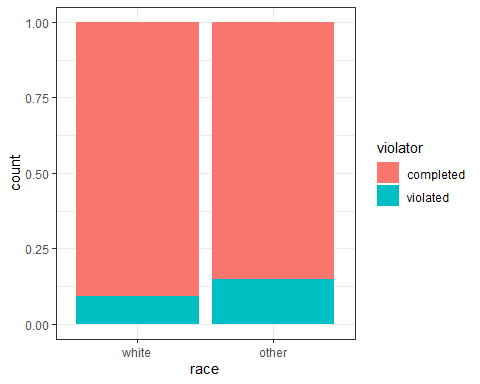
t1 = table(train$violator, train$male)  
prop.table(t1, margin = 2)

##   
## female male  
## completed 0.8673469 0.8880000  
## violated 0.1326531 0.1120000

I would not consider the violator sex to be a good indicator of parole violation as male and female violator percentages are within 2.5% of each other. Male = 11.2%, Female = 13.3%.

Race -> Violator

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



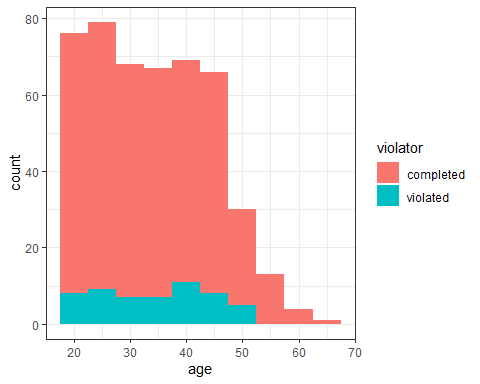
t2 = table(train$violator, train$race)  
prop.table(t2, margin = 2)

##   
## white other  
## completed 0.90774908 0.85148515  
## violated 0.09225092 0.14851485

Given that there is a 14.85% chance that a non-white ethnicity broke parole vs. a 9.23 % chance a white ethnicity broke parole I would say that this is a good indicator of “violator”

Age -> Violator

ggplot(train, aes(x=age, fill= violator)) + geom\_histogram(binwidth = 5) + theme\_bw()

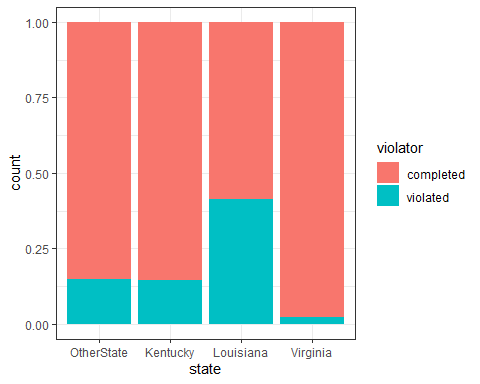


# t3 = table(train$violator, train$age)  
# prop.table(t3, margin = 2) ## this displayed too much info so for the sake of tidying up the markdown doc I commented it out.

Creating a table with each age in its own group was not overly helpful, however, looking at the histogram, I would say that age is an okay at best indicator of future violation as it appears that no-one over the age of 53 appears to violate parole and that individuals around the age of 30 seem to violate parole the most. I will test this by adding and removing it from my final model and comparing the AIC values.

State -> Violator

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



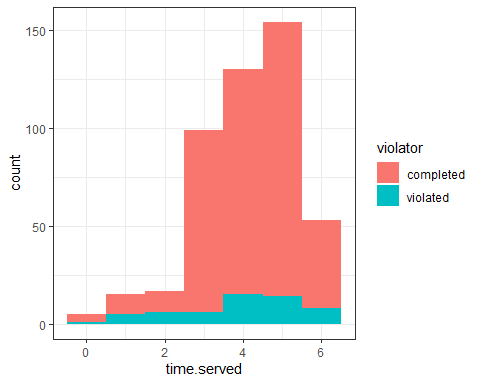
t4 = table(train$violator, train$state)  
prop.table(t4, margin = 2)

##   
## OtherState Kentucky Louisiana Virginia  
## completed 0.85263158 0.85542169 0.58620690 0.97890295  
## violated 0.14736842 0.14457831 0.41379310 0.02109705

State appears to be a good indicator of parole violation as there is a large range in data. At the low end is Virginia with a 2.11% chance of parole violation and at the high end is Louisiana with a 41.38% chance of parole violation.

time.served -> Violator

ggplot(train, aes(x=time.served, fill= violator)) + geom\_histogram(binwidth = 1) + theme\_bw()

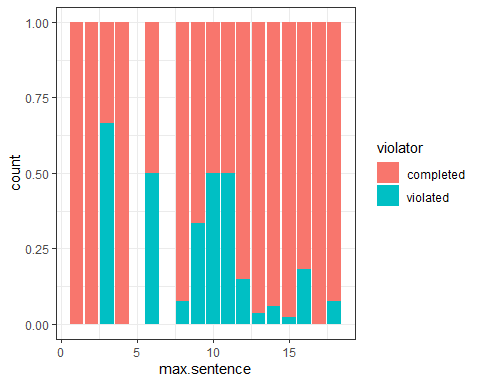


# t5 = table(train$violator, train$time.served)  
# prop.table(t5, margin = 2) ## this displayed too much info so for the sake of tidying up the markdown doc I commented it out.

Time served appears to be a poor indicator of parole violation as the percentage of violations vary widely accross all amounts of time served.

max.sentence -> Violator

ggplot(train, aes(x=max.sentence, fill= violator)) + geom\_bar(position = "fill") + theme\_bw()



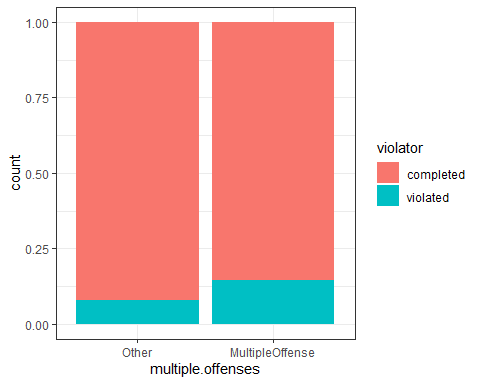
t6 = table(train$violator, train$max.sentence)  
prop.table(t6, margin = 2)

##   
## 1 2 3 4 6 8  
## completed 1.00000000 1.00000000 0.33333333 1.00000000 0.50000000 0.92307692  
## violated 0.00000000 0.00000000 0.66666667 0.00000000 0.50000000 0.07692308  
##   
## 9 10 11 12 13 14  
## completed 0.66666667 0.50000000 0.50000000 0.85204082 0.96363636 0.94000000  
## violated 0.33333333 0.50000000 0.50000000 0.14795918 0.03636364 0.06000000  
##   
## 15 16 17 18  
## completed 0.97619048 0.81818182 1.00000000 0.92307692  
## violated 0.02380952 0.18181818 0.00000000 0.07692308

Max sentence appears to be an okay indicator of violators as longer sentences appear to have a lower percentage of violators.

multiple.offenses -> Violator

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



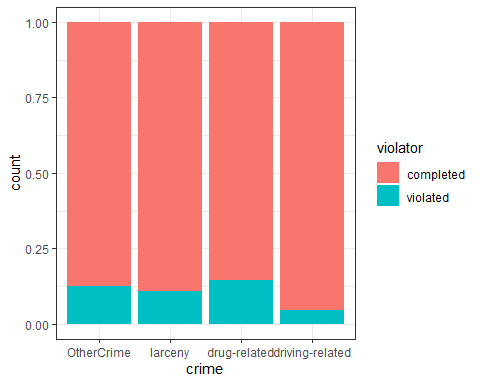
t7 = table(train$violator, train$multiple.offenses)  
prop.table(t7, margin = 2)

##   
## Other MultipleOffense  
## completed 0.91981132 0.85440613  
## violated 0.08018868 0.14559387

Multiple offense is a good indicator of violators as those with multiple offenses have a 14.56% chance to violate parole and other has a 8.02 % chance to violate parole.

crime -> Violator

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



t8 = table(train$violator, train$crime)  
prop.table(t8, margin = 2)

##   
## OtherCrime larceny drug-related driving-related  
## completed 0.87445887 0.89189189 0.85436893 0.95384615  
## violated 0.12554113 0.10810811 0.14563107 0.04615385

As with age, the crime does not seem to matter much when comparing to potential violators. The real difference appears to mostly be between driving related crimes and all the others as the others have a violator rate within 4% (roughly) of each other. I will test the model with and without crime to see if there is a significant difference in AIC values.

#### Task 3

Test Model -> Violator Significant Factors = race, age, state, max.sentence, multiple.offenses, crime Appears to be most significant = 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

AIC with state as the predictor of violator = 283.18 This looks like a low AIC however I cannot say without having other models to compare it to. State was chosen as it appeared to have the biggest difference when comparing the percent of violators against those that don’t violate parole.

#### Task 4

Backward Stepwise Regression

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

##   
## Call:  
## glm(formula = violator ~ male + race + age + state + time.served +   
## max.sentence + multiple.offenses + crime, 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) -2.885777 1.197993 -2.409 0.01600 \*   
## malemale -0.137577 0.411340 -0.334 0.73803   
## raceother 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.offensesMultipleOffense 1.711032 0.396463 4.316 1.59e-05 \*\*\*  
## crimelarceny 0.392910 0.514075 0.764 0.44469   
## crimedrug-related -0.210563 0.413351 -0.509 0.61047   
## crimedriving-related -0.727043 0.690775 -1.053 0.29257   
## ---  
## 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 ~ 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

Using Backward Stepwise Regression on the training set I ended with a final AIC value of 258.98 which uses the predictors of race, multiple.offenses and state to predict the violator variable and is the best AIC value I have received so far. I believe that this model is fairly intuitive although I would raise questions as to why most of the data went unused when calculating predictor values. If I were unfamiliar with this kind of data and AIC values I would question how we could get a good prediction with so few of the variables on the table being used as predictors.

#### Task 5

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   
## -1.3609 -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.offensesMultipleOffense 1.73482 0.39421 4.401 1.08e-05 \*\*\*  
## raceother 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 AIC value is 258.98 (as expected). This AIC value indicates that this is a good model as it is the same AIC value as our optimal model. Values of significance are; being from the state of Virgina, if there are multiple.offenses and the race of the individual. If you are in the state of virginia you have a low chance of parole violation. If you have less than 1 offense you have a low chance of parole violation. If you are white, you have a lower chance of parole violation. stateVirginia, multiple.offensesOther, racewhite all have p values less than .05 making them significant.

#### Task 6

Predictions

Parolee1 = data.frame(state = "Louisiana", multiple.offenses = "MultipleOffense", race = "white")  
predict(backmod, Parolee1, type = "response")

## 1   
## 0.3379961

Parolee2 = data.frame(state = "Kentucky", multiple.offenses = "Other", race = "other")  
predict(backmod, Parolee2, type = "response")

## 1   
## 0.2069629

Parolee3 = data.frame(state = "Louisiana", multiple.offenses = "MultipleOffense", race = "other")  
predict(backmod, Parolee3, type = "response")

## 1   
## 0.6038624

Parolee4 = data.frame(state = "Virginia", multiple.offenses = "Other", race = "white")  
predict(backmod, Parolee4, type = "response")

## 1   
## 0.002196228

According to our predictions, Parolee1 has a 33.8% chance of violating parole and Parolee2 has a 20.7% chance of violating parole. To further test the validity of the model I added Parolee3 with all of the factors that would lead to the highest parole violation assumption and Parolee4 to test the factors with the lowest suspected violator rate. Parolee3 has a 60.4% chance of violating parole. Parolee4 has a 0.22% chance of violating parole. This makes sense within our model.

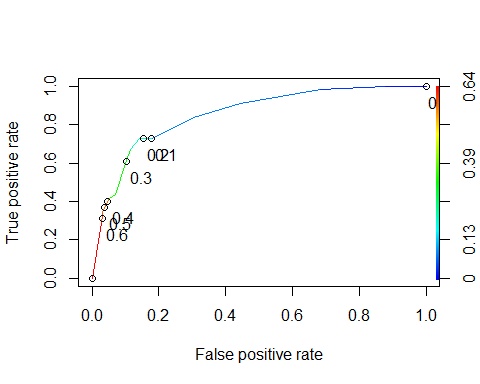
#### Task 7

ROC Curve

predictions = predict(backmod, 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)  
  
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))



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

## [1] 0.8524576

#### Task 8

Accuracy, Sensitivity, 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(ROCRperf, ROCRpred))

## [,1]  
## sensitivity 0.7272727  
## specificity 0.8588517  
## cutoff 0.2069629

This model is fairly accurate although it would be better if we could encompass more of the data and we have a fairly high cutoff value of 20.7%. These means that more than a fifth of the parolees we predict will or will not be violators will be misclassified. The sensitivity and specificity do not appear to be well balanced.

Misclassifying parolees could lead to more parolees becoming violators and ultimitaley be costly to the legal system as more resources would be needed to deal with misclassified violators. Resources could also be spent unnecessarily on those who would not normally become violators.

#### Task 9

Training-set Accuracy

ConM1 = table(train$violator, predictions > 0.2069629)  
ConM1

##   
## FALSE TRUE  
## completed 359 59  
## violated 15 40

(ConM1[1,1]+ConM1[2,2])/nrow(train)

## [1] 0.8435518

ConM2 = table(train$violator, predictions > 0.60)  
ConM2

##   
## FALSE TRUE  
## completed 406 12  
## violated 39 16

(ConM2[1,1]+ConM2[2,2])/nrow(train)

## [1] 0.8921776

The original table has an accuracy of 84.36%. After some trial and error I found the threshold value of 0.60 to maximize the accuracy of the table without going outside of bounds. I attempted to add more accuracy and got to 0.60386 and still had the same accuracy as 0.6 which is 89.23%. This is a decent improvement (almost 5%) in accuracy.

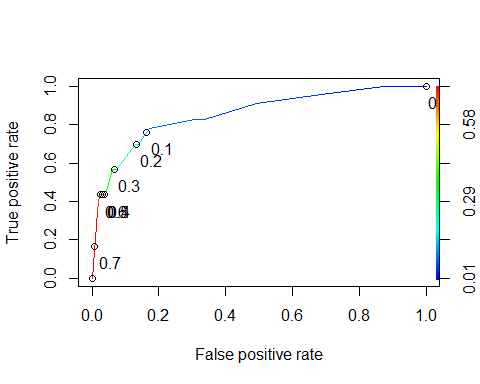
#### Task 10

Testing-set Accuracy

mod3 = glm(violator ~ state + multiple.offenses + race, test, family = "binomial")  
summary(mod3)

##   
## Call:  
## glm(formula = violator ~ state + multiple.offenses + race, family = "binomial",   
## data = test)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.4993 -0.3986 -0.2328 -0.2270 2.7099   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -2.49273 0.54784 -4.550 5.36e-06 \*\*\*  
## stateKentucky -0.49375 0.88667 -0.557 0.57762   
## stateLouisiana 1.89491 0.66311 2.858 0.00427 \*\*   
## stateVirginia -2.43099 0.89277 -2.723 0.00647 \*\*   
## multiple.offensesMultipleOffense 1.27765 0.58876 2.170 0.03000 \*   
## raceother 0.05111 0.60654 0.084 0.93284   
## ---  
## 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: 99.121 on 196 degrees of freedom  
## AIC: 111.12  
##   
## Number of Fisher Scoring iterations: 6

predictions2 = predict(mod3, type = "response")  
  
ROCRpred2 = prediction(predictions2, test$violator)  
  
ROCRperf2 = performance(ROCRpred2, "tpr", "fpr")  
plot(ROCRperf2, colorize = TRUE, print.cutoffs.at=seq(0,1, by = 0.1),text.adj=c(-0.2, 1.7))



as.numeric(performance(ROCRpred2, "auc")@y.values)

## [1] 0.856206

ConM3 = table(test$violator, predictions2 > 0.6)  
  
ConM3

##   
## FALSE TRUE  
## completed 175 4  
## violated 13 10

After running the threshold that was optimized for the training set I went and built the model for the testing set and ran with the threshold of 0.6. The accuracy came down a little on the testing set to 85.62% which is still decent. I would say that this model is usable and can be used to predict new data.