# Fields, Alex

## BAN 502

### Module 3 - Assignment 2

#### Task 2 - Q&A

My thought proces is that I would want to use logistic regression for predicting categorical values for a parolee violating/not violoating parole. I would want to use a box plot with categorical variables and box plots for numeric variables.

#### Task 3 - Q&A

The information that would “predict” violator in my opinion would be Gender (male), race, age and state. These seem to logically explain this variable. I decided to start my model with Violator ~ State. This gave an AIC of 390. All states we statistically significant except for Kentucky. The model seems “ok”. I feel it can be better.

#### Task 4 - Q&A

When running forward and backwards models I noticed that for forward, we keep the variables: state, multiple.offenses, age, race max.sentence. All variables are significant except age and max.sentence. The AIC on this is very low, close to 252 which is what we want.

For backwards selection, it seems very similiar in terms of AIC being around 252. However, variables age state and max.sentence are not significant. Since both are bringing roughly the same results I will examine the logic of the results.

State and Age seem less significant since these factors dont really play a role as to whether you violate parole or not. If a certain state has a stricter bond/parole law than another, that may play a role. Its hard to tell if someones age really affects this since everyones maturity level is different at certain ages. Max Sentence makes sense to affect the model since the higher the sentence either, you will want to break it to never do time or you’ll want to abide by the rules if the sentence is only a fwe months so you dont get an extended sentence.

#### Task 5 - Q&A

When running a logistic regression model on violator being the predictor and State, multiple.offenses and race I can see that this model seems good. Its not perfect but the AIC is 365, which is pretty low and closer to what we want. States of Louisiana, Kentucky are not significant in the model but this could be for many reasons. I belive this ok since states can have different parole laws (3 strike rule) and this can skew the data.

#### Task 8 - Q&A

The accuracy of the model is 25% while the sensitivity 83.73% and specificity 18.18% respectivly. The issue of incorrectly classifying a Parolee could mean more jail time if they are falsely accused of violating parole. That could mean years added on a persons sentence.

#### Task 9 - Q&A

When looking at the optimal probability threshold we can see that around > 0.5 which yields about 90.4% accuracy.

#### Task 10 - Q&A

When recreating the same procedure from Task 9, I can see that using a threshold of > 0.5 yields a 13% increase in model accuracy. Sensitivity was 0.005586592, specificity was 0.956521739 and the cutoff was closer to 1. The accuracy of the model was 38% respectivly. The test seemed to be a better model than the training set but it was still poor being so inaccurate.

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()  
## )

parole = as\_tibble(parole)  
  
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,  
"other" = "1",  
"Kentucky" = "2",  
"Louisiana" = "3",  
"Virginia" = "4"))  
  
parole = parole %>% mutate(crime = as\_factor(as.character(crime))) %>%  
mutate(crime = fct\_recode(crime,  
"other" = "1",  
"larceny" = "2",  
"drug-related crime" = "3",  
"driving-related crime" = "4"))  
  
parole = parole %>% mutate(violator = as\_factor(as.character(violator))) %>%  
mutate(violator = fct\_recode(violator,  
"Violated Parole" = "0",  
"No Parole Violation" = "1"))  
  
parole = parole %>% drop\_na() #drops N/A's  
str(parole)

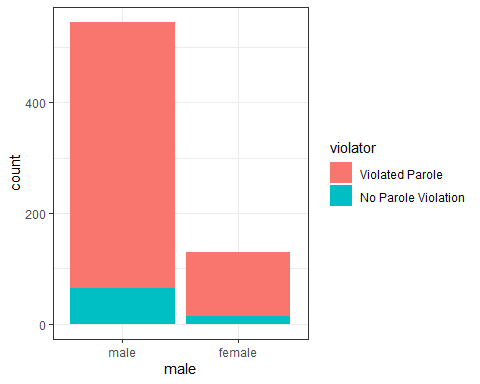
## tibble [675 x 9] (S3: tbl\_df/tbl/data.frame)  
## $ male : Factor w/ 2 levels "male","female": 1 2 1 1 1 1 1 2 2 1 ...  
## $ race : Factor w/ 2 levels "white","other": 1 1 2 1 2 2 1 1 1 2 ...  
## $ age : num [1:675] 33.2 39.7 29.5 22.4 21.6 46.7 31 24.6 32.6 29.1 ...  
## $ state : Factor w/ 4 levels "other","Kentucky",..: 1 1 1 1 1 1 1 1 1 1 ...  
## $ time.served : num [1:675] 5.5 5.4 5.6 5.7 5.4 6 6 4.8 4.5 4.7 ...  
## $ max.sentence : num [1:675] 18 12 12 18 12 18 18 12 13 12 ...  
## $ multiple.offenses: num [1:675] 0 0 0 0 0 0 0 0 0 0 ...  
## $ crime : Factor w/ 4 levels "driving-related crime",..: 1 2 2 3 3 1 2 3 2 4 ...  
## $ violator : Factor w/ 2 levels "Violated Parole",..: 1 1 1 1 1 1 1 1 1 1 ...

training data

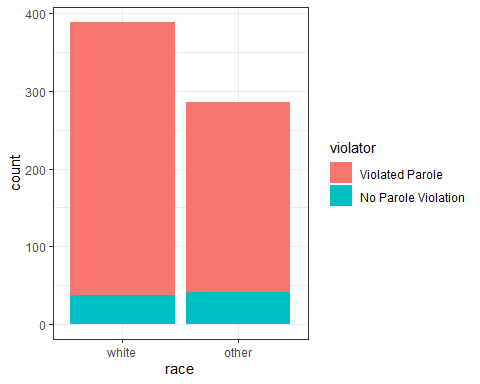
set.seed(12345)  
train.rows = createDataPartition(y = parole$violator, p = 0.7, list =FALSE)  
train = slice(parole, train.rows)  
test = slice(parole, -train.rows)

Visualizing what fits best with Violator with bar plot

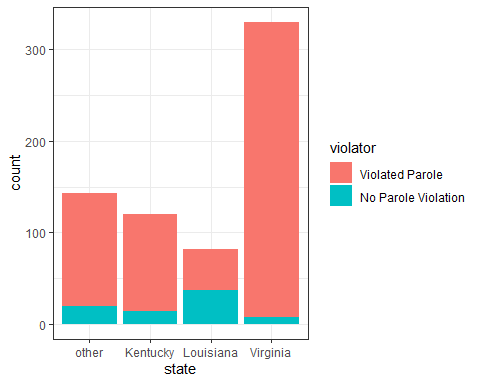
library(ggplot2)  
  
ggplot(parole, aes(x=male, fill = violator)) + geom\_bar() + theme\_bw()



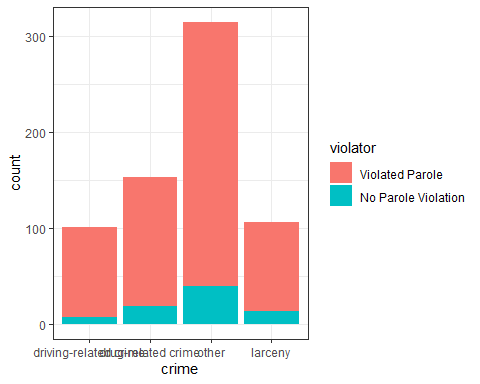
ggplot(parole, aes(x=race, fill = violator)) + geom\_bar() + theme\_bw()



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

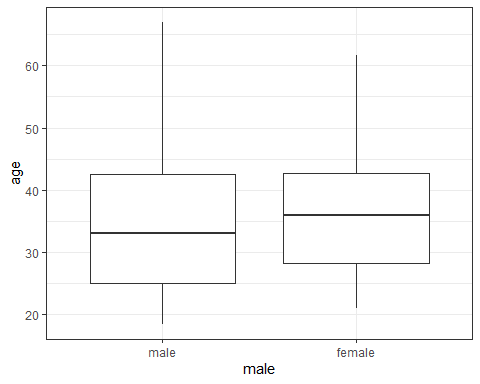


ggplot(parole, aes(x=crime, fill = violator)) + geom\_bar() + theme\_bw()

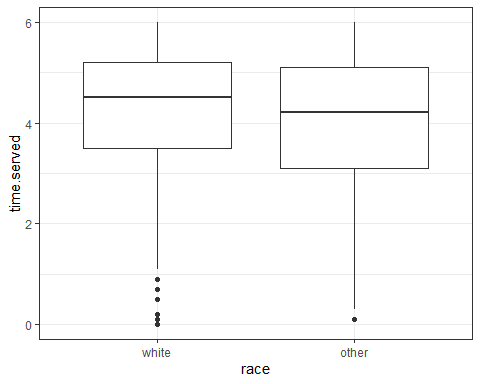


Visualizing what fits best with Violator with box plot

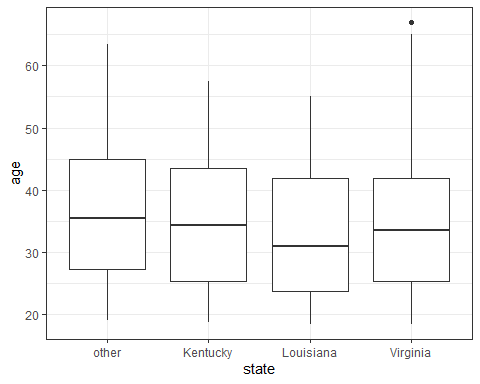
library(ggplot2)  
  
ggplot(parole, aes(x=male, y=age)) + geom\_boxplot() + theme\_bw()



ggplot(parole, aes(x=race, y=time.served)) + geom\_boxplot() + theme\_bw()



ggplot(parole, aes(x=state, y=age)) + geom\_boxplot() + theme\_bw()



ggplot(parole, aes(x=age, y=violator)) + geom\_boxplot() + theme\_bw()



tabular data for prediction

t1 = table(parole$violator, parole$multiple.offenses)  
prop.table(t1, margin = 2)

##   
## 0 1  
## Violated Parole 0.9201278 0.8535912  
## No Parole Violation 0.0798722 0.1464088

t2 = table(parole$violator, parole$max.sentence)  
prop.table(t2, margin = 2)

##   
## 1 2 3 4 5  
## Violated Parole 0.75000000 1.00000000 0.33333333 1.00000000 0.00000000  
## No Parole Violation 0.25000000 0.00000000 0.66666667 0.00000000 1.00000000  
##   
## 6 8 9 10 11  
## Violated Parole 0.56250000 0.85000000 0.66666667 0.66666667 0.60000000  
## No Parole Violation 0.43750000 0.15000000 0.33333333 0.33333333 0.40000000  
##   
## 12 13 14 15 16  
## Violated Parole 0.86496350 0.96103896 0.95454545 0.98333333 0.88888889  
## No Parole Violation 0.13503650 0.03896104 0.04545455 0.01666667 0.11111111  
##   
## 17 18  
## Violated Parole 0.93333333 0.89743590  
## No Parole Violation 0.06666667 0.10256410

Starting with Violator as the Predictor and State as the Response variable

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

##   
## Call:  
## glm(formula = violator ~ state, family = "binomial", data = parole)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.0955 -0.4981 -0.2071 -0.2071 2.7760   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -1.8165 0.2411 -7.534 4.92e-14 \*\*\*  
## stateKentucky -0.2079 0.3728 -0.558 0.577   
## stateLouisiana 1.6207 0.3277 4.946 7.58e-07 \*\*\*  
## stateVirginia -2.0153 0.4517 -4.461 8.15e-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: 382.89 on 671 degrees of freedom  
## AIC: 390.89  
##   
## Number of Fisher Scoring iterations: 6

Forward and Backwards selection

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

##   
## 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   
## 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.offenses 1.711032 0.396463 4.316 1.59e-05 \*\*\*  
## crimedrug-related crime 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   
## ---  
## 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   
## -1.177 -1.177 -1.177 -1.177 1.177   
##   
## No Coefficients  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 655.72 on 473 degrees of freedom  
## Residual deviance: 655.72 on 473 degrees of freedom  
## AIC: 655.72  
##   
## Number of Fisher Scoring iterations: 0

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

## Start: AIC=655.72  
## violator ~ -1  
##   
## Df Deviance AIC  
## + state 3 327.44 333.44  
## + max.sentence 1 331.53 333.53  
## + male 2 339.72 343.72  
## + time.served 1 344.20 346.20  
## + age 1 359.88 361.88  
## + crime 3 400.86 406.86  
## + multiple.offenses 1 510.52 512.52  
## + race 1 545.42 547.42  
## <none> 655.72 655.72  
##   
## Step: AIC=283.18  
## violator ~ state - 1  
##   
## 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 - 1  
##   
## 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 - 1  
##   
## 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 - 1,   
## 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|)   
## stateother -2.5109 0.3635 -6.907 4.96e-12 \*\*\*  
## stateKentucky -2.4372 0.3767 -6.470 9.79e-11 \*\*\*  
## stateLouisiana -2.4071 0.5331 -4.515 6.32e-06 \*\*\*  
## stateVirginia -6.1188 0.6825 -8.965 < 2e-16 \*\*\*  
## multiple.offenses 1.7348 0.3942 4.401 1.08e-05 \*\*\*  
## raceother 1.0938 0.3897 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: 655.72 on 473 degrees of freedom  
## Residual deviance: 246.98 on 467 degrees of freedom  
## AIC: 258.98  
##   
## Number of Fisher Scoring iterations: 6

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

summary(backmod)

##   
## 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 \*\*\*  
## raceother 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.offenses 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

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

##   
## 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.offenses 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

Prediction

#parolee 1  
newdata1 = data.frame(state = "Louisiana", multiple.offenses = 1, race = "white")  
predict(mod1, newdata1, type="response")

## 1   
## 0.3379961

#parolee 2  
newdata2 = data.frame(state = "Kentucky", multiple.offenses = 0, race = "other")#2 stands for "other"  
predict(mod1, newdata2, type="response")

## 1   
## 0.2069629

ctrl = trainControl(method = "cv", number = 10)  
  
set.seed(12345)  
modKFold = train(violator ~., train, method = "glm", trControl = ctrl)  
summary(modKFold)

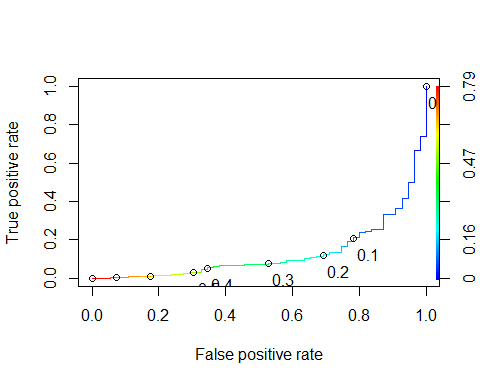
##   
## Call:  
## NULL  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.6055 -0.3932 -0.2643 -0.1384 2.9470   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -3.750397 1.318165 -2.845 0.00444 \*\*   
## malefemale 0.137577 0.411340 0.334 0.73803   
## 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.offenses 1.711032 0.396463 4.316 1.59e-05 \*\*\*  
## `crimedrug-related crime` 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   
## ---  
## 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

Develop ROC curve

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

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

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] 0.1289691

Create Specificity & Sensitivity

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 0.007177033  
## specificity 0.927272727  
## cutoff 0.696514922

Test thresholds to evaluate accuracy

#confusion matrix  
t1 = table(train$violator,predictions > .01166761)  
t1

##   
## FALSE TRUE  
## Violated Parole 92 326  
## No Parole Violation 0 55

Calculate accuracy

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

## [1] 0.3107822

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

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

##   
## FALSE TRUE  
## Violated Parole 406 12  
## No Parole 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  
## Violated Parole 414 4  
## No Parole Violation 46 9

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

## [1] 0.8942918

A naive prediction (everyone doesn’t violate)

t1 = table(train$violator,predictions > 1)  
t1

##   
## FALSE  
## Violated Parole 418  
## No Parole Violation 55

(t1[1])/nrow(train)

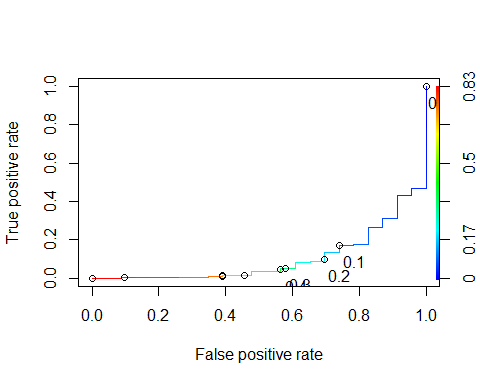
## [1] 0.8837209

Recreating for test set

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

##   
## 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 \*   
## raceother 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.offenses 1.448327 0.657212 2.204 0.02754 \*   
## crimedrug-related crime -1.726431 1.060781 -1.628 0.10363   
## crimeother -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

predictions = predict(allmod, type = "response")  
  
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))



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

## [1] 0.1012873

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 0.005586592  
## specificity 0.913043478  
## cutoff 0.802512427

threshold

#confusion matrix  
t1 = table(test$violator,predictions > 0.5)  
t1

##   
## FALSE TRUE  
## Violated Parole 176 3  
## No Parole Violation 13 10

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

## [1] 0.3932347