## Logistic Regression (Classification)

### Anthony Sumter

#install.packages("tidyverse","MASS", "caret", "ROCR")  
library(tidyverse)

## -- Attaching packages ------------------------------------------------------ tidyverse 1.2.1 --

## 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)

## Warning: package 'caret' was built under R version 3.6.2

## 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 3.6.2

## Loading required package: gplots

## Warning: package 'gplots' was built under R version 3.6.2

##   
## Attaching package: 'gplots'

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

parole = read.csv("parole.csv")  
parole = parole %>% mutate(male = as\_factor(as.character(male)), race = as\_factor (as.character (race)), state = as\_factor(as.character (state)), crime = as\_factor(as.character (crime)), multiple.offenses = as\_factor(as.character(multiple.offenses)), violator = as\_factor(as.character(violator))) %>%  
mutate(male = fct\_recode(male,"female" = "0","male" = "1")) %>% mutate(race = fct\_recode(race,"White" = "1","Otherwise" = "2"))%>% mutate(state = fct\_recode(state,"OtherState" = "1","Kentucky" = "2", "Louisiana"= "3", "Virgina" = "4"))%>% mutate(crime = fct\_recode(crime,"OtherCrime" = "1", "Larceny" = "2", "DrugRelated" = "3", "DrivingRelated" = "4")) %>% mutate(multiple.offenses = fct\_recode(multiple.offenses,"Otherwise" = "0", "MultipleOffenses" = "1")) %>% mutate(violator = fct\_recode(violator,"Violated" = "1", "NonViolated" = "0"))

#### Task 1

**Split the data into training and testing sets. Your training set should have 70% of the data. Use a random number (set.seed) of 12345.**

parole = parole %>% drop\_na()  
str(parole)

## '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","Otherwise": 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 : int 18 12 12 18 12 18 18 12 13 12 ...  
## $ multiple.offenses: Factor w/ 2 levels "Otherwise","MultipleOffenses": 1 1 1 1 1 1 1 1 1 1 ...  
## $ crime : Factor w/ 4 levels "DrivingRelated",..: 1 2 2 3 3 1 2 3 2 4 ...  
## $ violator : Factor w/ 2 levels "NonViolated",..: 1 1 1 1 1 1 1 1 1 1 ...

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

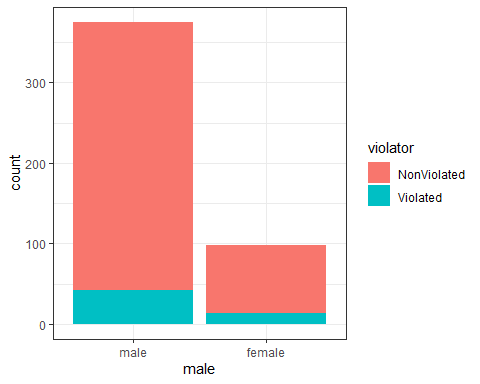
#### Task 2

**Our objective is to predict whether or not a parolee will violate his/her parole. In this task, use appropriate data visualizations and/or tables to identify which variables in the training set appear to be most predictive of the response variable “violator”. Provide a brief explanation of your thought process.**

I would think that the variables that would be the most predictive of the response variable violator would be the type of crime because a more severe crime committed would likely lead to stricter parole guidelines which would increase the chance of someone not being able to adhere to those standards. Also, multiple offenses because it is a greater oportunity to violate parole because of the likelihood that an individual would continue to stay in trouble with the law.

**Male**

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

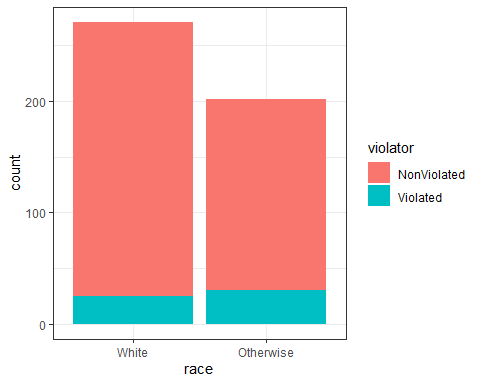


t1 = table(train$violator, train$male)  
prop.table(t1, margin = 2 )

##   
## male female  
## NonViolated 0.8880000 0.8673469  
## Violated 0.1120000 0.1326531

**Race**

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

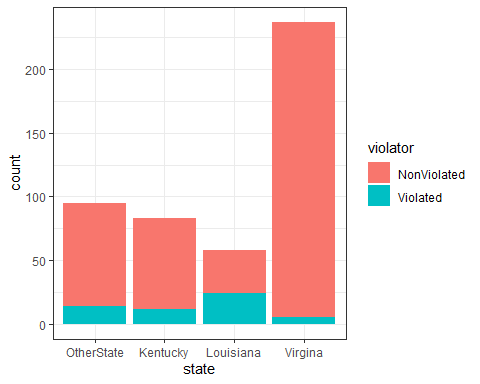


t2 = table(train$violator, train$race)  
prop.table(t2, margin = 2 )

##   
## White Otherwise  
## NonViolated 0.90774908 0.85148515  
## Violated 0.09225092 0.14851485

**State**

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

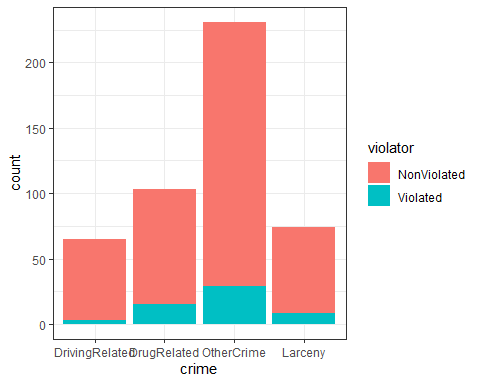


t3 = table(train$violator, train$state)  
prop.table(t3, margin = 2 )

##   
## OtherState Kentucky Louisiana Virgina  
## NonViolated 0.85263158 0.85542169 0.58620690 0.97890295  
## Violated 0.14736842 0.14457831 0.41379310 0.02109705

**Crime**

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

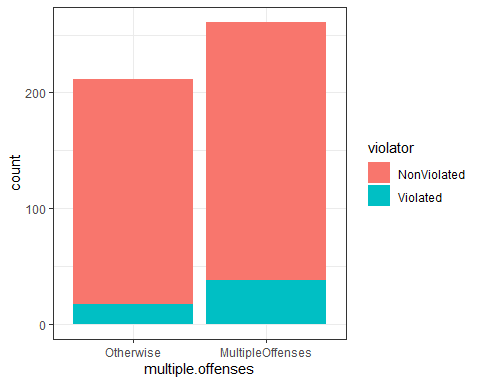


t4 = table(train$violator, train$crime)  
prop.table(t4, margin = 2 )

##   
## DrivingRelated DrugRelated OtherCrime Larceny  
## NonViolated 0.95384615 0.85436893 0.87445887 0.89189189  
## Violated 0.04615385 0.14563107 0.12554113 0.10810811

**Multiple Offenses**

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



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

##   
## Otherwise MultipleOffenses  
## NonViolated 0.91981132 0.85440613  
## Violated 0.08018868 0.14559387

#### Task 3

**Identify the variable from Task 2 that appears to you to be most predictive of “violator”. Create a logistic regression model using this variable to predict violator. Comment on the quality of the model.**

The quality of this model would be pretty decent due to the fact the AIC value is pretty low and the variable chose is significant because of the p value being less than .05.

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

##   
## Call:  
## glm(formula = violator ~ multiple.offenses, family = "binomial",   
## data = train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -0.5610 -0.5610 -0.4089 -0.4089 2.2465   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -2.4398 0.2529 -9.648 <2e-16 \*\*\*  
## multiple.offensesMultipleOffenses 0.6702 0.3078 2.177 0.0295 \*   
## ---  
## 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: 335.02 on 471 degrees of freedom  
## AIC: 339.02  
##   
## Number of Fisher Scoring iterations: 5

#### Task 4

**Using forward stepwise, backward stepwise, or by manually building a model, create the best model you can to predict “violator”. Use only the training data set and use AIC to evaluate the “goodness” of the models. Comment on the quality of your final model. In particular, note which variables are significant and comment on how intuitive the model may (or may not) be.**

Using the backward stepwise method of the training data set verified that the quality of this model would be pretty decent due to the fact the AIC value is pretty low. four out of the final 6 varibles that were used in the final version of the backward stepwise method were significant. Those four were race otherwise, the state of Kentucky, the state of Virgina, and multiple offenses.

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

##   
## Call:  
## glm(formula = violator ~ male + race + state + crime + multiple.offenses,   
## family = "binomial", data = train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.5672 -0.4146 -0.2676 -0.1394 2.9787   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -3.07008 0.67349 -4.558 5.15e-06 \*\*\*  
## malefemale 0.06636 0.40761 0.163 0.87068   
## raceOtherwise 1.11309 0.39609 2.810 0.00495 \*\*   
## stateKentucky 0.01788 0.48156 0.037 0.97039   
## stateLouisiana 0.05163 0.50264 0.103 0.91819   
## stateVirgina -3.70319 0.65031 -5.695 1.24e-08 \*\*\*  
## crimeDrugRelated 0.49122 0.72755 0.675 0.49957   
## crimeOtherCrime 0.65912 0.67618 0.975 0.32967   
## crimeLarceny 1.03083 0.78687 1.310 0.19018   
## multiple.offensesMultipleOffenses 1.68969 0.39298 4.300 1.71e-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: 244.76 on 463 degrees of freedom  
## AIC: 264.76  
##   
## 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=264.76  
## violator ~ male + race + state + crime + multiple.offenses  
##   
## Df Deviance AIC  
## - crime 3 246.78 260.78  
## - male 1 244.78 262.78  
## <none> 244.76 264.76  
## - race 1 252.75 270.75  
## - multiple.offenses 1 264.54 282.54  
## - state 3 326.53 340.53  
##   
## Step: AIC=260.78  
## violator ~ male + race + state + multiple.offenses  
##   
## Df Deviance AIC  
## - male 1 246.98 258.98  
## <none> 246.78 260.78  
## - race 1 254.91 266.91  
## - multiple.offenses 1 267.44 279.44  
## - state 3 332.58 340.58  
##   
## 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 \*\*\*  
## raceOtherwise 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   
## stateVirgina -3.60795 0.63788 -5.656 1.55e-08 \*\*\*  
## multiple.offensesMultipleOffenses 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

#### Task 5

**Create a logistic regression model using the training set to predict “violator” using the variables: state, multiple.offenses, and race. Comment on the quality of this model.Be sure to note which variables are significant.**

It would appear that this model would be pretty decent for the same reason just like the model in the previous task due to the fact the AIC value is pretty low. four out of the final 6 varibles that were used in the final version of the backward stepwise method were significant. Those four were race otherwise, the state of Kentucky, the state of Virgina, and multiple offenses.

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   
## stateVirgina -3.60795 0.63788 -5.656 1.55e-08 \*\*\*  
## multiple.offensesMultipleOffenses 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

#### Task 6

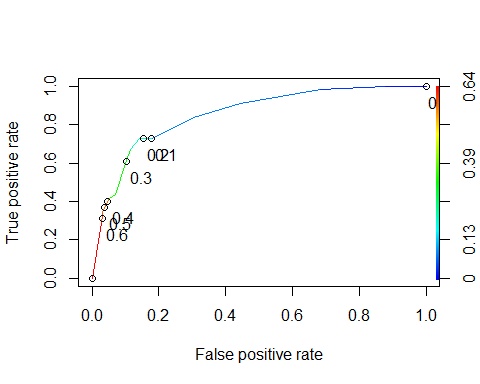
**What is the predicted probability of parole violation of the two following parolees? Parolee1: Louisiana with multiple offenses and white race Parolee2: Kentucky with no multiple offenses and other race**

Parolee1 = train %>% filter(state == "Louisiana")  
Parolee1 = train %>% filter(multiple.offenses == "1")  
Parolee1 = train %>% filter(race == "1")  
Parolee2 = train %>% filter(state == "Kentucky")  
Parolee2 = train %>% filter(multiple.offenses == "0")  
Parolee2 = train %>% filter(race == "2")

#### Task 7

**Develop an ROC curve and determine the probability threshold that best balances specificity and sensitivity (on the training set).**

predictions = predict(mod2, newdata=train, type="response")  
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

**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?**

The accuracy of the model is 0.8435518, the sensitivity of the model is 0.7272727 and the specificity of the model is 0.8588517. The implications of incorrectly classifying a parolee would mean that the model quality would suffer and impact the ROC curve.

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

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

##   
## FALSE TRUE  
## NonViolated 359 59  
## Violated 15 40

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

## [1] 0.8435518

#### Task 9

**Identify a probability threshold (via trial-and-error) that best maximizes accuracy on the training set.**

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

##   
## FALSE TRUE  
## NonViolated 405 13  
## Violated 36 19

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

## [1] 0.8964059

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

##   
## FALSE TRUE  
## NonViolated 406 12  
## Violated 39 16

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

## [1] 0.8921776

#### Task 10

**Use your probability threshold from Task 9 to determine accuracy of the model on the testing set.**

The probability threshold of .6 verifies that the model is classified as having a positive classification as the probability of .89 exceeds the identified threshold.