## Module 3: Assignment 2 – Classification with Logistic Regression

# Lea Shipley

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")

## 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()  
## )

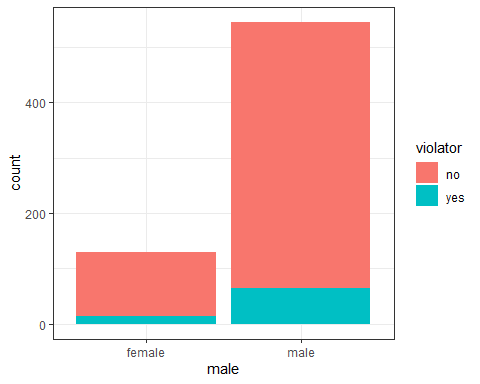
parole = parole %>% mutate(male = as\_factor(as.numeric(male))) %>%  
mutate(male = fct\_recode(male,"male" = "1","female" = "0")) %>%  
 mutate(race = as\_factor(as.numeric(race))) %>%  
 mutate(race = fct\_recode(race, "white" = "1", "other" = "2")) %>%  
 mutate(state = as\_factor(as.numeric(state))) %>%  
 mutate(state = fct\_recode(state, "Kentucky" = "2", "Louisiana" = "3", "Virginia" = "4", "other" = "1")) %>%  
 mutate(crime = as\_factor(as.numeric(crime))) %>%  
 mutate(crime = fct\_recode(crime, "larceny" = "2", "drugs" = "3", "driving" = "4", "other" = "1")) %>%  
 mutate(multiple.offenses = as\_factor(as.numeric(multiple.offenses))) %>%  
 mutate(multiple.offenses = fct\_recode(multiple.offenses, "yes" = "1", "no" = "0")) %>%  
 mutate(violator = as.factor(as.numeric(violator))) %>%   
 mutate(violator = fct\_recode(violator, "yes" = "1", "no" = "0"))

**Task 1**

set.seed(12345)   
train.rows = createDataPartition(y = parole$violator, p=0.7, list = FALSE) #70% in training  
train = parole[train.rows,]   
test = parole[-train.rows,]

**Task 2**

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

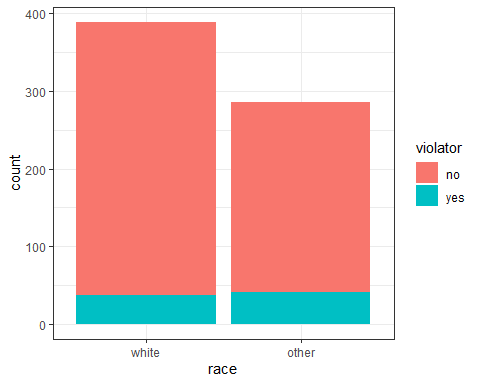


t1 = table(parole$violator, parole$male) #create a table object  
prop.table(t1, margin = 2 ) #crosstab with proportions

##   
## female male  
## no 0.8923077 0.8825688  
## yes 0.1076923 0.1174312

Males do have a slightly higher probability of violating parole than females.

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

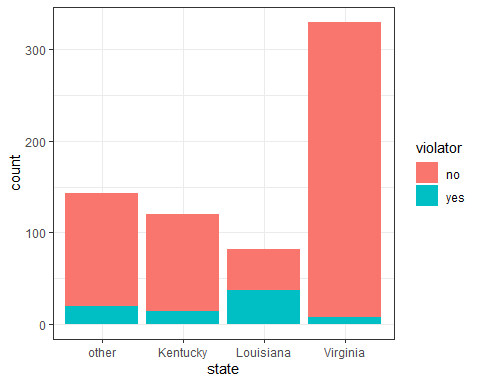


t2 = table(parole$violator, parole$race) #create a table object  
prop.table(t2, margin = 2 ) #crosstab with proportions

##   
## white other  
## no 0.90488432 0.85664336  
## yes 0.09511568 0.14335664

White parolees are less likely to violate parole (0.09 vs. 0.14 for other races).

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

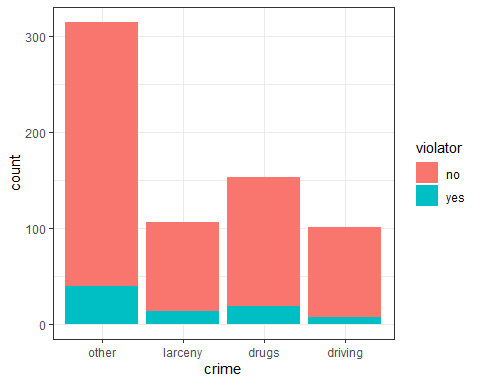


t3 = table(parole$violator, parole$state) #create a table object  
prop.table(t3, margin = 2 ) #crosstab with proportions

##   
## other Kentucky Louisiana Virginia  
## no 0.86013986 0.88333333 0.54878049 0.97878788  
## yes 0.13986014 0.11666667 0.45121951 0.02121212

State seems to be a significant predictor of violating parole, as there is a high change (0.45) that someone in Louisana will violate parole, and a very low chance (0.02) in virginia.

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

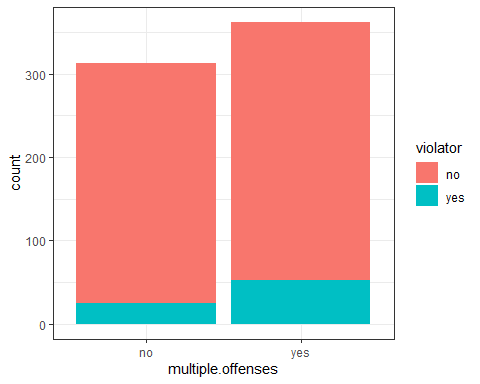


t4 = table(parole$violator, parole$crime) #create a table object  
prop.table(t4, margin = 2 ) #crosstab with proportions

##   
## other larceny drugs driving  
## no 0.87619048 0.87735849 0.87581699 0.93069307  
## yes 0.12380952 0.12264151 0.12418301 0.06930693

The type of crime does not seem to be a very significant predictor of “violator.” Larceny, drug, and “other” charges all have values very close together, while a driving-related offense does have less of a probability of parole violation.

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



t5 = table(parole$violator, parole$multiple.offenses) #create a table object  
prop.table(t5, margin = 2 ) #crosstab with proportions

##   
## no yes  
## no 0.9201278 0.8535912  
## yes 0.0798722 0.1464088

Having multiple offenses is also a good predictor of parole violation.

**Task 3**

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

State appears to be the most predictive variable based on the visualizations and tables. The logistic regression model using “state” as the predictor has a AIC value of 390.89, which doesn’t seem terribly high. P-values for Louisiana and Virginia are statistically significant.

**Task 4**

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) -2.34460 0.52858 -4.436 9.18e-06 \*\*\*  
## malemale -0.06636 0.40761 -0.163 0.87068   
## raceother 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   
## stateVirginia -3.70319 0.65031 -5.695 1.24e-08 \*\*\*  
## crimelarceny 0.37171 0.51172 0.726 0.46760   
## crimedrugs -0.16790 0.40729 -0.412 0.68017   
## crimedriving -0.65912 0.67618 -0.975 0.32967   
## multiple.offensesyes 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") #use ~1 to build an empty model   
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

#backward  
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 \*\*\*  
## 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.offensesyes 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

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  
## + multiple.offenses 1 335.02 339.02  
## + race 1 336.51 340.51  
## <none> 340.04 342.04  
## + crime 3 335.07 343.07  
## + male 1 339.72 343.72  
##   
## 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  
## + 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  
## + male 1 254.91 266.91  
## + crime 3 252.75 268.75  
##   
## Step: AIC=258.98  
## violator ~ state + multiple.offenses + race  
##   
## Df Deviance AIC  
## <none> 246.98 258.98  
## + male 1 246.78 260.78  
## + 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.offensesyes 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

Both the forward stepwise and backward stepwise models use “race,” “state,” and “multiple.offenses” as the predictor variables. The state of Virginia, having multiple offenses, and race of “other” are the statistically significant variables. Interestingly, the state of Louisiana is not statistically significant. These models do have a lower AIC value than the model using “state” as the only predictor variable. The model includes “multiple.offenses,” which intuitively would seem like a good predictor. And, when considering the visualizations, these variables combined do seem like they would be good 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.offensesyes 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 for this model is the same as the forward stepwise and backward stepwise models in the previous step. Race(other), having multiple offenses, and the state of Virginia are significant.

**Task 6**

newdata = data.frame(state = "Louisiana", multiple.offenses = "yes", race = "white", SibSp = 3)  
predict(forwardmod, newdata, type="response")

## 1   
## 0.3379961

The predicted probability of parole violation for Parolee 1 (Louisiana with multiple offenses and white race) is 0.34.

newdata = data.frame(state = "Kentucky", multiple.offenses = "no", race = "other", SibSp = 3)  
predict(forwardmod, newdata, type="response")

## 1   
## 0.2069629

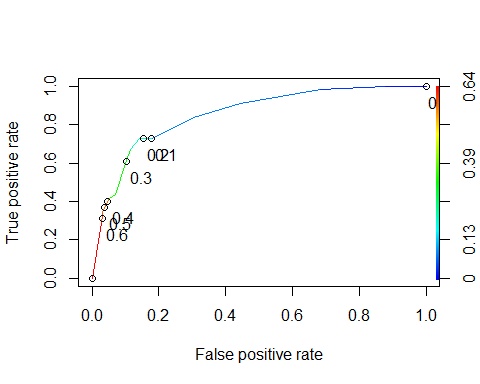
The predicted probability of parole violation for Parolee 2 (Kentucky with no multiple offenses and other race) is 0.21.

**Task 7**

predictions = predict(mod2, type="response") #develop predicted probabilities  
head(predictions)

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

#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))



#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 0.7272727  
## specificity 0.8588517  
## cutoff 0.2069629

The probability threshold that best balances specificity and sensitivity on the training set is 0.21.

#confusion matrix  
t6 = table(train$violator,predictions > 0.2069629)  
t6

##   
## FALSE TRUE  
## no 359 59  
## yes 15 40

(t6[1,1]+t1[2,2])/nrow(train) #calculating accuracy

## [1] 0.8942918

**Task 8**

Given the cutoff of 0.2069629 from Task 7, the accuracy of the model on the training set is 0.89, the sensitivity is 0.73, and the specificity is 0.86. Incorrectly classifying a parolee could result in denying parole for inmates who would be unlikely to violate parole. Or, granting parole to an inmate who is likely to commit another crime and violate parole, which could also have negative implications for society.

**Task 9**

t6 = table(train$violator,predictions > 0.5) # trial and error to maximize accuracy   
t6

##   
## FALSE TRUE  
## no 405 13  
## yes 36 19

(t6[1,1]+t6[2,2])/nrow(train)

## [1] 0.8964059

t6 = table(train$violator,predictions > 0.6) # trial and error to maximize accuracy   
t6

##   
## FALSE TRUE  
## no 406 12  
## yes 39 16

(t6[1,1]+t6[2,2])/nrow(train)

## [1] 0.8921776

t6 = table(train$violator,predictions > 0.4) # trial and error to maximize accuracy   
t6

##   
## FALSE TRUE  
## no 405 13  
## yes 36 19

(t6[1,1]+t6[2,2])/nrow(train)

## [1] 0.8964059

t6 = table(train$violator,predictions > 0.3) # trial and error to maximize accuracy   
t6

##   
## FALSE TRUE  
## no 376 42  
## yes 22 33

(t6[1,1]+t6[2,2])/nrow(train)

## [1] 0.8646934

t6 = table(train$violator,predictions > 0.1) # trial and error to maximize accuracy   
t6

##   
## FALSE TRUE  
## no 350 68  
## yes 15 40

(t6[1,1]+t6[2,2])/nrow(train)

## [1] 0.8245243

**Task 10**

test\_preds = predict(mod2, newdata = test, type = "response")  
head(test\_preds)

## 1 2 3 4 5 6   
## 0.07509978 0.07509978 0.19512504 0.07509978 0.57878248 0.19512504

t7 = table(test$violator,test\_preds > 0.5)   
t7

##   
## FALSE TRUE  
## no 174 5  
## yes 15 8

(t7[1,1]+t7[2,2])/nrow(test)

## [1] 0.9009901