# Logistic Regression (Classification)

## Nicole Westrick

### BAN 502: Mod.3 - Assign. 2

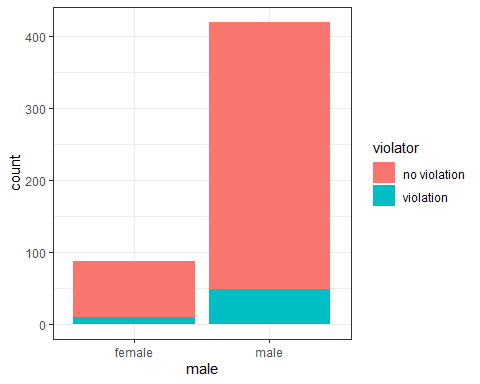
parole = parole %>%  
 mutate(male = as\_factor(male)) %>%  
 mutate(male = fct\_recode(male, "male" = "1", "female" = "0")) %>%  
 mutate(race = as\_factor(race)) %>%  
 mutate(race = fct\_recode(race, "white" = "1", "otherwise" = "2")) %>%  
 mutate(state = as\_factor(state)) %>%  
 mutate(state = fct\_recode(state, "Kentucky" = "2", "Louisiana" = "3", "Virginia" = "4", "Other" = "1")) %>%  
 mutate(crime = as\_factor(crime)) %>%  
 mutate(crime = fct\_recode(crime, "larceny" = "2", "drug-related" = "3", "driving-related" = "4", "other" = "1")) %>%  
 mutate(multiple.offenses = as\_factor(multiple.offenses)) %>%  
 mutate(multiple.offenses = fct\_recode(multiple.offenses, "multiple" = "1", "otherwise" = "0")) %>%  
 mutate(violator = as\_factor(violator)) %>%  
 mutate(violator = fct\_recode(violator, "violation" = "1", "no violation" = "0"))

**Task 1:**

set.seed(12345)  
parole\_split = initial\_split(parole, prob = 0.70, strata = violator)  
train = training(parole\_split)  
test = testing(parole\_split)

**Task 2:**

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

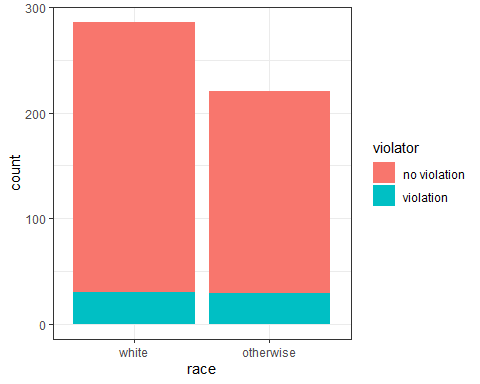


t1 = table(train$violator, train$male)   
prop.table(t1, margin = 2 )

##   
## female male  
## no violation 0.8850575 0.8833333  
## violation 0.1149425 0.1166667

Gender does not appear to be a predictor of violator.

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

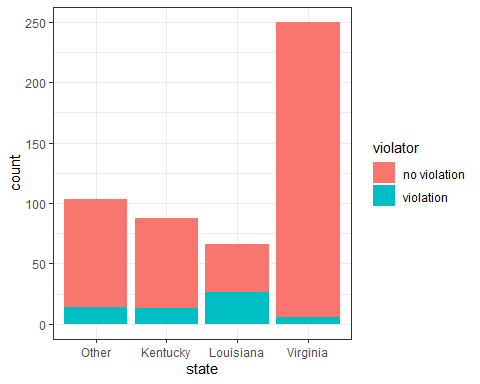


t2 = table(train$violator, train$race)   
prop.table(t2, margin = 2 )

##   
## white otherwise  
## no violation 0.8951049 0.8687783  
## violation 0.1048951 0.1312217

Race does not appear to be a significant predictor of violator.

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

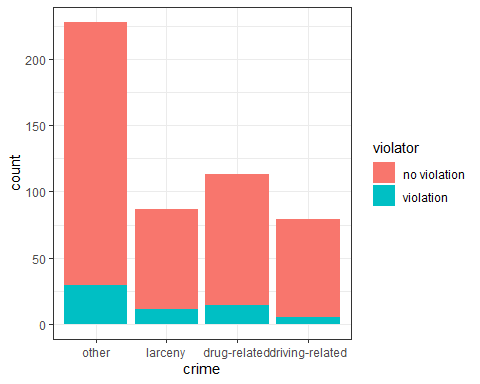


t3 = table(train$violator, train$state)   
prop.table(t3, margin = 2 )

##   
## Other Kentucky Louisiana Virginia  
## no violation 0.8640777 0.8522727 0.6060606 0.9760000  
## violation 0.1359223 0.1477273 0.3939394 0.0240000

State does appear to be a predictor of violator, specifically Louisiana and Virginia. Someone in Louisiana is more likely to violate parole than someone in Virginia.

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

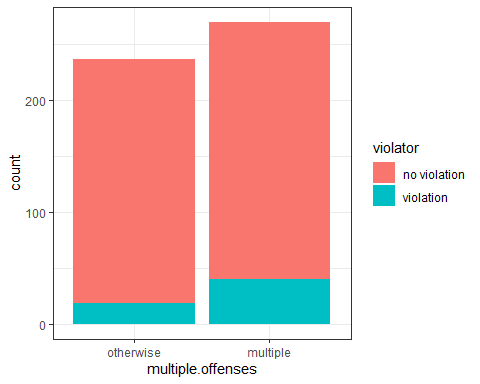


t4 = table(train$violator, train$crime)   
prop.table(t4, margin = 2 )

##   
## other larceny drug-related driving-related  
## no violation 0.87280702 0.87356322 0.87610619 0.93670886  
## violation 0.12719298 0.12643678 0.12389381 0.06329114

Crime does not appear to be a significant predictor of violator.

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

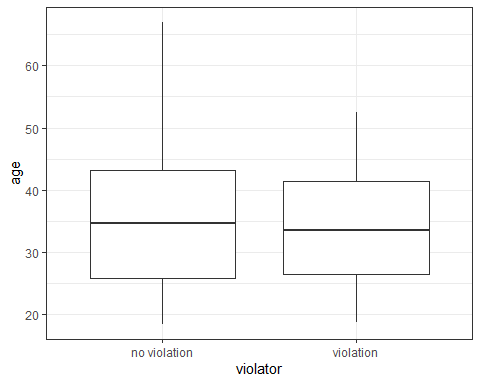


t5 = table(train$violator, train$multiple.offenses)   
prop.table(t5, margin = 2 )

##   
## otherwise multiple  
## no violation 0.91983122 0.85185185  
## violation 0.08016878 0.14814815

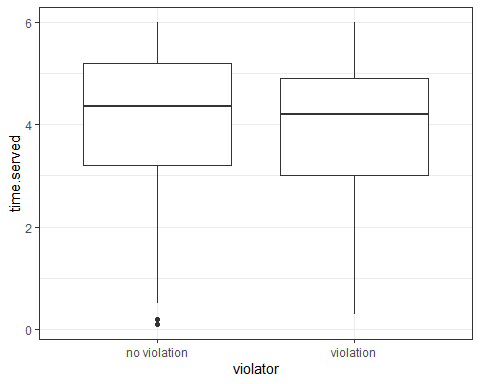
Multiple offenses does not appear to be a significant predictor of violator.

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



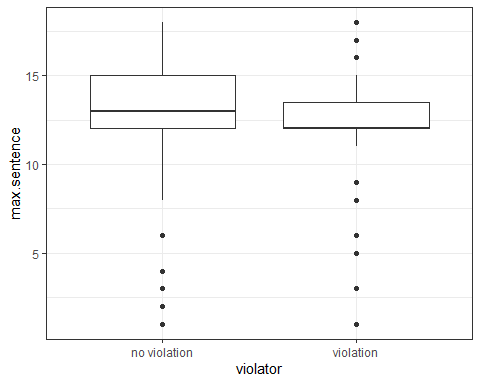
Age does not appear to be a significant predictor of violator. However, it does appear that someone over the age of 55 is less likely to violate.

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



Time served does not appear to be a predictor of violator.

ggplot(train, aes(x=violator, y= max.sentence)) + geom\_boxplot() + theme\_bw()



Max sentence does not appear to be a predictor of violator.

**Task 3:**

train\_model =   
 logistic\_reg() %>%   
 set\_engine("glm")   
  
train\_recipe = recipe(violator ~ state, train) %>%  
 step\_dummy(all\_nominal(), -all\_outcomes())   
  
logreg\_wf = workflow() %>%  
 add\_recipe(train\_recipe) %>%   
 add\_model(train\_model)  
  
train\_fit = fit(logreg\_wf, train)  
  
summary(train\_fit$fit$fit$fit)

##   
## Call:  
## stats::glm(formula = ..y ~ ., family = stats::binomial, data = data)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.0008 -0.5405 -0.2204 -0.2204 2.7312   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -1.84958 0.28751 -6.433 1.25e-10 \*\*\*  
## state\_Kentucky 0.09704 0.41584 0.233 0.815481   
## state\_Louisiana 1.41880 0.38226 3.712 0.000206 \*\*\*  
## state\_Virginia -1.85583 0.50341 -3.686 0.000227 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 364.67 on 506 degrees of freedom  
## Residual deviance: 300.70 on 503 degrees of freedom  
## AIC: 308.7  
##   
## Number of Fisher Scoring iterations: 6

Both Louisiana and Virginia are significant predictors of violators. The AIC value is 308.7.

**Task 4:**

train\_model =   
 logistic\_reg() %>%   
 set\_engine("glm")   
  
train\_recipe = recipe(violator ~ state + multiple.offenses, train) %>%  
 step\_dummy(all\_nominal(), -all\_outcomes())   
  
logreg\_wf = workflow() %>%  
 add\_recipe(train\_recipe) %>%   
 add\_model(train\_model)  
  
train\_fit2 = fit(logreg\_wf, train)  
  
summary(train\_fit2$fit$fit$fit)

##   
## Call:  
## stats::glm(formula = ..y ~ ., family = stats::binomial, data = data)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.2000 -0.4952 -0.2460 -0.2460 2.6505   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -2.4971 0.3565 -7.005 2.47e-12 \*\*\*  
## state\_Kentucky 0.4601 0.4451 1.034 0.3013   
## state\_Louisiana 0.9181 0.4114 2.231 0.0257 \*   
## state\_Virginia -2.6172 0.5332 -4.908 9.20e-07 \*\*\*  
## multiple.offenses\_multiple 1.6319 0.3663 4.456 8.37e-06 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 364.67 on 506 degrees of freedom  
## Residual deviance: 279.41 on 502 degrees of freedom  
## AIC: 289.41  
##   
## Number of Fisher Scoring iterations: 6

When multiple.offenses is added to the model, the AIC value is lowered to 289.41. The significance of Louisiana is also lowered. The best predictors of violator now appear to be the state of Virginia and multiple offenses. This model’s quality is better than the first.

train\_model =   
 logistic\_reg() %>%   
 set\_engine("glm")   
  
train\_recipe = recipe(violator ~ state + multiple.offenses + age, train) %>%  
 step\_dummy(all\_nominal(), -all\_outcomes())   
  
logreg\_wf = workflow() %>%  
 add\_recipe(train\_recipe) %>%   
 add\_model(train\_model)  
  
train\_fit3 = fit(logreg\_wf, train)  
  
summary(train\_fit3$fit$fit$fit)

##   
## Call:  
## stats::glm(formula = ..y ~ ., family = stats::binomial, data = data)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.2020 -0.4941 -0.2463 -0.2455 2.6512   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -2.5088292 0.6469655 -3.878 0.000105 \*\*\*  
## age 0.0003256 0.0149339 0.022 0.982608   
## state\_Kentucky 0.4600596 0.4450963 1.034 0.301315   
## state\_Louisiana 0.9191133 0.4141928 2.219 0.026484 \*   
## state\_Virginia -2.6167570 0.5336153 -4.904 9.40e-07 \*\*\*  
## multiple.offenses\_multiple 1.6320914 0.3663914 4.455 8.41e-06 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 364.67 on 506 degrees of freedom  
## Residual deviance: 279.40 on 501 degrees of freedom  
## AIC: 291.4  
##   
## Number of Fisher Scoring iterations: 6

When age is added to the model, along with state and multiple.offenses, the AIC is increased to 291.4. This indicates that the model that includes only state and multiple.offenses has better quality. Age is not a significant predictor.

**Task 5:**

train\_model =   
 logistic\_reg() %>%   
 set\_engine("glm")   
  
train\_recipe = recipe(violator ~ state + multiple.offenses + race, train) %>%  
 step\_dummy(all\_nominal(), -all\_outcomes())   
  
logreg\_wf = workflow() %>%  
 add\_recipe(train\_recipe) %>%   
 add\_model(train\_model)  
  
train\_fit4 = fit(logreg\_wf, train)  
  
summary(train\_fit4$fit$fit$fit)

##   
## Call:  
## stats::glm(formula = ..y ~ ., family = stats::binomial, data = data)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.2598 -0.4718 -0.2675 -0.2173 2.7414   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -2.5431 0.3579 -7.106 1.20e-12 \*\*\*  
## state\_Kentucky 0.4036 0.4470 0.903 0.367   
## state\_Louisiana 0.7135 0.4481 1.592 0.111   
## state\_Virginia -2.7907 0.5570 -5.010 5.43e-07 \*\*\*  
## multiple.offenses\_multiple 1.5998 0.3684 4.342 1.41e-05 \*\*\*  
## race\_otherwise 0.4215 0.3527 1.195 0.232   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 364.67 on 506 degrees of freedom  
## Residual deviance: 277.99 on 501 degrees of freedom  
## AIC: 289.99  
##   
## Number of Fisher Scoring iterations: 6

A model including state, multiple.offenses, and race as predictors of violator shows only the state of Virginia and multiple offenses as significant predictors. This model has an AIC value of 289.99, which is just slightly higher than the model with only state and multiple.offenses as predictors.

**Task 6:**

newdata = data.frame(state = "Louisiana", multiple.offenses = "multiple", race = "white")  
predict(train\_fit4, newdata, type="prob")

## # A tibble: 1 x 2  
## `.pred\_no violation` .pred\_violation  
## <dbl> <dbl>  
## 1 0.557 0.443

newdata2 = data.frame(state = "Kentucky", multiple.offenses = "otherwise", race = "otherwise")  
predict(train\_fit4, newdata2, type="prob")

## # A tibble: 1 x 2  
## `.pred\_no violation` .pred\_violation  
## <dbl> <dbl>  
## 1 0.848 0.152

Parolee 1 has a 44% chance of violating parole. Parolee 2 has a 15% chance of violating parole.

**Task 7:**

train\_model =   
 logistic\_reg(mode = "classification") %>%   
 set\_engine("glm")   
  
train\_recipe = recipe(violator ~., train)  
  
logreg\_wf = workflow() %>%  
 add\_recipe(train\_recipe) %>%   
 add\_model(train\_model)  
  
train\_fit5 = fit(logreg\_wf, train)  
  
summary(train\_fit5$fit$fit$fit)

##   
## Call:  
## stats::glm(formula = ..y ~ ., family = stats::binomial, data = data)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.5189 -0.4324 -0.2695 -0.1793 2.8067   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -3.039052 1.116175 -2.723 0.00647 \*\*   
## malemale 0.281802 0.439946 0.641 0.52182   
## raceotherwise 0.443687 0.358593 1.237 0.21598   
## age 0.004903 0.015346 0.320 0.74933   
## stateKentucky 0.544529 0.480560 1.133 0.25717   
## stateLouisiana 0.774125 0.508455 1.523 0.12788   
## stateVirginia -2.948822 0.576468 -5.115 3.13e-07 \*\*\*  
## time.served -0.079212 0.113591 -0.697 0.48559   
## max.sentence 0.039556 0.048019 0.824 0.41008   
## multiple.offensesmultiple 1.590077 0.372029 4.274 1.92e-05 \*\*\*  
## crimelarceny 0.316021 0.473564 0.667 0.50456   
## crimedrug-related -0.394782 0.410285 -0.962 0.33594   
## crimedriving-related -0.311261 0.562188 -0.554 0.57981   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 364.67 on 506 degrees of freedom  
## Residual deviance: 274.08 on 494 degrees of freedom  
## AIC: 300.08  
##   
## Number of Fisher Scoring iterations: 6

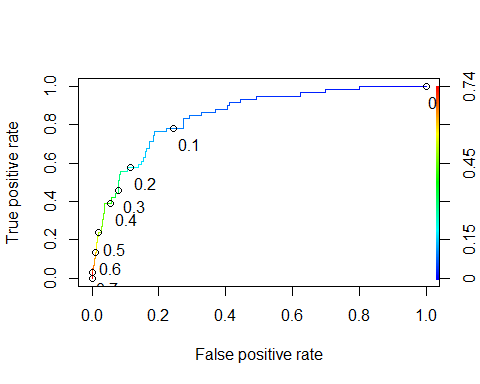
predictions = predict(train\_fit5, train, type="prob")   
head(predictions)

## # A tibble: 6 x 2  
## `.pred\_no violation` .pred\_violation  
## <dbl> <dbl>  
## 1 0.933 0.0673  
## 2 0.961 0.0395  
## 3 0.916 0.0842  
## 4 0.897 0.103   
## 5 0.897 0.103   
## 6 0.944 0.0560

predictions = predict(train\_fit5, train, type="prob")[2]  
head(predictions)

## # A tibble: 6 x 1  
## .pred\_violation  
## <dbl>  
## 1 0.0673  
## 2 0.0395  
## 3 0.0842  
## 4 0.103   
## 5 0.103   
## 6 0.0560

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] 0.8471928

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 0.7627119  
## specificity 0.8125000  
## cutoff 0.1258245

A model including all variables as predictors of violator verifies that Virginia and multiple offenses are the only significant predictors of violator. This model has an AIC value of 300.08, which is in the same range as other tested models. After developing an ROC curve, the probability threshold that best balances sensitivity and specificity is 0.1258. The AUC is 0.8472.

**Task 8:**

t1 = table(train$violator,predictions > 0.1258245)  
t1

##   
## FALSE TRUE  
## no violation 364 84  
## violation 15 44

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

## [1] 0.8047337

Using the cutoff 0.1258245, the accuracy of the model is calculated as 0.8047. Using the same cutoff, the sensitivity is 0.7627 and the specificity is 0.8125.

There could be serious implications of incorrectly classifying a parolee. Longer sentences and probation guidelines are often determined based on an offenders past and their probability of a repeat offense, or violation. Incorrectly classifying someone as having a low probability of violation could put their community and lives at risk.

**Task 9:**

t1 = table(train$violator,predictions > 0.3)  
t1

##   
## FALSE TRUE  
## no violation 413 35  
## violation 32 27

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

## [1] 0.8678501

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

##   
## FALSE TRUE  
## no violation 444 4  
## violation 52 7

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

## [1] 0.8895464

A probability threshold of 0.6 maximizes the accuracy of the model (0.8895).

**Task 10:**

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

##   
## FALSE TRUE  
## no violation 444 4  
## violation 52 7

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

## [1] 2.684524

Using the probability threshold that maximizes the accuracy of the training model on the testing results in an accuracy of 2.6845.