library(e1071)  
library(ROCR)  
library(tidymodels)

## Registered S3 method overwritten by 'tune':  
## method from   
## required\_pkgs.model\_spec parsnip

## -- Attaching packages -------------------------------------- tidymodels 0.1.4 --

## v broom 0.7.10 v recipes 0.1.17  
## v dials 0.0.10 v rsample 0.1.1   
## v dplyr 1.0.7 v tibble 3.1.6   
## v ggplot2 3.3.5 v tidyr 1.1.4   
## v infer 1.0.0 v tune 0.1.6   
## v modeldata 0.1.1 v workflows 0.2.4   
## v parsnip 0.1.7 v workflowsets 0.1.0   
## v purrr 0.3.4 v yardstick 0.0.9

## -- Conflicts ----------------------------------------- tidymodels\_conflicts() --  
## x purrr::discard() masks scales::discard()  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()  
## x rsample::permutations() masks e1071::permutations()  
## x recipes::step() masks stats::step()  
## x tune::tune() masks e1071::tune()  
## \* Learn how to get started at https://www.tidymodels.org/start/

library(tidyverse)

## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v readr 2.1.0 v forcats 0.5.1  
## v stringr 1.4.0

## -- Conflicts ------------------------------------------ tidyverse\_conflicts() --  
## x readr::col\_factor() masks scales::col\_factor()  
## x purrr::discard() masks scales::discard()  
## x dplyr::filter() masks stats::filter()  
## x stringr::fixed() masks recipes::fixed()  
## x dplyr::lag() masks stats::lag()  
## x readr::spec() masks yardstick::spec()

library(dplyr)

### Task 2: I used bar charts to visualize sex, race, state, and other categorical variables. I used box plots for variables with quanatative data, including max sentence, age, and time served. Using these graphs allowed me to see how the variables impacted the parole violation rate.

### Task 3: State is the most predictive of violator. The AIC is around 390, and the coefficient for Louisiana is positive, which makes sense given that parolees are more likely to violate in Louisiana versus any other state.

### Task 4: The AIC went down for the model that includes state, multiple offenses, and crime, with a value of around 266. Multiple offenses and Virgina were both significant variables, which I don’t find intuitive because Louisiana has a higher rate of parole violation.

### Task 5: The AIC went down to 256, which is an improvement. Virginia, Kentucky, race Otherwise (i.e. not white), and multiple offenses (yes) were all significant variables.

### Task 6: the predicted probability for violating parole for Parolee 1 is 0.3311299.The predicted probability for violating parole for Parolee 2 is 0.2015788

### Task 7: The probability threshold that best balances the training set is as follows: sensitivity 0.7222222 and specificity 0.8369305 with a cutoff at cutoff 0.2015788.

### Task 8: Disclaimer- there is an excellent chance I just missed this. For accuracy, 396/471 is .84. For sensitivity, 36/54 is .66. For specificity, 360/417 is .86. The implications of incorrectly classifying a parolee can do one of two ways. Either parolees who would not violate parole are incorrectly watched/given certain restictions, which limit their freedoms in real ways. Alternatively, not classifying a parolee as a violator when they end up violating parole will lead them back to prison or illegal activities. Correctly classifying parolees lets them be monitored appropriately, which is good for their rights as citizens and the justice system.

### Task 9: between .6 and .65 has a better accuracy, but the data is imbalanced because there are more non-violators than violators, so you have to be mindful in accessing model quality.

### Task 10: the accuracy of the model from the train set when run on the test set is 0.8970588. This is similar to the accuracy of the train set, which suggest that the model does not overfit the dataset.

parole <- read\_csv("~/P.Module3/P.Module3/Assigment 3.2/parole.csv")

## Rows: 675 Columns: 9

## -- Column specification --------------------------------------------------------  
## Delimiter: ","  
## dbl (9): male, race, age, state, time.served, max.sentence, multiple.offense...

##   
## i Use `spec()` to retrieve the full column specification for this data.  
## i Specify the column types or set `show\_col\_types = FALSE` to quiet this message.

View(parole)

parole <- parole %>% mutate(male=as.factor(male))  
parole <- parole %>% mutate(race=as.factor(race))  
parole <- parole %>% mutate(state=as.factor(state))  
parole <- parole %>% mutate(crime=as.factor(crime))  
parole <- parole %>% mutate(multiple.offenses=as.factor(multiple.offenses))  
parole <- parole %>% mutate(violator=as.factor(violator))

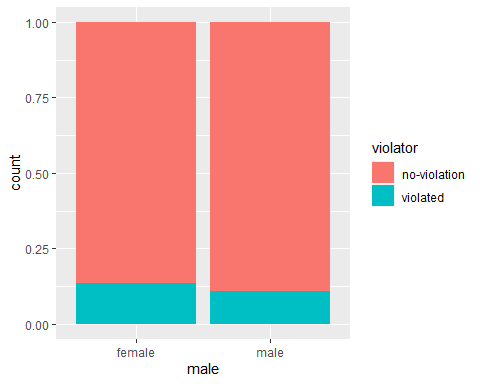
parole <- parole %>%  
 mutate(male= fct\_recode(male,  
 "male"="1",  
 "female"="0"))  
parole <- parole %>%  
 mutate(race= fct\_recode(race,  
 "white"="1",  
 "otherwise"="2"))  
  
parole <- parole %>%  
 mutate(state= fct\_recode(state,  
 "Kentucky"="2",  
 "Louisiana"="3",  
 "Virginia"="4",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

Task 1

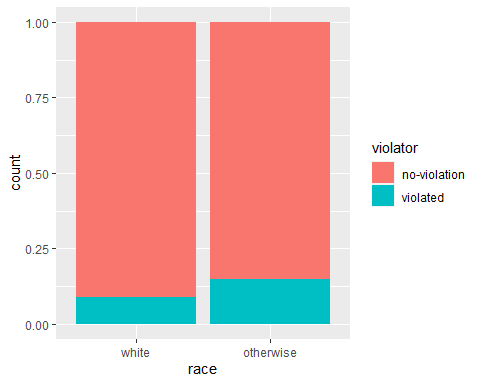
set.seed(12345)  
parole\_split <- initial\_split(parole, prop=.70, strata=violator)  
train=training(parole\_split)  
test=testing(parole\_split)

Task 2

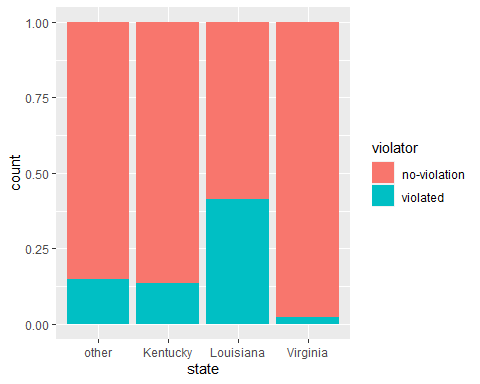
ggplot(train, aes(x=male, fill=violator))+geom\_bar(position="fill")



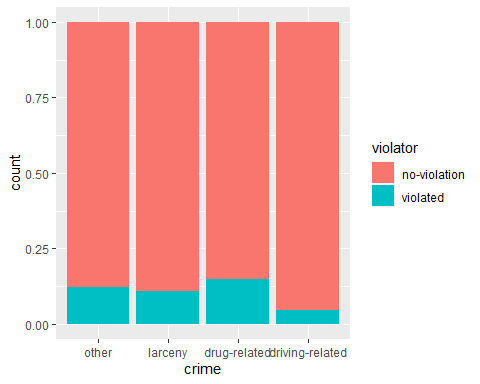
ggplot(train, aes(x=race, fill=violator))+geom\_bar(position="fill")



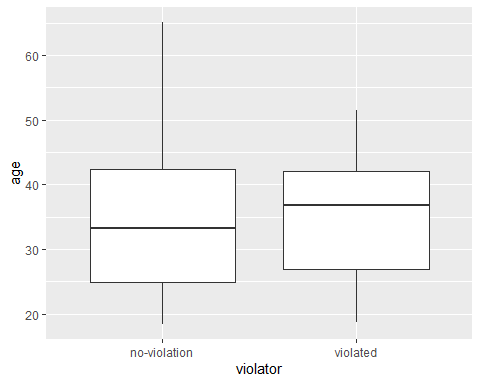
ggplot(train, aes(x=state, fill=violator))+geom\_bar(position="fill")



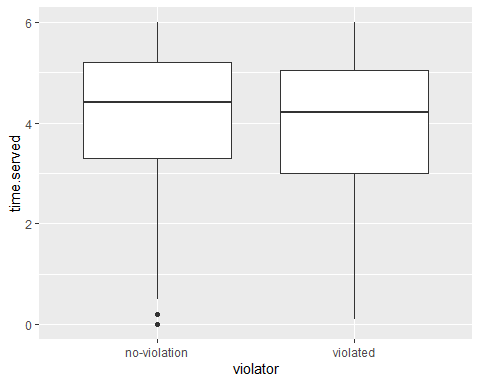
ggplot(train, aes(x=crime, fill=violator))+geom\_bar(position="fill")



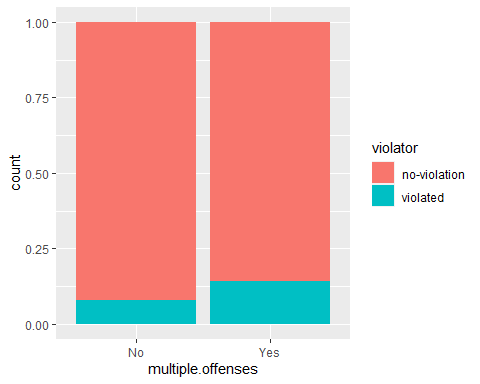
ggplot(train, aes(x=violator, y=age))+geom\_boxplot()



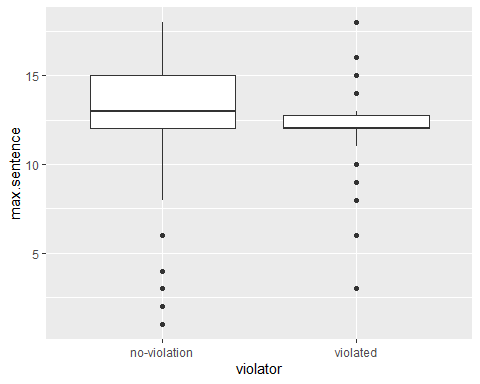
ggplot(train, aes(x=violator, y=time.served))+geom\_boxplot()



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



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



Building a logistic regression model with State

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

summary(parole\_fit$fit$fit$fit)

##   
## Call:  
## stats::glm(formula = ..y ~ ., family = stats::binomial, data = data)  
##   
## 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 \*\*\*  
## state\_Kentucky -0.2079 0.3728 -0.558 0.577   
## state\_Louisiana 1.6207 0.3277 4.946 7.58e-07 \*\*\*  
## state\_Virginia -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

Task 4 prediction

parole\_model2 =  
 logistic\_reg() %>%  
 set\_engine("glm")  
  
parole\_state\_recipe2=recipe(violator ~ state + crime + multiple.offenses, train) %>%  
 step\_dummy(all\_nominal(), -all\_outcomes())  
  
logreg\_wf2= workflow() %>%  
 add\_recipe(parole\_state\_recipe2) %>%  
 add\_model(parole\_model2)  
  
parole\_fit2=fit(logreg\_wf2, train)

summary(parole\_fit2$fit$fit$fit)

##   
## Call:  
## stats::glm(formula = ..y ~ ., family = stats::binomial, data = data)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.3464 -0.4576 -0.2279 -0.1556 2.9734   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -2.20379 0.38929 -5.661 1.50e-08 \*\*\*  
## state\_Kentucky 0.06103 0.47710 0.128 0.8982   
## state\_Louisiana 0.71580 0.43478 1.646 0.0997 .   
## state\_Virginia -2.98967 0.58276 -5.130 2.89e-07 \*\*\*  
## crime\_larceny 0.32155 0.49807 0.646 0.5185   
## crime\_drug.related -0.04392 0.39984 -0.110 0.9125   
## crime\_driving.related -0.77015 0.67476 -1.141 0.2537   
## multiple.offenses\_Yes 1.55527 0.38715 4.017 5.89e-05 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 335.47 on 470 degrees of freedom  
## Residual deviance: 250.57 on 463 degrees of freedom  
## AIC: 266.57  
##   
## Number of Fisher Scoring iterations: 6

parole\_model3 =  
 logistic\_reg() %>%  
 set\_engine("glm")  
  
parole\_state\_recipe3=recipe(violator ~ state + race + multiple.offenses, train) %>%  
 step\_dummy(all\_nominal(), -all\_outcomes())  
  
logreg\_wf3= workflow() %>%  
 add\_recipe(parole\_state\_recipe3) %>%  
 add\_model(parole\_model3)  
  
parole\_fit3=fit(logreg\_wf3, train)

summary(parole\_fit3$fit$fit$fit)

##   
## Call:  
## stats::glm(formula = ..y ~ ., family = stats::binomial, data = data)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.3572 -0.4013 -0.2705 -0.1557 2.9726   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -2.47873 0.36085 -6.869 6.46e-12 \*\*\*  
## state\_Kentucky -0.01418 0.46926 -0.030 0.97590   
## state\_Louisiana 0.11876 0.49950 0.238 0.81206   
## state\_Virginia -3.58422 0.63848 -5.614 1.98e-08 \*\*\*  
## race\_otherwise 1.11646 0.39092 2.856 0.00429 \*\*   
## multiple.offenses\_Yes 1.65689 0.39652 4.179 2.93e-05 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 335.47 on 470 degrees of freedom  
## Residual deviance: 244.52 on 465 degrees of freedom  
## AIC: 256.52  
##   
## Number of Fisher Scoring iterations: 6

predictions for task 6

newdata=data.frame(state="Louisiana", multiple.offenses="Yes", race="white")  
predict(parole\_fit3, newdata, type="prob")

## # A tibble: 1 x 2  
## `.pred\_no-violation` .pred\_violated  
## <dbl> <dbl>  
## 1 0.669 0.331

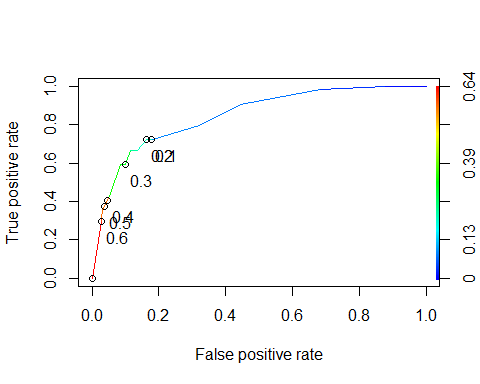
newdata2=data.frame(state="Kentucky", multiple.offenses="No", race="otherwise")  
predict(parole\_fit3, newdata2, type="prob")

## # A tibble: 1 x 2  
## `.pred\_no-violation` .pred\_violated  
## <dbl> <dbl>  
## 1 0.798 0.202

predictions=predict(parole\_fit3, train, type="prob")[2]  
head(predictions)

## # A tibble: 6 x 1  
## .pred\_violated  
## <dbl>  
## 1 0.0774  
## 2 0.204   
## 3 0.204   
## 4 0.0774  
## 5 0.0774  
## 6 0.204

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] 0.8460121

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 0.7222222  
## specificity 0.8369305  
## cutoff 0.2015788

t1 = table(train$violator,predictions > 0.2015788)  
t1

##   
## FALSE TRUE  
## no-violation 360 57  
## violated 18 36

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

##   
## FALSE TRUE  
## no-violation 404 13  
## violated 35 19

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

## [1] 0.8980892

t2 = table(train$violator,predictions > 0.6)  
t2

##   
## FALSE TRUE  
## no-violation 405 12  
## violated 38 16

(t2[1,1]+t1[2,2])/nrow(train)

## [1] 0.9002123

t3 = table(train$violator,predictions > .65)  
  
t3

##   
## FALSE  
## no-violation 417  
## violated 54

(t3[1,1]+t1[2,2])/nrow(train)

## [1] 0.92569

predict\_test <- predict(parole\_fit3,test)  
predict\_test

## # A tibble: 204 x 1  
## .pred\_class   
## <fct>   
## 1 no-violation  
## 2 no-violation  
## 3 no-violation  
## 4 no-violation  
## 5 violated   
## 6 no-violation  
## 7 no-violation  
## 8 no-violation  
## 9 no-violation  
## 10 no-violation  
## # ... with 194 more rows

parole\_fit3 %>% predict(test) %>% bind\_cols(test) %>% metrics(truth=violator, estimate=.pred\_class)

## # A tibble: 2 x 3  
## .metric .estimator .estimate  
## <chr> <chr> <dbl>  
## 1 accuracy binary 0.897  
## 2 kap binary 0.381