# Module 5 Assignment 1

## BAN 502

### Jennifer Brill

library(tidyverse)

## ── Attaching packages ──────────────────────────────────────────────────────── tidyverse 1.2.1 ──

## ✔ ggplot2 3.1.0 ✔ purrr 0.2.5  
## ✔ tibble 1.4.2 ✔ dplyr 0.7.8  
## ✔ tidyr 0.8.2 ✔ stringr 1.3.1  
## ✔ readr 1.1.1 ✔ forcats 0.3.0

## ── Conflicts ─────────────────────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()

library(caret)

## Loading required package: lattice

##   
## Attaching package: 'caret'

## The following object is masked from 'package:purrr':  
##   
## lift

library(nnet)

library(readr)  
parole <- read\_csv("parole (1).csv")

## Parsed with column specification:  
## cols(  
## male = col\_integer(),  
## race = col\_integer(),  
## age = col\_double(),  
## state = col\_integer(),  
## time.served = col\_double(),  
## max.sentence = col\_integer(),  
## multiple.offenses = col\_integer(),  
## crime = col\_integer(),  
## violator = col\_integer()  
## )

parole = parole %>%  
 mutate(male = as\_factor(as.character(male))) %>%  
 mutate(male = fct\_recode(male, "male" = "1", "female" = "0")) %>%  
 mutate(race = as\_factor(as.character(race))) %>%  
 mutate(race = fct\_recode(race, "white" = "1", "otherwise" = "2")) %>%  
 mutate(state = as\_factor(as.character(state))) %>%  
 mutate(state = fct\_recode(state, "kentucky" = "2", "louisiana" = "3", "virginia" = "4", "other" = "1")) %>%  
 mutate(crime = as\_factor(as.character(crime))) %>%  
 mutate(crime = fct\_recode(crime, "larceny" = "2", "drug" = "3", "driving" = "4", "other" = "1")) %>%  
 mutate(multiple.offenses = as\_factor(as.character(multiple.offenses))) %>%  
 mutate(multiple.offenses = fct\_recode(multiple.offenses, "multiple" = "1", "single" = "0")) %>%  
 mutate(violator = as\_factor(as.character(violator))) %>%  
 mutate(violator = fct\_recode(violator, "violated" = "1", "didnotviolate" = "0"))  
str(parole)

## Classes 'tbl\_df', 'tbl' and '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 "other","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 "single","multiple": 1 1 1 1 1 1 1 1 1 1 ...  
## $ crime : Factor w/ 4 levels "driving","drug",..: 1 2 2 3 3 1 2 3 2 4 ...  
## $ violator : Factor w/ 2 levels "didnotviolate",..: 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,]

fitControl = trainControl(method = "cv", number = 10)  
nnetGrid = expand.grid(size = 12, decay = 0.1)  
set.seed(1234)  
nnetBasic = train(violator ~., parole, method = "nnet", tuneGrid = nnetGrid, trControl = fitControl, verbose = FALSE, trace = FALSE)

predNetBasic = predict(nnetBasic, train)  
head(predNetBasic)

## [1] didnotviolate didnotviolate didnotviolate didnotviolate didnotviolate  
## [6] didnotviolate  
## Levels: didnotviolate violated

confusionMatrix(predNetBasic, train$violator, positive = "violated")

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction didnotviolate violated  
## didnotviolate 412 18  
## violated 6 37  
##   
## Accuracy : 0.9493   
## 95% CI : (0.9254, 0.9672)  
## No Information Rate : 0.8837   
## P-Value [Acc > NIR] : 6.784e-07   
##   
## Kappa : 0.7273   
## Mcnemar's Test P-Value : 0.02474   
##   
## Sensitivity : 0.67273   
## Specificity : 0.98565   
## Pos Pred Value : 0.86047   
## Neg Pred Value : 0.95814   
## Prevalence : 0.11628   
## Detection Rate : 0.07822   
## Detection Prevalence : 0.09091   
## Balanced Accuracy : 0.82919   
##   
## 'Positive' Class : violated   
##

This model appears to fit the training set of data well. The accuracy rate is high at 94%. The sensitivity is 67% and specificity is 98%. The no information rate is also high at 88% which tells us that if we assumed everyone in the data set did not violate parole, we would 88% accurate.

fitControl = trainControl(method = "cv", number = 10)  
nnetGrid = expand.grid(size = seq(from = 1, to = 12, by = 1), decay = seq(from = 0.1, to = 0.5, by = 0.1))  
set.seed(1234)  
nnetFit = train(violator ~., parole, method = "nnet", trControl = fitControl, tuneGrid = nnetGrid, trace = FALSE, verbose = FALSE)

predNet = predict(nnetFit, train)  
head(predNet)

## [1] didnotviolate didnotviolate didnotviolate didnotviolate didnotviolate  
## [6] didnotviolate  
## Levels: didnotviolate violated

confusionMatrix(predNet, train$violator, positive = "violated")

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction didnotviolate violated  
## didnotviolate 414 30  
## violated 4 25  
##   
## Accuracy : 0.9281   
## 95% CI : (0.901, 0.9497)  
## No Information Rate : 0.8837   
## P-Value [Acc > NIR] : 0.0009298   
##   
## Kappa : 0.5599   
## Mcnemar's Test P-Value : 1.807e-05   
##   
## Sensitivity : 0.45455   
## Specificity : 0.99043   
## Pos Pred Value : 0.86207   
## Neg Pred Value : 0.93243   
## Prevalence : 0.11628   
## Detection Rate : 0.05285   
## Detection Prevalence : 0.06131   
## Balanced Accuracy : 0.72249   
##   
## 'Positive' Class : violated   
##

The model is only 2% less accurate at 92% than the previous model, which was 94% accurate. The sensitivity of the model is 45% and the specificity is 99%. The no information is 88%, which is the same as the previous model. I would consider this a good quality model for predicting if a person in training data set will violate parole or not.

predNetBasicTest = predict(nnetBasic, test)  
head(predNetBasicTest)

## [1] didnotviolate didnotviolate didnotviolate didnotviolate didnotviolate  
## [6] didnotviolate  
## Levels: didnotviolate violated

confusionMatrix(predNetBasicTest, test$violator, positive = "violated")

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction didnotviolate violated  
## didnotviolate 176 14  
## violated 3 9  
##   
## Accuracy : 0.9158   
## 95% CI : (0.8687, 0.9502)  
## No Information Rate : 0.8861   
## P-Value [Acc > NIR] : 0.10836   
##   
## Kappa : 0.4732   
## Mcnemar's Test P-Value : 0.01529   
##   
## Sensitivity : 0.39130   
## Specificity : 0.98324   
## Pos Pred Value : 0.75000   
## Neg Pred Value : 0.92632   
## Prevalence : 0.11386   
## Detection Rate : 0.04455   
## Detection Prevalence : 0.05941   
## Balanced Accuracy : 0.68727   
##   
## 'Positive' Class : violated   
##

This model is pretty accurate in predicting parole violators with the test data. There is a 91% accuracy rate, 39% sensitivity rate, and 98% specificity rate. The no information rate is 88%. The percentages are very similar for the testing data as they were for the training data when using the nnetBasic model.

predNetTest = predict(nnetFit, test)  
head(predNetTest)

## [1] didnotviolate violated didnotviolate didnotviolate didnotviolate  
## [6] didnotviolate  
## Levels: didnotviolate violated

confusionMatrix(predNetTest, test$violator, positive = "violated")

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction didnotviolate violated  
## didnotviolate 176 17  
## violated 3 6  
##   
## Accuracy : 0.901   
## 95% CI : (0.8512, 0.9385)  
## No Information Rate : 0.8861   
## P-Value [Acc > NIR] : 0.29683   
##   
## Kappa : 0.3322   
## Mcnemar's Test P-Value : 0.00365   
##   
## Sensitivity : 0.26087   
## Specificity : 0.98324   
## Pos Pred Value : 0.66667   
## Neg Pred Value : 0.91192   
## Prevalence : 0.11386   
## Detection Rate : 0.02970   
## Detection Prevalence : 0.04455   
## Balanced Accuracy : 0.62205   
##   
## 'Positive' Class : violated   
##

This model is also fairly accurate in predicting parole violators when looking at the test data. There is a 90% accuracy rate, 26% sensitivity rate, 98% specificty rate, and a no informaiton rate of 88%. This model is almost as accurate when applied to the testing data set as it is for the training data set.

Because both of the models have such a high accuracy rating when applied to both the training and testing data set, I do not believe either model overfits the data. Given the statistics presented above, both models should be useful when applying them to new data.