Assignment 1 - Johann Visser - 21293681

2023-04-05

## Question 1

### Part A

False positives and false negatives. In other words, values are either ‘falsely’ identified to be present (false positives) or not present (false negatives).False positives and false negatives can have serious consequences when making real world decisions, resulting in incorrect classifications which could be used to perform real world decision, which could have significant safety, economic or environmental consequences. Unfortunately it is not possible to have a perfect model and models are tuned to optimise the benefit and risk associated with the decision. The context of the decision making within the business, including processes and systems within which the model will be operating, should determine which way the model is tuned.

### Part B

An examples scenario could be the diagnostics of cancer within human patients. A false negative in medicine could lead to a patient’s cancer not being detected, which could result in serious health consequences or even prove to be fatal. A false positive could on the other hand lead the patient to receiving treatment, like chemotherapy, unnecessarily, resulting in life changing side effects. Neither outcomes should be taken lightly and sufficient tuning is required to minimise the likelihood of either of these outcomes. ‘Human-in-the-loop’ should always be there to monitor, challenge and validate (as much as is possible) the recommendations of such systems and models.

### Part C

The confusion matrix will likely (but not necessarily) change between the training and testing data. In general, it is likely that there will be an increase in false positives and false negatives due to the model learning from a sample of data. The model is not able to see all possible combinations of predictor and response values, and is required to use what it has learned from training to predict most likely responses. Training data could also have outliers or noise that introduces residual error into the model. Depending on the design of the model, it could also either overfit, meaning it is not able to generalise as well, resulting in more false positives, or it could unfit, meaning it generalises too much, resulting in more false negatives.

## Question 2

### Part A

# Importing the 'Auto' data from ISLR2  
library(ISLR2)  
data(Auto)  
  
# Dividing the data into two groups of equal size  
auto\_1 <- Auto[c(1:(nrow(Auto)/2)),]  
auto\_2 <- Auto[c((nrow(Auto)/2)+1):(nrow(Auto)),]  
  
# Providing the value for accelerate for the first record of the second half of the data  
cat("The value for accelerate for the first record of the second half of the data is", auto\_2[1,"acceleration"])

## The value for accelerate for the first record of the second half of the data is 17.4

### Part B

# Build the model  
auto\_model <- lm(formula = acceleration ~ displacement + horsepower:weight + mpg:weight, data = Auto)  
summary(auto\_model)

##   
## Call:  
## lm(formula = acceleration ~ displacement + horsepower:weight +   
## mpg:weight, data = Auto)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -5.4573 -1.5096 -0.1186 1.1713 7.9039   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 1.576e+01 7.386e-01 21.338 < 2e-16 \*\*\*  
## displacement 1.250e-03 2.931e-03 0.426 0.67003   
## horsepower:weight -7.607e-06 1.400e-06 -5.432 9.88e-08 \*\*\*  
## weight:mpg 3.295e-05 1.031e-05 3.195 0.00151 \*\*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 2.203 on 388 degrees of freedom  
## Multiple R-squared: 0.3674, Adjusted R-squared: 0.3626   
## F-statistic: 75.13 on 3 and 388 DF, p-value: < 2.2e-16

From the above results, we can see that the p-values for both the pairwise predictors, horsepower:weight and mpg:weight, are less than 0.05, which means we would want to keep these in our model. Displacement produces a p-value of greater than 0.05, therefore we shall exclude it.

### Part C

# Build the model excluding displacement  
auto\_model <- lm(formula = acceleration ~ horsepower:weight + mpg:weight, data = Auto)  
summary(auto\_model)

##   
## Call:  
## lm(formula = acceleration ~ horsepower:weight + mpg:weight, data = Auto)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -5.4057 -1.5130 -0.1156 1.1496 7.7253   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 1.581e+01 7.298e-01 21.659 < 2e-16 \*\*\*  
## horsepower:weight -7.052e-06 5.178e-07 -13.620 < 2e-16 \*\*\*  
## weight:mpg 3.309e-05 1.030e-05 3.213 0.00142 \*\*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 2.2 on 389 degrees of freedom  
## Multiple R-squared: 0.3672, Adjusted R-squared: 0.3639   
## F-statistic: 112.8 on 2 and 389 DF, p-value: < 2.2e-16

The expression for the final model is: acceleration = 1.581e+01 - 7.052e-06 \* horsepower:weight + 3.309e-05 \* weight:mpg

## Question 3

### Part A

# Creating the new variable 'mpgclass' and store in a new table  
Auto\_3a <- Auto  
Auto\_3a$mpgclass <- ifelse(Auto$mpg < 20 , "low",  
 ifelse(Auto$mpg < 27, "medium", "high"))  
  
# Proportion of each category  
prop.table(table(Auto\_3a$mpgclass))

##   
## high low medium   
## 0.3341837 0.3852041 0.2806122

Proportions are presented above

### Part B

#Import MASS library  
library(MASS)

##   
## Attaching package: 'MASS'

## The following object is masked from 'package:ISLR2':  
##   
## Boston

# New table with only selected columns  
Auto\_3b <- Auto\_3a[, c("acceleration", "displacement", "horsepower", "weight", "mpgclass")]  
  
# New table with only predictors  
Auto\_3b\_predictors <- Auto\_3b[,-5]  
  
# Build the model excluding displacement  
Auto\_3b\_model <- lda(mpgclass ~ acceleration + displacement + horsepower + weight, data = Auto\_3b)  
  
# Perform predictions  
predictions <- predict(Auto\_3b\_model, newdata = Auto\_3b\_predictors)  
  
# Confusion matrix  
conf <- table(Auto\_3b$mpgclass, predictions$class)  
conf

##   
## high low medium  
## high 107 1 23  
## low 3 134 14  
## medium 44 16 50

The confusion matrix can be seen above

# Calculating classification error  
error <- 1 - sum(diag(conf)) / sum(conf)  
error <- cat(round(error\*100,1),"%")

## 25.8 %

The classification error can be found above

### Part C

# Creating a test dataset using only cars from the year '75' and only predictors  
Auto\_3c\_test <- subset(Auto\_3a, year == 75)  
Auto\_3c\_test <- Auto\_3c\_test[, c("acceleration", "displacement", "horsepower", "weight", "mpgclass")]  
  
#Predictions table  
Auto\_3c\_test\_predictions <- Auto\_3c\_test[, c("acceleration", "displacement", "horsepower", "weight")]  
  
# Perform predictions  
predictions\_3c\_test <- predict(Auto\_3b\_model, newdata = Auto\_3c\_test\_predictions)  
  
# Confusion matrix  
conf <- table(Auto\_3c\_test$mpgclass, predictions\_3c\_test$class)  
conf

##   
## high low medium  
## high 3 0 0  
## low 0 14 1  
## medium 1 1 10

The confusion matrix can be seen above

# Calculating classification error  
error <- 1 - sum(diag(conf)) / sum(conf)  
error <- cat(round(error\*100,1),"%")

## 10 %

The classification error can be found above

### Part D

# Taking the test data from (3c)  
Auto\_3d\_test <- Auto\_3c\_test  
  
#Training data using cars from all other years  
Auto\_3d\_train <- subset(Auto\_3a, year != 75)  
  
# New table with only selected columns  
Auto\_3d\_train <- Auto\_3d\_train[, c("acceleration", "displacement", "horsepower", "weight", "mpgclass")]  
  
# New table with only predictors  
Auto\_3d\_train\_predictors <- Auto\_3d\_train[,-5]  
  
# Build the model excluding displacement  
Auto\_3d\_model <- lda(mpgclass ~ acceleration + displacement + horsepower + weight, data = Auto\_3d\_train)  
  
# Perform predictions  
predictions <- predict(Auto\_3d\_model, newdata = Auto\_3d\_train\_predictors)  
  
# Confusion matrix  
conf <- table(Auto\_3d\_train$mpgclass, predictions$class)  
conf

##   
## high low medium  
## high 107 1 20  
## low 3 119 14  
## medium 48 14 36

The confusion matrix can be seen above

# Calculating classification error  
error <- 1 - sum(diag(conf)) / sum(conf)  
error <- cat(round(error\*100,1),"%")

## 27.6 %

The training error can be seen above

### Part E

# New table with only predictors from test data  
Auto\_3d\_test\_predictors <- Auto\_3d\_test[,-5]  
  
# Build the model excluding displacement  
Auto\_3e\_model <- lda(mpgclass ~ acceleration + displacement + horsepower + weight, data = Auto\_3d\_train)  
  
# Perform predictions  
predictions <- predict(Auto\_3e\_model, newdata = Auto\_3d\_test\_predictors)  
  
# Confusion matrix  
conf <- table(Auto\_3d\_test$mpgclass, predictions$class)  
conf

##   
## high low medium  
## high 3 0 0  
## low 0 14 1  
## medium 1 1 10

The confusion matrix can be seen above

# Calculating classification error  
error <- 1 - sum(diag(conf)) / sum(conf)  
error <- cat(round(error\*100,1),"%")

## 10 %

The training error can be seen above

### Part F

The classification error is higher for 3b and 3d than it is for 3c and 3e. In both the cases of 3c and 3e, the model is tested using a subset of the data, only taking cars from year 75. This could indicate that either the model is better at classifying ‘mpg’ for cars from this year, or because the sample size of the testing data is much smaller (29 compared to 392), resulting in unrepresentative test data sample to confidently evaluate the model. Interestingly, 3b performs slightly better than 3d, where 3d excludes cars from year 75; this could perhaps validate our hypothesis from earlier, that the model better predicts ‘mpg’ for cars from year 75, but the performance difference is too small to confidently conclude this. Comparing the confusion matrices of 3b and 3d, we note that the exclusion of cars from year 75 reduces the model’s ability to accurately classify low and medium ‘mpg’ values.

In our case, we see no variance in the test errors of 3c and 3e, but we may have expected 3e to perform worse. This is because the model in 3e was trained using cars not from year 75 and tested to predict ‘mpg’ for cars from the year 75. There is no representation of cars from the year 75 in the model, making it potentially more difficult to classify cars from this year, because the model has not seen some patterns that may be relevant to classification.