Assignment 1

Andily Theoridho (22764884), Choong Nicholas (21980614),

STAT4064

Split the work by the number of questions in order where 22764884 completed first 6 questions, 21980614 completed the final 6 questions and the final work is checked by both students. All statistical computation was completed in RStudio using ISLR2 Auto data.

Question 1 (a) :

In a binary classification problem, a confusion matrix summarizes the performance of a classification model by presenting the counts of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) predictions. An incorrect classification can be made in two ways: either the model predicts the positive class when the true class is negative (FP), or the model predicts the negative class when the true class is positive (FN).

It is important to distinguish between these two types of errors because they have different implications in different applications. In some cases, the cost of false positives (FP) may be higher than the cost of false negatives (FN), and vice versa. For instance, in medical diagnosis, a false negative result may be more costly than a false positive, since a missed diagnosis can be life-threatening. In contrast, in spam email filtering, a false positive may be more costly, as it may result in important emails being mistakenly classified as spam and lost.

Therefore, depending on the context of the application, a classification model can be optimized to minimize one type of error over the other, or to balance both types of errors by using a suitable metric, such as accuracy, precision, recall, F1-score, or area under the curve (AUC) of the receiver operating characteristic (ROC) curve.

Question 1 (b):

Consider a scenario where a bank is using a machine learning model to approve or reject loan applications. In this case, false positives and false negatives have different consequences. A false positive occurs when the model approves a loan application that should have been rejected, while a false negative occurs when the model rejects a loan application that should have been approved.

If the model makes a false positive error and approves a risky loan, the bank may incur financial losses if the borrower defaults on the loan. On the other hand, if the model makes a false negative error and rejects a low-risk loan, the bank may miss out on potential revenue and lose the opportunity to build a relationship with a new customer.

Therefore, it is crucial for the bank to consider both types of errors and optimize the model to minimize the overall cost of misclassification, considering the different costs of false positives and false negatives in their business context. For instance, the bank may prioritize minimizing false positives if the cost of a default is much higher than the lost revenue from missed opportunities. Conversely, the bank may prioritize minimizing false negatives if the cost of losing a potential customer is much higher than the cost of occasional defaults.

Question 1 (c):

One possible change we could observe is a decrease in the number of true positives (TP) and true negatives (TN) and an increase in the number of false positives (FP) and false negatives (FN) in the testing data, especially if the model has overfitted the training data. In this case, the model may have learned to memorize the training data instead of learning general patterns, and therefore, may not generalize well to new, unseen data.

Another possible change we could observe is a slight decrease or increase in the number of TP, TN, FP, and FN in the testing data, depending on the complexity of the model, the quality of the data, and the degree of variability between the training and testing data.

In general, we would expect the changes to be relatively small if the model is well-trained, and the testing data is representative of the real-world data the model will encounter. Additionally, we would expect the changes to be in the same direction as the changes in the performance metrics (such as accuracy, precision, recall, F1-score, or AUC) between the training and testing data. If the performance metrics degrade on the testing data, we would expect to see a decrease in TP and TN and an increase in FP and FN in the confusion matrix. Conversely, if the performance metrics improve on the testing data, we would expect to see an increase in TP and TN and a decrease in FP and FN in the confusion matrix.

Question 2 (a):

To divide the Auto data into two equal-sized groups, we can simply split the data frame into two halves using the split() function in R.

```
# Split the Auto data into two halves

auto_split <- split(Auto, ifelse(row.names(Auto) <= nrow(Auto)/2,
"first", "second"))


# List the value of acceleration of the first record of the second
half

auto_split[["second"]][1, "acceleration"]


> auto_split[["Second"]][1, "acceleration"]

[1] 11.5
```

Therefore, the value of acceleration for the first record of the second half of the Auto data is 11.5.

Question 2 (b):

To conduct a linear regression with pairwise interactions between the predictor variables, we can use the following formula:

$$model < -\, lm(acceleration \sim displacement\, +\, horsepower\, +\, weight\, +\, mpg \\ +\, horsepower{:}weight\, +\, mpg{:}weight, data\, =\, Auto)$$

To determine which predictor variables to keep in the model, we can perform hypothesis tests on the coefficients of each predictor and the interaction terms. We can use the t-test and the associated p-value to make the decision. If the p-value is less than 0.05, we reject the null hypothesis and conclude that the predictor is significant.

Here are the results of the hypothesis tests for the coefficients:

```
Coefficients:
                    Estimate Std. Error t value Pr(>|t|)
(Intercept)        3.191e+01  3.142e+00  10.155  < 2e-16 ***
displacement      -1.002e-02  2.589e-03  -3.869 0.000128 ***
horsepower        -1.736e-01  1.755e-02  -9.890  < 2e-16 ***
weight            -1.451e-03  1.034e-03  -1.403 0.161404
mpg               -2.830e-01  7.585e-02  -3.730 0.000220 ***
horsepower:weight  2.496e-05  4.724e-06   5.283 2.13e-07 ***
weight:mpg         9.649e-05  2.868e-05   3.365 0.000842 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

From the above results, we see that the coefficients for weight are not significant at the 5% level since their p-values are greater than 0.05. Therefore, we would remove the weight and the corresponding interaction which includes the weight predictor variables from the model. The remaining predictor variables displacement, horsepower and mpg are all significant at the 5% level, so we would keep them in the model.

Question 2 (c):

$$acceleration \sim displacement + horsepower + mpg$$

Which includes the predictor variables displacement, horsepower, and mpg. These variables were found to be significant at the 5% level in the linear regression analysis with pairwise interactions conducted on the Auto dataset.

Here is the value for each coefficient:

```
Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  24.165841   0.861810  28.041  < 2e-16 ***
displacement  0.006978   0.002278   3.064 0.002339 **
horsepower   -0.078445   0.005851 -13.408  < 2e-16 ***
mpg          -0.076179   0.021478  -3.547 0.000438 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
Residual standard error: 1.662 on 385 degrees of freedom
Multiple R-squared:  0.6428,     Adjusted R-squared:  0.6373
F-statistic: 115.5 on 6 and 385 DF,  p-value: < 2.2e-16
```

The coefficient for horsepower and mpg both has a negative sign, which suggests that as the horsepower and displacement of the vehicle increases, its acceleration decreases. The coefficient for displacement has a positive sign, which suggests that as the displacement of the vehicle increases, its acceleration also increases.

Question 3 (a):

We used conditional statements to assign the value of “mpgclass” based on the value of the “mpg”. Then, we factorise the variable based on the categories in order and calculate the proportion of each category. According to the result, 39% of the observations fall into the ‘low’ category, 28% fall into the ‘medium’ category, and 33% fall into the ‘high’ category.

Question 3 (b):

To conduct a linear discriminant analysis between the predictor variables, we can use the following formula:

$$model <- lm(\text{mpgclass} \sim \text{acceleration} + displacement + horsepower + weight, data = Auto)$$

To determine the classification error and show a confusion matrix, we first need to predict the probabilities on the whole data set.

Below is the confusion matrix.

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

The classification error is 0.258.

Question 3 (c):

A subset of data from the year 75 was extracted from the dataset as test data.

To determine the classification error and show a confusion matrix, we first need to predict the probabilities on the test data.

Below is the confusion matrix.

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

The classification error is 0.1.

Question 3 (d)

To create train and test data, we split the dataset by year. We used all the data except for the year 75 as train data, and we used only the data from the year 75 as test data.

$$model\ <-lm(\text{mpgclass} \sim \text{acceleration} + displacement\ +\ horsepower\ +\ weight,$$
$$\text{data}\ =\ \text{train})$$

To determine the classification error and show a confusion matrix, we first need to predict the probabilities on the train data.

Below is the confusion matrix.

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

The classification error is 0.276.

Question 3 (e):

To create train and test data, we split the dataset by year. We used all the data except for the year 75 as train data, and we used only the data from the year 75 as test data.

$$model\ < -\ lm(\text{mpgclass} \sim \text{acceleration} + displacement\ +\ horsepower\ +\ weight,$$
$$\text{data}\ =\ \text{train})$$

To determine the classification error and show a confusion matrix, we first need to predict the probabilities on the test data.

Below is the confusion matrix.

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

The classification error is 0.1.

Question 3 (f):

The LDA model trained on the entire dataset has a classification error of 0.258 whereas the model trained on the dataset excluding the year 75 has a classification error of 0.276. The confusion matrices of the two models reveal that they both correctly classified the same amount of high-category data, but the model trained on the partial dataset performed worse on the low- and medium-category data than the model trained on the whole data.

When a subset of the year 75 dataset was used as the test data, both models had the same performance, confusion matrices and classification errors. The confusion matrices show that both models misclassified 3 data points.

When constructing a classification rule using a smaller subset of the data, there is a higher chance that the rule will not generalize well to new data, resulting in a higher test error. Therefore, we would expect the test error from the model trained on the whole dataset to be smaller than the test error from the model trained on the partial dataset. However, the test results from both models are the exact same which could mean that the classification rule constructed using a smaller subset of the data generalise better to new data than the rule constructed using the entire dataset. Another possibility is that the test errors are similar due to chance or due to the specific characteristics of the data.