MATH 4322 Homework 3

Phu Nguyen

Fall 2021

Problem 1

Suppose we collect data for a group of students in a statistics class with variables X_1 = hours studied, X_2 =undergrad GPA, and Y = receive an A. We fit a logistic regression and produce estimated coefficient, $\hat{\beta}_0 = -6$, $\hat{\beta}_1 = 0.05$, $\hat{\beta}_2 = 1$. * $\hat{Y} = -6 + 0.05X_1 + 1X_2$

(a) Estimate the probability that a student who studies for 40 h and has an undergrad GPA of 3.5 gets an A in the class.

```
(Px = (exp(-6 + (0.05*40) + (1*3.5)))/(1 + exp(-6 + (0.05*40) + (1*3.5))))
```

```
## [1] 0.3775407
```

- This means the predicted probability that a student get an A in the class given the hours of studies is 40 and GPA of 3.5 is 37.75%.

(b) How many hours would the student in part (a) need to study to have a 50% chance of getting an A in the class?

- $log(p(X)/1 - p(X) = B_0 + B_1X + B_2X$

```
chance = (log(0.50/(1-0.50)) + 6 - (1*3.5))/0.05
chance
```

```
## [1] 50
```

- By the *logistic function* this show that a student need to study 50 hours to have a 50% chance of getting an A in the class. However, when I tried plugging in 50 hours in the function:

```
(Px1 = (exp(-6 + (0.05*50) + (1*3.5)))/(1 + exp(-6 + (0.05*40) + (1*3.5))))
```

```
## [1] 0.6224593
```

it give me a 62.24% of getting an A in the class given 50 hours and GPA of 3.5. So I tried to play with the number of hours and the closet number that would give 50% chance of getting an A is 45.619

```r
(Px2 = (exp(-6 + (0.05*45.619) + (1*3.5)))/(1 + exp(-6 + (0.05*40) + (1*3.5))))
```

```
## [1] 0.5000101
```

Problem 2

In this problem, you will develop a model to predict whether a given car gets high or low gas mileage based on the `Auto` data set in the `ISLR` package.

```r
library(ISLR)
head(Auto)
```

```
##   mpg cylinders displacement horsepower weight acceleration year origin
## 1  18         8          307        130   3504         12.0   70      1
## 2  15         8          350        165   3693         11.5   70      1
## 3  18         8          318        150   3436         11.0   70      1
## 4  16         8          304        150   3433         12.0   70      1
## 5  17         8          302        140   3449         10.5   70      1
## 6  15         8          429        198   4341         10.0   70      1
##                        name
## 1 chevrolet chevelle malibu
## 2         buick skylark 320
## 3        plymouth satellite
## 4             amc rebel sst
## 5               ford torino
## 6          ford galaxie 500
```

(a) Create a binary variable, `mpg01`, that contains a 1 if mpg contains a value above its median, and a 0 if mpg contains a value below its median. You can compute the median using the `median()` function. Note you may find it helpful to use the `data.frame()` function to create a single data set containing both `mpg01` and the other Auto variables.

```r
mpg01 = rep(0, length(Auto$mpg))
mpg01[Auto$mpg > median(Auto$mpg)] = 1
Auto = data.frame(Auto, mpg01)
summary(Auto)
```

```
##       mpg          cylinders      displacement     horsepower        weight    
##  Min.   : 9.00   Min.   :3.000   Min.   : 68.0   Min.   : 46.0   Min.   :1613  
##  1st Qu.:17.00   1st Qu.:4.000   1st Qu.:105.0   1st Qu.: 75.0   1st Qu.:2225  
##  Median :22.75   Median :4.000   Median :151.0   Median : 93.5   Median :2804  
##  Mean   :23.45   Mean   :5.472   Mean   :194.4   Mean   :104.5   Mean   :2978  
##  3rd Qu.:29.00   3rd Qu.:8.000   3rd Qu.:275.8   3rd Qu.:126.0   3rd Qu.:3615  
##  Max.   :46.60   Max.   :8.000   Max.   :455.0   Max.   :230.0   Max.   :5140  
##                                                                               
##   acceleration        year           origin                      name    
##  Min.   : 8.00   Min.   :70.00   Min.   :1.000   amc matador       :  5  
##  1st Qu.:13.78   1st Qu.:73.00   1st Qu.:1.000   ford pinto        :  5  
##  Median :15.50   Median :76.00   Median :1.000   toyota corolla    :  5  
##  Mean   :15.54   Mean   :75.98   Mean   :1.577   amc gremlin       :  4  
##  3rd Qu.:17.02   3rd Qu.:79.00   3rd Qu.:2.000   amc hornet        :  4  
```

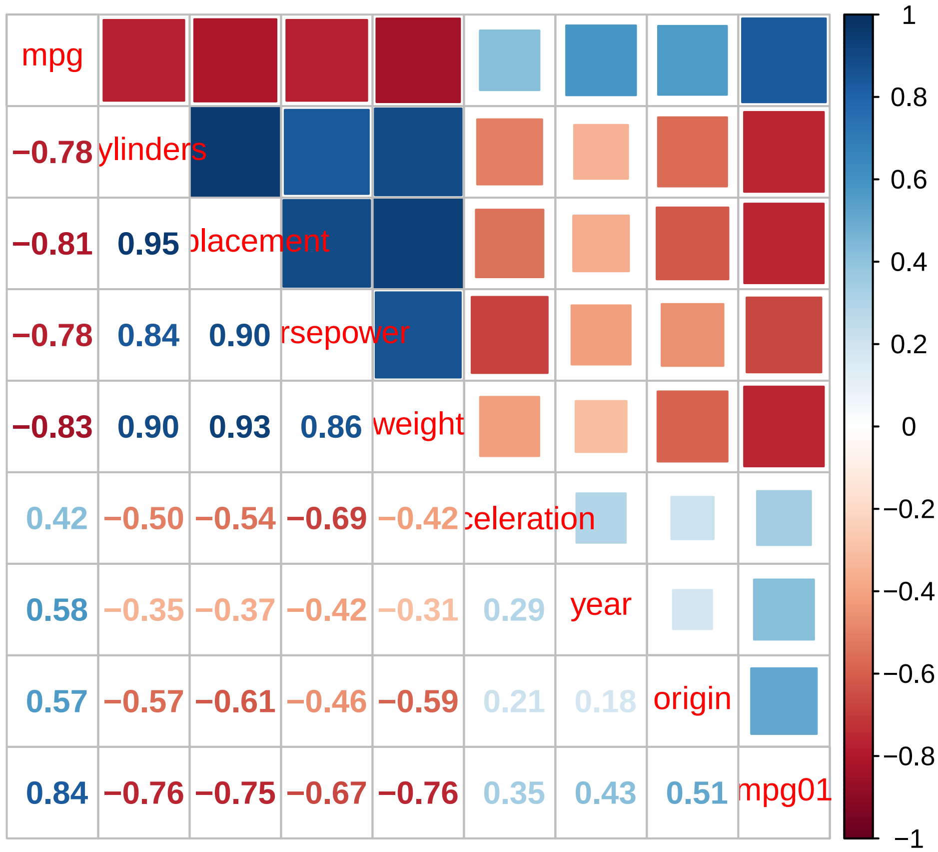
```
## Max.   :24.80   Max.   :82.00   Max.   :3.000   chevrolet chevette:  4
##                                                  (Other)           :365
##     mpg01
## Min.   :0.0
## 1st Qu.:0.0
## Median :0.5
## Mean   :0.5
## 3rd Qu.:1.0
## Max.   :1.0
##
```

(b) Explore the data graphically in order to investigate the association between `mpg01` and the other features. Which of the other features seem most likely to be useful in predicting `mpg01`? Scatterplots and boxplots may be useful tools to answer this question. Describe your findings.

```
#install.packages('corrplot')
correlation = cor(Auto[,-9])
library(corrplot)
```

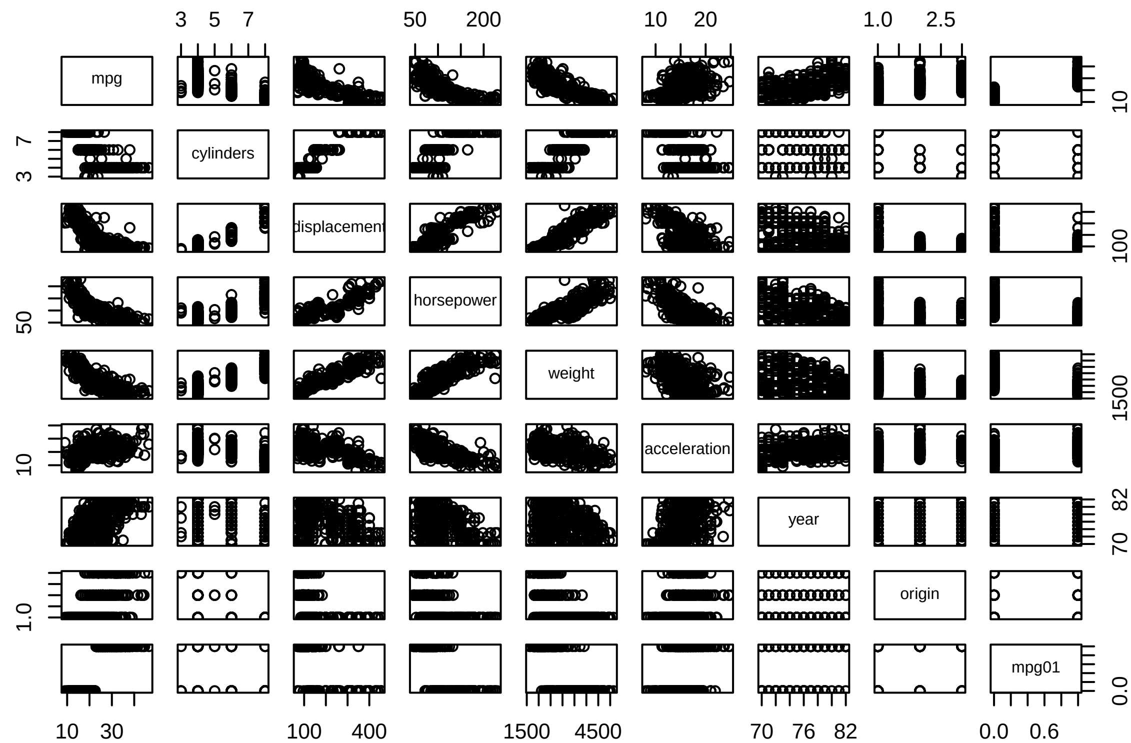
```
## corrplot 0.90 loaded
```

```
corrplot::corrplot.mixed(correlation, upper = "square")
```



* This correlation plot show that the variables that is useful for predicting *mpg* is *cylinders*, *displacement*, *horsepower*, and *weight*.

```
pairs(Auto [, -9])
```



* The scatterplot also show the same result

(c) Split the data into a training set and a test set.

```
set.seed(101)
#Selecting 75% of data
sample = sample.int(n = nrow(Auto), size = round(0.75 * nrow(Auto)), replace = FALSE)
training = Auto[sample,]
test = Auto[-sample,]  #store the left out rows
```

(d) Perform logistic regression on the training data in order to predict `mpg01` using the variables that seemed most associated with `mpg01` in (b). What is the test error of the model obtained? That is use the test data to predict and get the confusion matrix and determine the error rate.

```
#creating a model
cylinders = as.factor(Auto$cylinders)
auto.glm = glm(mpg01 ~ cylinders + weight + displacement + horsepower, data = training, family = "binomi
summary(auto.glm)
```

```
##
## Call:
## glm(formula = mpg01 ~ cylinders + weight + displacement + horsepower,
```

```
##     family = "binomial", data = training)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2775  -0.1159   0.1114   0.3735   3.3090
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)  11.3278851  1.9033741   5.951 2.66e-09 ***
## cylinders     0.0836262  0.4055608   0.206 0.836635
## weight       -0.0014795  0.0007757  -1.907 0.056476 .
## displacement -0.0137454  0.0096172  -1.429 0.152934
## horsepower   -0.0550929  0.0161035  -3.421 0.000623 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 407.52  on 293  degrees of freedom
## Residual deviance: 155.52  on 289  degrees of freedom
## AIC: 165.52
##
## Number of Fisher Scoring iterations: 7
```

```
#using the test and training set
glm.pred = predict.glm(auto.glm, newdata = test, type = "response")
summary(glm.pred)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## 0.000005 0.042007 0.428677 0.468618 0.873397 0.993818
```

```
yHat = glm.pred > 0.5
table(test$mpg01, yHat)
```

```
##    yHat
##     FALSE TRUE
##   0    46    5
##   1     5   42
```

```
#the accuracy rate:
(accuracy = (46 + 42)/98)
```

```
## [1] 0.8979592
```

```
#the error rate:
(error = (5 + 5)/98)
```

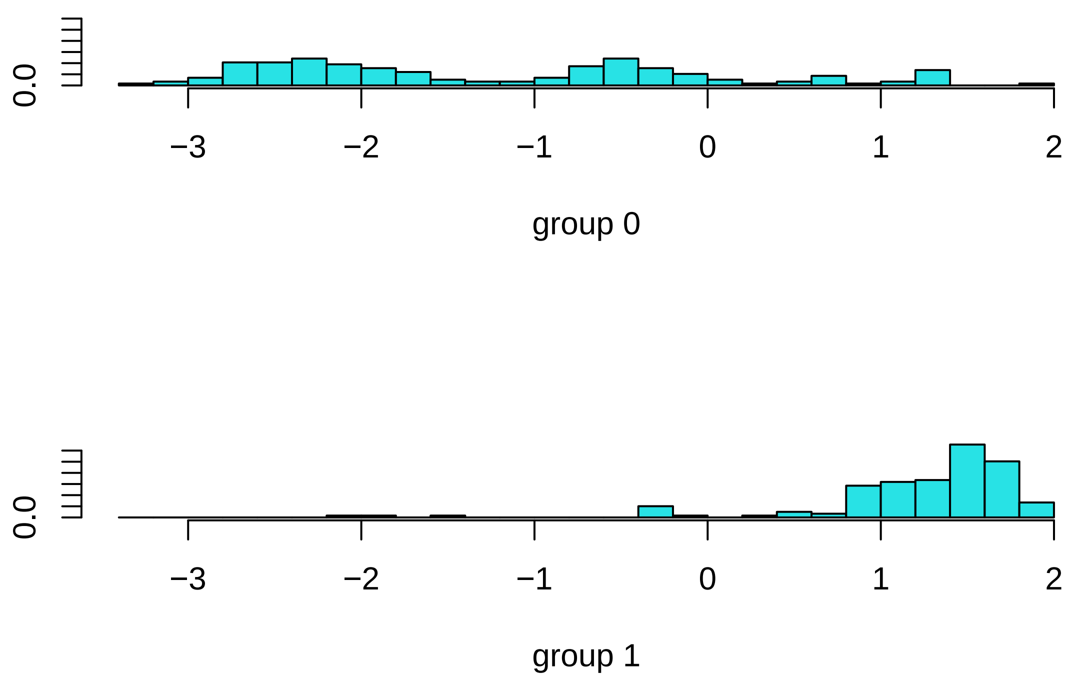
```
## [1] 0.1020408
```

(e) Perform LDA on the training data in order to predict `mpg01` using the variables that seemed most associated with mpg01 in (b). What is the test error of the model obtained? That is use the test data to predict and get the confusion matrix and determine the error rate.

```r
library(MASS)
mpg01.lda = lda(mpg01 ~ cylinders + weight + displacement + horsepower, data = training)
mpg01.lda
```

```
## Call:
## lda(mpg01 ~ cylinders + weight + displacement + horsepower, data = training)
##
## Prior probabilities of groups:
##         0         1
## 0.4931973 0.5068027
##
## Group means:
##   cylinders   weight displacement horsepower
## 0  6.779310 3608.386     274.4207  132.53793
## 1  4.167785 2323.758     114.3456   77.81879
##
## Coefficients of linear discriminants:
##                        LD1
## cylinders    -0.4665697390
## weight       -0.0008925017
## displacement  0.0002803062
## horsepower   -0.0023373068
```

```r
plot(mpg01.lda)
```



```r
lda.pred = predict(mpg01.lda, test)
names(lda.pred)
```

```
## [1] "class"     "posterior" "x"
```

```r
table(test$mpg01, lda.pred$class)
```

```
##
##      0  1
##   0 46  5
##   1  4 43
```

```r
#the accuracy rate:
(accuracy_rate = (46 + 43)/98)
```

```
## [1] 0.9081633
```

```r
#the error rate:
(error_rate = (5 + 4) / 98)
```

```
## [1] 0.09183673
```