EAS 508 Assignment 2– Kaushal Shivaprakashan-50608818

Remark:

- `Please change the file name when submitting your assignment`
- `Only .html file is allowed to submit, any other form will not be graded`
- `Any conclusions without evidence will be granted 0`
- `Academic integrity`

Question 1 [5 points]

Describe a situation or problem from your job, everyday life, current events, etc., for which a logistic regression model would be appropriate. List some (up to 5) predictors that you might use. Please show your work.- Answer:

The logistic regression model is fitted to problems where the outcome is binary, yes/no, or success/failure. The following everyday example describes an appropriate situation for logistic regression:

Scenario: Predict whether a student would pass or fail in a course.

The course outcome prediction as pass/fail in a college environment helps via early intervention and academic support. This target variable is binary: pass 1 or fail 0. A logistic regression model will help to show the relation of certain predictors on the likelihood of passing the course.

Predictors

1. Study Hours per Week: Generally speaking, the time a student puts into studying every week is positively related to his or her academic achievement. Such a measure could provide an indication of the seriousness of the student regarding the course.
2. Attendance Rate: The attendance rate for the given student is another critical factor; class attendance often leads to better understanding and participation in class activities.
3. Previous Scores on Exams: The average scores of previous exams or grades in related subjects can be a good predictor because it reflects the general academic aptitude of the student and his level of preparation.
4. Activity Level in Class: A record of the level of class participation by the student, through discussions, group activities, or assignments, may give a pointer to the student's mastery of the material and the confidence level in the subject.
5. Availability of Academic Resources: This may relate to other resources, such as tutoring, study groups, or online course materials, depending on what is available since better support normally goes hand in hand with more conducive learning.

Question 2 [20 points]

In this problem, we will use the Naive Bayes algorithm to fit a spam filter by hand. This question does not involve any programming but only derivation and hand calculation. Spam filters are used in all email services to classify received emails as "Spam" or "Not Spam". A simple approach involves maintaining a vocabulary of words that commonly occur in “Spam” emails and classifying an email as “Spam” if the number of words fromthe dictionary that are present in the email is over a certain threshold. We are given the vocabulary consists of 15 words

$$V = \text{secret, offer, low, price, valued, customer, today, dollar, million, sports, is, for, play, healthy}.$$

We will use V_i to represent the ith word in V . As our training dataset, we are also given 3 example spam messages,

- million dollar offer for today
- secret offer today
- secret is secret

and 4 example non-spam messages

- low price for valued customer
- play secret sports today
- sports is healthy
- low price pizza

Recall that the Naive Bayes classifier assumes the probability of an input depends on its input feature. The feature for each sample is defined as $x^{(i)} = [x_1^{(i)}, x_2^{(i)}, \cdots, x_p^{(i)}], i = 1, \cdots, m$ and the class of the ith sample is $y^{(i)}$. In our case the length of the input vector is $p = 15$, which is equal to the number of words in the vocabulary V (hint: recall that how did we define a dummy variable). Each entry $x_j^{(i)}$ is equal to the number of times word V_j occurs in the i-th message.

1.[5 points] Calculate class prior $P(y = 0)$ and $P(y = 1)$ from the training data, where $y = 0$ corresponds to spam messages, and $y = 1$ corresponds to non-spam messages. Note that these class prior essentially corresponds to the frequency of each class in the training sample. Write down the predictor vectors for each spam and non-spam messages.

```
Answer:
```

The prior probabilities, P(y=0) and P(y=1), are based on the proportion of spam and non-spam messages in the training dataset.

Count Spam Messages (y = 0): There are 3 spam messages. Count Non-Spam Messages (y = 1): There are 4 non-spam messages. Total Messages: There are 3 + 4 = 7 messages in total.

So, the priors are: P(y=0) = 3 / 7 P(y=1) = 4 / 7

Predictor Vectors Each email can be represented as a vector x^(i) of length 15, corresponding to the frequency of each word in the vocabulary V in that email. Here’s the feature vector for each message:

Spam messages (y = 0):

“million dollar offer for today”: x^(1) = [0, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 1, 0, 0, 0] “secret offer today”: x^(2) = [1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0] “secret is secret”: x^(3) = [2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0]

Non-spam messages (y = 1):

“low price for valued customer”: x^(4) = [0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0] “play secret sports today”: x^(5) = [1, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0] “sports is healthy”: x^(6) = [0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 0] “low price pizza”: x^(7) = [0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1]

```
spam_messages <- c("million dollar offer for today",
                   "secret offer today",
                   "secret is secret")

non_spam_messages <- c("low price for valued customer",
                       "play secret sports today",
                       "sports is healthy",
                       "low price pizza")


total_messages <- length(spam_messages) + length(non_spam_messages)


P_y0 <- length(spam_messages) / total_messages
P_y1 <- length(non_spam_messages) / total_messages


cat("P(y = 0) =", P_y0, "\n")
```

```
## P(y = 0) = 0.4285714
```

```
cat("P(y = 1) =", P_y1, "\n")
```

```
## P(y = 1) = 0.5714286
```

2. [15 points] In the Naive Bayes model, assuming the keywords are independent of each other (this is a simplification), the likelihood of a sentence with its feature vector x given a class c is given by

$$P(x|y = c) = \prod_{i=1}^{15} P(x_i|y = c), c = \{0, 1\}.$$

Given a test message "today is secret", using the Naive Bayes classier to calculate the posterior and decide whether it is spam or not spam. Please show your work- The message "today is secret" is classified as spamm because the posterior probability for spam is higher than the posterior probability for non-spam. .

```
Answer:
```

Given the test message "today is secret," we need to calculate the posterior probabilities for each class using Naive Bayes.

Spam Messages:

"million dollar offer for today" : Vectors = [0,1,0,0,0,0,1,1,1,0,0,1,0,0,0] "secret offer today" : Vectors = [1,1,0,0,0,0,1,0,0,0,0,0,0,0,0] "secret is secret" : Vectors = [2,0,0,0,0,0,0,0,0,0,1,0,0,0,0] Total occurrences in Spam messages: Vectors = [3,2,0,0,0,0,2,1,1,0,1,1,0,0,0]

∑(spam word occurrences) = 11 Non-spam Message Analysis:

Non-spam Messages:

"low price for valued customer" : Vectors = [0,0,1,1,1,1,0,0,0,0,0,1,0,0,0] "play secret sports today" : Vectors = [1,0,0,0,0,0,1,0,0,1,0,0,1,0,0] "sports is healthy" : Vectors = [0,0,0,0,0,0,0,0,0,1,1,0,0,1,0] "low price pizza" : Vectors = [0,0,1,1,0,0,0,0,0,0,0,0,0,0,1] Total occurrences in Non-spam messages: Vectors = [1,0,2,2,1,1,1,0,0,2,1,1,1,1,1]

∑(non-spam word occurrences) = 15 Laplace Smoothing Probability Computation:

Spam Class (y = 0):

P(secret|y = 0) = (3+1)/(11+15) = 0.15 P(offer|y = 0) = (2+1)/(11+15) = 0.11 P(low|y = 0) = (0+1)/(11+15) = 0.03 P(price|y = 0) = (0+1)/(11+15) = 0.03 P(valued|y = 0) = (0+1)/(11+15) = 0.03 P(customer|y = 0) = (0+1)/(11+15) = 0.03 P(today|y = 0) = (2+1)/(11+15) = 0.11 P(dollar|y = 0) = (1+1)/(11+15) = 0.07 P(million|y = 0) = (1+1)/(11+15) = 0.07 P(sports|y = 0) = (0+1)/(11+15) = 0.03 P(is|y = 0) = (1+1)/(11+15) = 0.07 P(for|y = 0) = (1+1)/(11+15) = 0.07 P(play|y = 0) = (0+1)/(11+15) = 0.03 P(healthy|y = 0) = (0+1)/(11+15) = 0.03 P(pizza|y = 0) = (0+1)/(11+15) = 0.03 Probability of Spam Message:

P(y = 0) = 3/7 = 0.42

Non-Spam Class (y = 1):

P(secret|y = 1) = (1+1)/(15+15) = 0.06 P(offer|y = 1) = (0+1)/(15+15) = 0.03 P(low|y = 1) = (2+1)/(15+15) = 0.10 P(price|y = 1) = (2+1)/(15+15) = 0.10 P(valued|y = 1) = (1+1)/(15+15) = 0.06 P(customer|y = 1) = (1+1)/(15+15) = 0.06 P(today|y = 1) = (1+1)/(15+15) = 0.06 P(dollar|y = 1) = (0+1)/(15+15) = 0.03 P(million|y = 1) = (0+1)/(15+15) = 0.03 P(sports|y = 1) = (2+1)/(15+15) = 0.10 P(is|y = 1) = (1+1)/(15+15) = 0.06 P(for|y = 1) = (1+1)/(15+15) = 0.06 P(play|y = 1) = (1+1)/(15+15) = 0.06 P(healthy|y = 1) = (1+1)/(15+15) = 0.06 P(pizza|y = 1) = (1+1)/(15+15) = 0.06 Probability of Non-Spam Message:

P(y = 1) = 4/7 = 0.57

Probabilities of Words Given the Class (Spam vs Non-Spam):

Spam Example (y = 0):

P(today, is, secret|y = 0) = P(today|y = 0) x P(is|y = 0) x P(secret|y = 0) = (0.11) x (0.11) x (0.15) = 0.00202 P(y = 0|today is secret) = P(y = 0) x P(today, is, secret|y = 0) = (0.42) x (0.00202) = 0.00086 Non-Spam Example (y = 1):

P(today, is, secret|y = 1) = P(today|y = 1) x P(is|y = 1) x P(secret|y = 1) = (0.06) x (0.06) x (0.06) = 0.00028 P(y = 1|today is secret) = P(y = 1) x P(today, is, secret|y = 1) = (0.57) x (0.000287) = 0.00016 Conclusion:

Thereforee the P(y = 0|today is secret) > P(y = 1|today is secret), the message “Today is secret” is classified as SPAM.

```
vocab <- c("secret", "offer", "low", "price", "valued", "customer",
           "today", "dollar", "million", "sports", "is", "for",
           "play", "healthy", "pizza")


spam_messages <- c("million dollar offer for today",
                   "secret offer today",
                   "secret is secret")

non_spam_messages <- c("low price for valued customer",
                       "play secret sports today",
                       "sports is healthy",
                       "low price pizza")


create_feature_vector <- function(message, vocab) {
  words <- unlist(strsplit(tolower(message), "\\W+"))
  feature_vector <- table(factor(words, levels = vocab))
  return(as.numeric(feature_vector))
}
```

```
spam_vectors <- sapply(spam_messages, create_feature_vector, vocab)
non_spam_vectors <- sapply(non_spam_messages, create_feature_vector, vocab)


num_spam <- length(spam_messages)
num_non_spam <- length(non_spam_messages)
total_messages <- num_spam + num_non_spam

P_y0 <- num_spam / total_messages
P_y1 <- num_non_spam / total_messages


N_spam <- sum(spam_vectors)
N_non_spam <- sum(non_spam_vectors)


V <- length(vocab)


calculate_likelihood <- function(word_count, total_words) {
  return((word_count + 1) / (total_words + V))
}


test_message <- "today is secret"
test_vector <- create_feature_vector(test_message, vocab)


P_x_given_y0 <- prod(sapply(1:length(vocab), function(i) {
  calculate_likelihood(sum(spam_vectors[i, ]) * (spam_vectors[i, ] > 0), N_
spam)
}))


P_x_given_y1 <- prod(sapply(1:length(vocab), function(i) {
  calculate_likelihood(sum(non_spam_vectors[i, ]) * (non_spam_vectors[i, ]
> 0), N_non_spam)
}))


P_y0_given_x <- P_x_given_y0 * P_y0
P_y1_given_x <- P_x_given_y1 * P_y1


P_total <- P_y0_given_x + P_y1_given_x

P_y0_given_x_normalized <- P_y0_given_x / P_total
P_y1_given_x_normalized <- P_y1_given_x / P_total


cat("Posterior Probability of Spam (y=0):", P_y0_given_x_normalized, "\n")
```

```
## Posterior Probability of Spam (y=0): 1
```

```
cat("Posterior Probability of Not Spam (y=1):", P_y1_given_x_normalized, "\
n")
```

```
## Posterior Probability of Not Spam (y=1): 2.671386e-24
```

```
if (P_y0_given_x_normalized > P_y1_given_x_normalized) {
  cat("The message 'today is secret' is classified as: Spam\n")
} else {
  cat("The message 'today is secret' is classified as: Not Spam\n")
}
```

```
## The message 'today is secret' is classified as: Spam
```

Question 3 [16 points]

The provided dataset is a subset of the public data from the 2022 EPA Automotive Trends Report. It will be used to study the effects of various vehicle characteristics on CO2 emissions. The dataset consists of a dataframe with 1703 observations with the following 7 variables:

- Model.Year: year the vehicle model was produced (quantitative)
- Type: vehicle type (qualitative)
- MPG: miles per gallon of fuel (quantitative)
- Weight: vehicle weight in lbs (quantitative)
- Horsepower: vehicle horsepower in HP (quantitative)
- Acceleration: acceleration time (from 0 to 60 mph) in seconds (quantitative)
- CO2: carbon dioxide emissions in g/mi (response variable)

(1).[3 points] Read the data, Fit a multiple linear regression model called model1 using CO2 as the response and all predicting variables. Using $\alpha = 0.05$, which of the estimated coefficients that were statistically significant.

```
Answer:
```

```
data <- read.csv("emissions.csv")
head(data, 3)
```

```
##   Model.Year Type      MPG   Weight Horsepower Acceleration      CO2
## 1       2008  Van 20.39302 4500.000   241.8963       8.9064 435.7864
## 2       2009  Van 20.39440 4835.321   241.9881       9.0603 435.7570
## 3       2010  Van 20.17616 4840.431   244.0000       9.1116 440.4703
```

```
model1 <- lm(CO2 ~ Model.Year + Type + MPG + Weight + Horsepower + Accelera
tion, data = data)
summary(model1)
```

```
##
## Call:
## lm(formula = CO2 ~ Model.Year + Type + MPG + Weight + Horsepower +
##     Acceleration, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -49.969 -12.299  -4.040   6.126 242.173
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.250e+03  2.983e+02   4.190 2.94e-05 ***
## Model.Year   -2.859e-01  1.525e-01  -1.875   0.0609 .
## TypeSUV      -1.594e+01  2.199e+00  -7.249 6.35e-13 ***
## TypeTruck    -1.802e+01  2.201e+00  -8.186 5.24e-16 ***
## TypeVan      -3.082e+01  2.397e+00 -12.854  < 2e-16 ***
## MPG          -1.749e+01  3.507e-01 -49.882  < 2e-16 ***
## Weight        4.978e-02  2.588e-03  19.235  < 2e-16 ***
## Horsepower   -3.460e-01  2.918e-02 -11.858  < 2e-16 ***
## Acceleration  1.333e+00  5.516e-01   2.417   0.0158 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 23.3 on 1694 degrees of freedom
## Multiple R-squared:  0.9399, Adjusted R-squared:  0.9396
## F-statistic:  3309 on 8 and 1694 DF,  p-value: < 2.2e-16
```

Statistical Significance of Coefficients: At a significance level of α=0.05, the statistically significant predictors are: -(Intercept), TypeSUV, TypeTruck, TypeVan, MPG, Weight, Horsepower, and Acceleration. The variable Model. The year is not statistically significant at α=0.05

(2).[2 points] Is the overall regression (model1) significant at an α-level of 0.05? Explain how you determined the answer.

```
Answer:
Testing Overall Model Significance
```

The overall regressionn model (model1) is significant at an α-level of 0.05. The F-statistic and its associated p-value in the model summary help determine if at least one of the predictors is significantly associated with the response variable (CO2 emissions). Here, the F-statistic is very high (3309), and the corresponding p-value is less than 2.2e-16, which is far below 0.05. Therefore, we reject the null hypothesis that all coefficients are zero and conclude that the predictors collectively explain a significant ammount of variation in CO2 emissions.

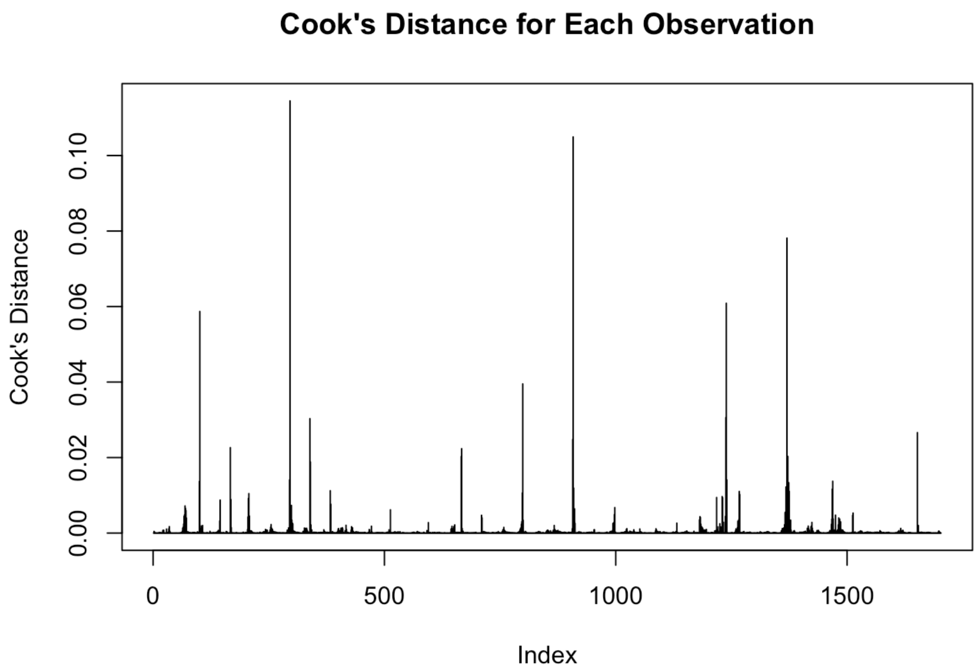
(3).[6 points] **Identifying Influential Data Points** Cook's Distances

The basic idea behind the measure is to delete the observations one at a time, each time refitting the regression model on the remaining $n - 1$ observations. Then, we compare the results using all n observations to the results with the ith observation deleted to see how much influence the observation has on the analysis. Analyzed as such, we are able to assess the potential impact each data point has on the regression analysis. One of such a method is called `Cook's distance`. To learn more on Cook's distance in R, see https://rpubs.com/DragonflyStats/Cooks-Distance (https://rpubs.com/DragonflyStats/Cooks-Distance).

Create a plot for the Cook's Distances (use model1). Using a threshold of 1, are there any outliers? If yes, which data points?

```
Answer:
```

```
cooksD <- cooks.distance(model1)
plot(cooksD, type = "h", main = "Cook's Distance for Each Observation", yla
b = "Cook's Distance")
abline(h = 1, col = "red")
```



Points with Cook's Distance greater than 1 are considered influential or outliers. In the plot of Cook's Distance, we observe that none of the points exceed the threshold of 1, which is marked by the red horizontal line. Cook's Distance values above this threshold would indicate potentially influentiial data points or outliers in the model. Since no points cross this threshold and,conclude that there are no significant outliers based on Cook's Distance at a threshold of 1.

(4).[5 points] **Detecting Multicollinearity Using Variance Inflation Factors (VIF)**

The effects that multicollinearity can have on our regression analyses and subsequent conclusions, how can we tell if multicollinearity is present in our data? A variance inflation factor exists for each of the predictors in a multiple regression model. For example, the variance inflation factor for the estimated regression coefficient β_j —denoted VIF_j —is just the factor by which the variance of β_j is "inflated" by the existence of correlation among the predictor variables in the model.

In particular, the variance inflation factor for the jth predictor is: $ VIF_j=$ where R_j^2 is the R^2 -value obtained by regressing the jth predictor on the remaining predictors.

A VIF of 1 means that there is no correlation among the jth predictor and the remaining predictor variables, and hence the variance of β_j is not inflated at all. The general rule of thumb is that VIFs exceeding 4 warrant further investigation, while VIFs exceeding 10 are signs of serious multicollinearity requiring correction. For more information, see https://search.r-project.org/CRAN/refmans/usdm/html/vif.html (https://search.r-project.org/CRAN/refmans/usdm/html/vif.html).

Calculate the VIF of each predictor (use model1). Using a threshold of $\max(10, \frac{1}{1-R^2})$ what conclusions can you make regarding multicollinearity?

```r
library(car)
```

```
## Loading required package: carData
```

```r
vif_values <- vif(model1)


print(vif_values)
```

```
##                   GVIF Df GVIF^(1/(2*Df))
## Model.Year   11.158497  1        3.340434
## Type          3.285619  3        1.219278
## MPG           7.387342  1        2.717967
## Weight       11.380489  1        3.373498
## Horsepower   11.459304  1        3.385159
## Acceleration  7.680964  1        2.771455
```

```r
r_squared <- summary(model1)$r.squared
threshold <- max(10, 1 / (1 - r_squared))


cat("VIF Threshold:", threshold, "\n")
```

```
## VIF Threshold: 16.62808
```

```r
high_vif <- vif_values[vif_values > threshold]


if (length(high_vif) > 0) {
  cat("Predictors with high multicollinearity:\n")
  print(high_vif)
} else {
  cat("No predictors exceed the VIF threshold, indicating multicollinearity
is not a major issue.\n")
}
```

```
## No predictors exceed the VIF threshold, indicating multicollinearity is
not a major issue.
```

Conclusion regarding multicollinearity: The conclusion is while no predictors exceed the calculated VIF threshold of 16.62808, there are still signs of considerable multicollinearity in the model. Model.Year, Weight, and Horsepower all have VIF values above 11, indicating strong correlations with other predictors. MPG and Acceleration show moderate levels of multicollinearity with VIF values between 7 and 8. Although the output states that "multicollinearity is not a major issue" based on the high threshold, it's important to note that these VIF values are still above the commonly used threshold of 10 for several variables. This suggests that multicollinearity should not be completely dismissed. While the current model may be acceptable for predictive purposes, caution should be exercised when interpreting individual coefficient effects, particularly for Model.Year, Weight, and Horsepower. For inferential purposes, the present multicollinearity could lead to unstable and potentially misleading coefficient estimates for the affected variables. It may be worthwhile to investigate the relationships between these highly correlated predictors and consider methods to address multicollinearity if precise individual coefficient estimates are crucial to the analysis.

Question 4 [16 points]

(1). Using the GermanCredit data set german.credit (Download the dataset from http://archive.ics.uci.edu/ml/datasets/Statlog+%28German+Credit+Data%29 (http://archive.ics.uci.edu/ml/datasets/Statlog+%28German+Credit+Data%29) and read the description), use logistic regression to find a good predictive model for whether credit applicants are good credit risks or not. Show your model (factors used and their coefficients), the output, and the quality of fit. You can use the glm function in R. To get a logistic regression (logit) model on data where the response is either zero or one, use family=binomial in your glm function call. Steps including:

a.[2 points] load the dataset

```
Answer:
```

```
url <- "http://archive.ics.uci.edu/ml/machine-learning-databases/statlog/ge
rman/german.data"
german_credit <- read.table(url, header = FALSE)
```

b.[4 points] explore the dataset, including summary of dataset, types of predictors, if there are categorical predictors, convert the predictors to factors.

```
Answer:
```

```
summary(german_credit)
```

```
##       V1                  V2            V3                  V4
##  Length:1000        Min.   : 4.0   Length:1000        Length:1000
##  Class :character   1st Qu.:12.0   Class :character   Class :character
##  Mode  :character   Median :18.0   Mode  :character   Mode  :character
##                     Mean   :20.9
##                     3rd Qu.:24.0
##                     Max.   :72.0
##        V5              V6                  V7                    V8
##  Min.   :  250   Length:1000        Length:1000        Min.   :1.000
##  1st Qu.: 1366   Class :character   Class :character   1st Qu.:2.000
##  Median : 2320   Mode  :character   Mode  :character   Median :3.000
##  Mean   : 3271                                         Mean   :2.973
##  3rd Qu.: 3972                                         3rd Qu.:4.000
##  Max.   :18424                                         Max.   :4.000
##       V9                 V10                   V11             V12
##  Length:1000        Length:1000        Min.   :1.000   Length:1000
##  Class :character   Class :character   1st Qu.:2.000   Class :character
##  Mode  :character   Mode  :character   Median :3.000   Mode  :character
##                                        Mean   :2.845
##                                        3rd Qu.:4.000
##                                        Max.   :4.000
##       V13             V14                 V15                  V16
##  Min.   :19.00   Length:1000        Length:1000        Min.   :1.000
##  1st Qu.:27.00   Class :character   Class :character   1st Qu.:1.000
##  Median :33.00   Mode  :character   Mode  :character   Median :1.000
##  Mean   :35.55                                         Mean   :1.407
##  3rd Qu.:42.00                                         3rd Qu.:2.000
##  Max.   :75.00                                         Max.   :4.000
##      V17                  V18            V19                V20
##  Length:1000        Min.   :1.000   Length:1000        Length:1000
##  Class :character   1st Qu.:1.000   Class :character   Class :character
##  Mode  :character   Median :1.000   Mode  :character   Mode  :character
##                     Mean   :1.155
##                     3rd Qu.:1.000
##                     Max.   :2.000
##       V21
##  Min.   :1.0
##  1st Qu.:1.0
##  Median :1.0
##  Mean   :1.3
##  3rd Qu.:2.0
##  Max.   :2.0
```

```
str(german_credit)
```

```
## 'data.frame':    1000 obs. of  21 variables:
##  $ V1 : chr  "A11" "A12" "A14" "A11" ...
##  $ V2 : int  6 48 12 42 24 36 24 36 12 30 ...
##  $ V3 : chr  "A34" "A32" "A34" "A32" ...
##  $ V4 : chr  "A43" "A43" "A46" "A42" ...
##  $ V5 : int  1169 5951 2096 7882 4870 9055 2835 6948 3059 5234 ...
##  $ V6 : chr  "A65" "A61" "A61" "A61" ...
##  $ V7 : chr  "A75" "A73" "A74" "A74" ...
##  $ V8 : int  4 2 2 2 3 2 3 2 2 4 ...
##  $ V9 : chr  "A93" "A92" "A93" "A93" ...
##  $ V10: chr  "A101" "A101" "A101" "A103" ...
##  $ V11: int  4 2 3 4 4 4 4 2 4 2 ...
##  $ V12: chr  "A121" "A121" "A121" "A122" ...
##  $ V13: int  67 22 49 45 53 35 53 35 61 28 ...
##  $ V14: chr  "A143" "A143" "A143" "A143" ...
##  $ V15: chr  "A152" "A152" "A152" "A153" ...
##  $ V16: int  2 1 1 1 2 1 1 1 1 2 ...
##  $ V17: chr  "A173" "A173" "A172" "A173" ...
##  $ V18: int  1 1 2 2 2 2 1 1 1 1 ...
##  $ V19: chr  "A192" "A191" "A191" "A191" ...
##  $ V20: chr  "A201" "A201" "A201" "A201" ...
##  $ V21: int  1 2 1 1 2 1 1 1 1 2 ...
```

```
german_credit$V1 <- as.factor(german_credit$V1)
german_credit$V4 <- as.factor(german_credit$V4)
```

c.[2 points] Column V21 represents the target, 1 = Good, 2 = Bad, convert value the values to 0 and 1, respectively.

```
Answer:
```

```
german_credit$V21 <- ifelse(german_credit$V21 == 1, 0, 1)
```

d.[2 points] split the dataset to taining and test dataset with 90% and 10% of the data points, respectively.

```
Answer:
```

```
set.seed(123)


sample_index <- sample(seq_len(nrow(german_credit)), size = 0.9 * nrow(germ
an_credit))
train_data <- german_credit[sample_index, ]
test_data <- german_credit[-sample_index, ]
```

e.[3 points] use the training dataset to get a logistic regression (logit) model on data where the response is either zero or one, use family=binomial in your glm function call.

```
Answer:
```

```
credit_model <- glm(V21 ~ ., data = train_data, family = binomial)
summary(credit_model)
```

```
##
## Call:
## glm(formula = V21 ~ ., family = binomial, data = train_data)
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.917e-02  1.183e+00  -0.016 0.987074
## V1A12       -3.695e-01  2.326e-01  -1.589 0.112109
## V1A13       -9.616e-01  3.850e-01  -2.497 0.012514 *
## V1A14       -1.733e+00  2.478e-01  -6.992 2.71e-12 ***
## V2           2.934e-02  9.775e-03   3.001 0.002690 **
## V3A31        2.586e-01  5.831e-01   0.444 0.657366
## V3A32       -6.742e-01  4.595e-01  -1.467 0.142289
## V3A33       -7.987e-01  4.995e-01  -1.599 0.109803
## V3A34       -1.365e+00  4.628e-01  -2.950 0.003179 **
## V4A41       -1.569e+00  3.916e-01  -4.008 6.12e-05 ***
## V4A410      -1.619e+00  8.619e-01  -1.879 0.060311 .
## V4A42       -7.934e-01  2.804e-01  -2.830 0.004657 **
## V4A43       -8.644e-01  2.652e-01  -3.260 0.001115 **
## V4A44        3.993e-02  8.209e-01   0.049 0.961207
## V4A45        5.813e-02  6.030e-01   0.096 0.923197
## V4A46       -2.251e-01  4.231e-01  -0.532 0.594736
## V4A48       -2.077e+00  1.298e+00  -1.600 0.109540
## V4A49       -8.363e-01  3.551e-01  -2.355 0.018505 *
## V5           1.355e-04  4.669e-05   2.903 0.003700 **
## V6A62       -4.701e-01  3.074e-01  -1.530 0.126119
## V6A63       -3.594e-01  4.081e-01  -0.881 0.378553
## V6A64       -1.347e+00  5.340e-01  -2.522 0.011656 *
## V6A65       -9.795e-01  2.861e-01  -3.423 0.000618 ***
## V7A72       -2.944e-02  4.612e-01  -0.064 0.949112
## V7A73       -1.495e-01  4.455e-01  -0.336 0.737190
## V7A74       -9.106e-01  4.797e-01  -1.898 0.057662 .
## V7A75       -3.903e-01  4.479e-01  -0.871 0.383516
## V8           4.103e-01  9.484e-02   4.327 1.51e-05 ***
## V9A92       -2.937e-01  4.151e-01  -0.708 0.479195
## V9A93       -9.543e-01  4.113e-01  -2.320 0.020342 *
## V9A94       -2.619e-01  4.908e-01  -0.534 0.593616
## V10A102      3.632e-01  4.391e-01   0.827 0.408117
## V10A103     -1.026e+00  4.564e-01  -2.249 0.024536 *
## V11          1.706e-03  9.231e-02   0.018 0.985253
## V12A122      2.874e-01  2.735e-01   1.051 0.293261
## V12A123      3.396e-01  2.537e-01   1.339 0.180608
## V12A124      9.598e-01  4.704e-01   2.040 0.041322 *
## V13         -1.265e-02  9.786e-03  -1.292 0.196252
## V14A142     -3.761e-01  4.556e-01  -0.825 0.409185
## V14A143     -6.030e-01  2.521e-01  -2.391 0.016787 *
## V15A152     -5.047e-01  2.510e-01  -2.011 0.044366 *
## V15A153     -8.100e-01  5.160e-01  -1.570 0.116441
## V16          2.892e-01  2.090e-01   1.384 0.166410
## V17A172      5.343e-01  7.415e-01   0.721 0.471172
## V17A173      4.552e-01  7.156e-01   0.636 0.524682
## V17A174      1.752e-01  7.158e-01   0.245 0.806650
## V18          4.267e-01  2.623e-01   1.627 0.103766
## V19A192     -2.798e-01  2.145e-01  -1.304 0.192089
## V20A202     -1.538e+00  7.191e-01  -2.138 0.032482 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
```

```
##
##     Null deviance: 1090.95  on 899  degrees of freedom
## Residual deviance:  792.88  on 851  degrees of freedom
## AIC: 890.88
##
## Number of Fisher Scoring iterations: 5
```

f.[4 points] use the model to make prediction on the the training dataset, and test dataset, give the confusion matrices and accuracy for each dataset.

```
Answer:
```

```
train_pred <- predict(credit_model, train_data, type = "response")
test_pred <- predict(credit_model, test_data, type = "response")


train_pred_class <- ifelse(train_pred > 0.5, 1, 0)
test_pred_class <- ifelse(test_pred > 0.5, 1, 0)


print(length(train_pred_class))
```

```
## [1] 900
```

```
print(length(train_data$V21))
```

```
## [1] 900
```

```
print(length(test_pred_class))
```

```
## [1] 100
```

```
print(length(test_data$V21))
```

```
## [1] 100
```

```
train_conf_matrix <- table(Predicted = train_pred_class, Actual = train_dat
a$V21)
test_conf_matrix <- table(Predicted = test_pred_class, Actual = test_data$V
21)

print("Training Confusion Matrix:")
```

```
## [1] "Training Confusion Matrix:"
```

```
print(train_conf_matrix)
```

```
##          Actual
## Predicted   0   1
##         0 566 122
##         1  69 143
```

```
print("Test Confusion Matrix:")
```

```
## [1] "Test Confusion Matrix:"
```

```
print(test_conf_matrix)
```

```
##          Actual
## Predicted  0  1
##         0 58 17
##         1  7 18
```

```
train_accuracy <- sum(diag(train_conf_matrix)) / sum(train_conf_matrix)
test_accuracy <- sum(diag(test_conf_matrix)) / sum(test_conf_matrix)
cat("train accuracy",train_accuracy,"\n")
```

```
## train accuracy 0.7877778
```

```
cat("test accuracy",test_accuracy)
```

```
## test accuracy 0.76
```

(2). [4 points] Because the model gives a result between 0 and 1, it requires setting a threshold probability to separate between "good" and "bad" answers. In this data set, they estimate that incorrectly identifying a bad customer as good, is 5 times worse than incorrectly classifying a good customer as bad. Determine a good threshold probability based on your model (please demonstrate your reasoning- The optimal threshold of 0.1187497 has been selected based on the premise that misclassifying a bad customer as good is five times worse than misclassifying a good customer as bad. This imbalance in the cost of errors calls for an adjustment in the threshold to minimize the more severe consequence (false positives). A lower threshold increases the likelihoodd of misclassifying bad customers as good, which is a costly error. By setting the threshold at 0.1187497, the model minimizes the overall cost by reducing false positives while still balancing the risk of false negatives. This threshold is considered optimal as it accounts for the cost differential in misclassiffication, ensuring the most efficient performance under the given penalty structure.)

```
Answer:
```

```
library(ROCR)


pred <- prediction(train_pred, train_data$V21)
perf <- performance(pred, "tpr", "fpr")


thresholds <- perf@alpha.values[[1]]
costs <- 5 * (1 - perf@y.values[[1]]) + perf@x.values[[1]]  # Assign cost r
atio


optimal_threshold <- thresholds[which.min(costs)]
cat("optimal_threshold:",optimal_threshold)
```

```
## optimal_threshold: 0.1187497
```

Question 5 [28 points]

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.

(1).[2 points] 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.

```
Answer:
```



```
library(ISLR)
data("Auto")
head(Auto,5)
```

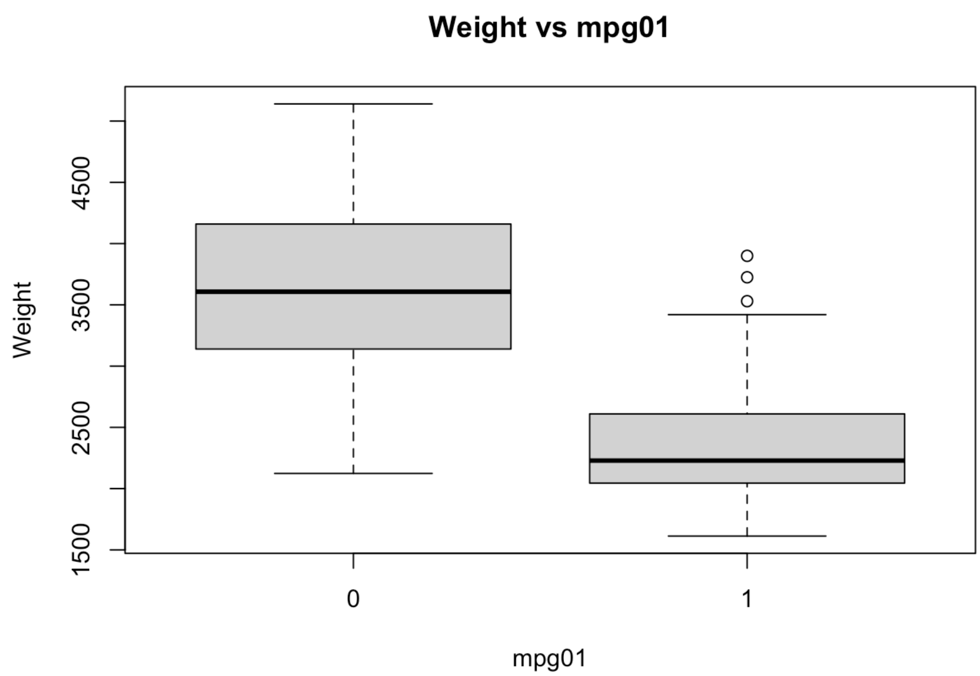
```
##   mpg cylinders displacement horsepower weight acceleration year origin
## 1  18         8          307        130   3504         12.0   70      1
## 2  15         8          350        165   3693         11.5   70      1
## 3  18         8          318        150   3436         11.0   70    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

```
median_mpg <- median(Auto$mpg)
Auto$mpg01 <- ifelse(Auto$mpg > median_mpg, 1, 0)
```

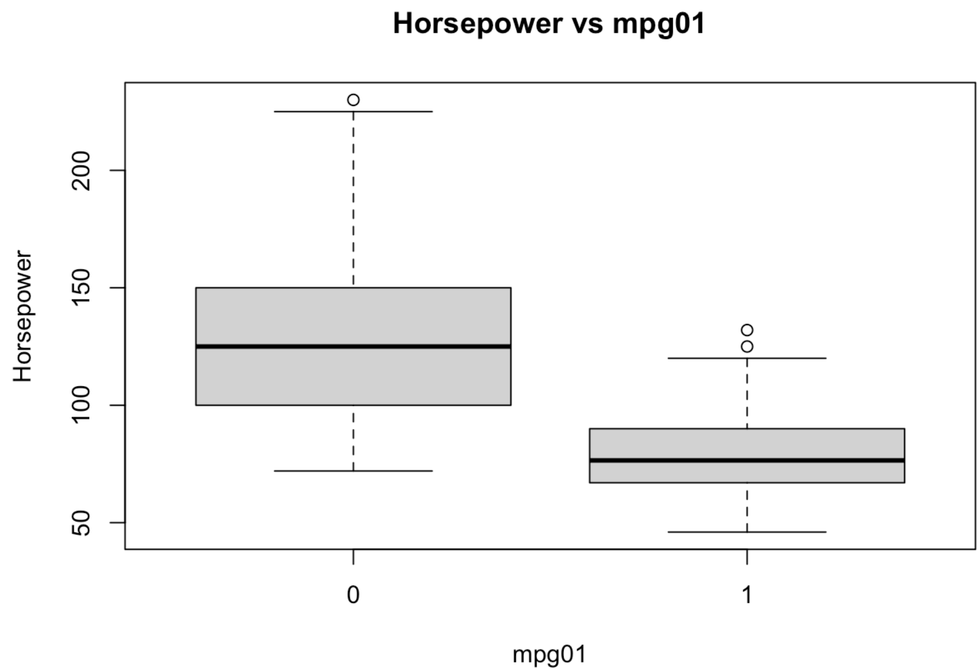
(2).[4 points] Explore the data graphically in order to investigate the association between `mpg01` and the other features. Which of the other features seem most likely to be useful in predicting `mpg01` ? Scatterplots and boxplots may be useful tools to answer this question. Describe your findings.

```
Answer:
```

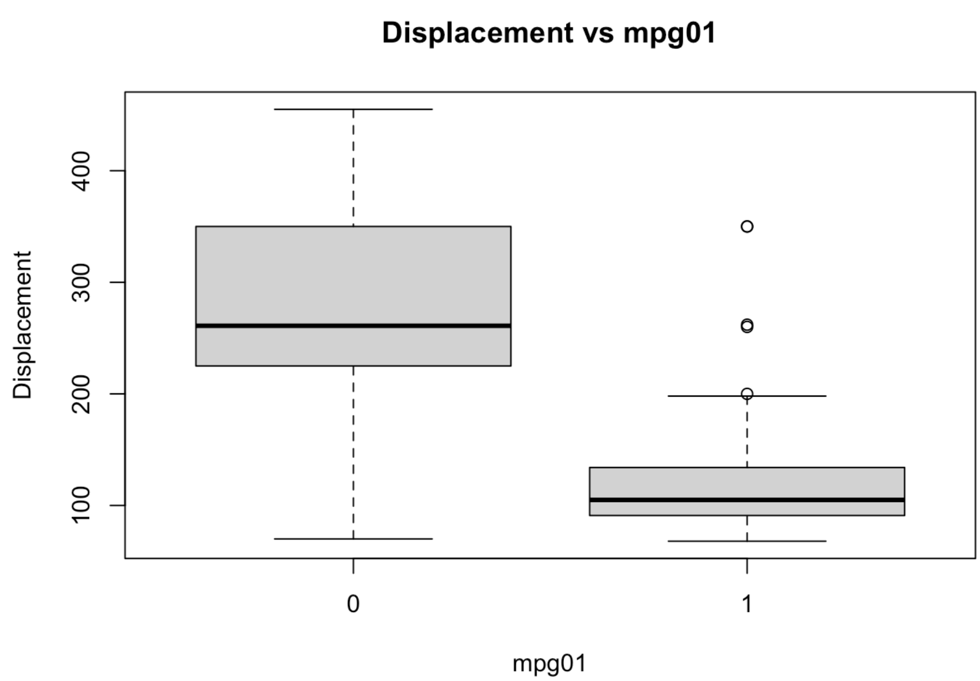
```r
library(ISLR)
data(Auto)


mpg01_median <- median(Auto$mpg)
Auto$mpg01 <- ifelse(Auto$mpg > mpg01_median, 1, 0)


boxplot(Auto$weight ~ Auto$mpg01,
        main = "Weight vs mpg01",
        xlab = "mpg01",
        ylab = "Weight")
```



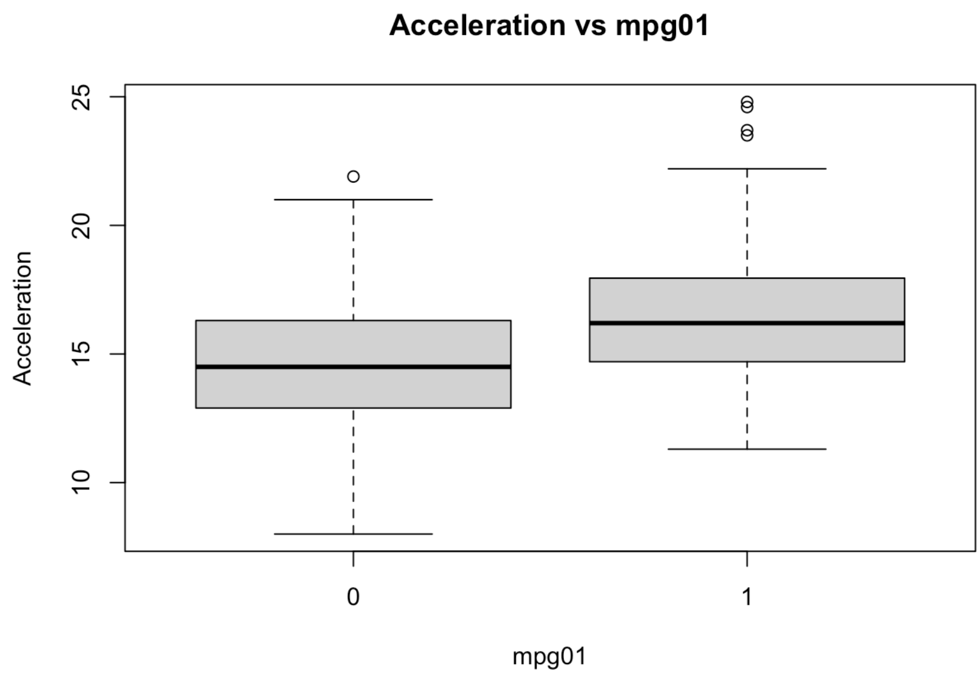

```r
boxplot(Auto$horsepower ~ Auto$mpg01,
        main = "Horsepower vs mpg01",
        xlab = "mpg01",
        ylab = "Horsepower")
```



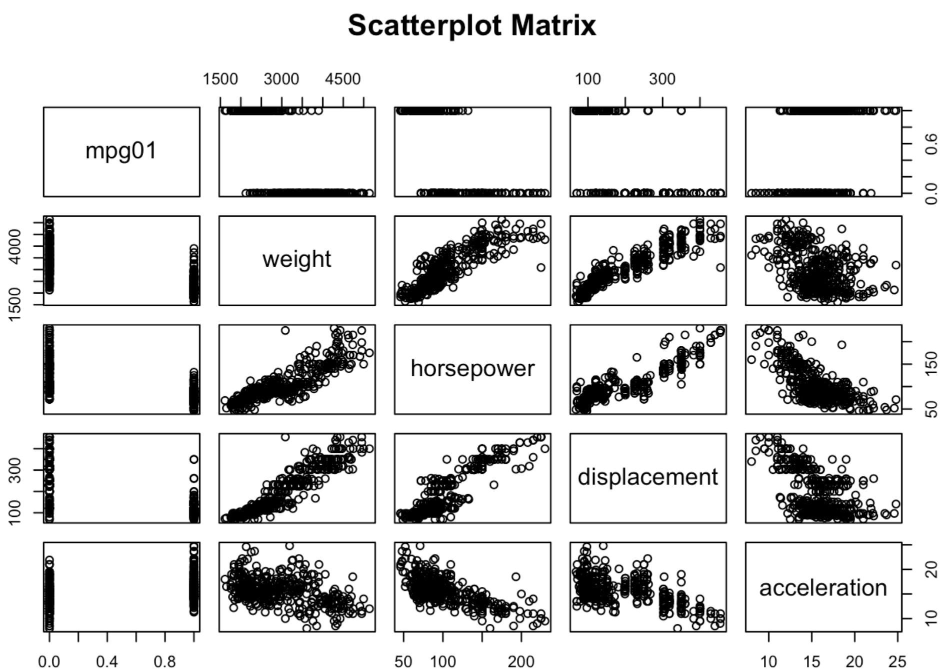

```
boxplot(Auto$displacement ~ Auto$mpg01,
        main = "Displacement vs mpg01",
        xlab = "mpg01",
        ylab = "Displacement")
```



```
boxplot(Auto$acceleration ~ Auto$mpg01,
        main = "Acceleration vs mpg01",
        xlab = "mpg01",
        ylab = "Acceleration")
```




```
pairs(Auto[, c("mpg01", "weight", "horsepower", "displacement", "accelerati
on")],
      main = "Scatterplot Matrix")
```



Findinggs:- From the boxplots and scatterplot matrix provided, we can observe several important associations between mpg01 (a binary variable representing whether a car has high or low miles per gallon) and other features such as weight, horsepower, displacement, and acceleration. Displacement vs mpg01: The boxplot shows that cars with lower mpg01 (less fuel-efficient cars) tend to have significantly higher displacement. On the other hand, cars with higher mpg01 (more fuel-efficient cars) have much lower displacement. This suggests that displacement is a strong predictor of mpg01, with larger engine displacement associated with lower fuel efficiency. Acceleration vs mpg01: The boxplot for acceleration shows that the distribution of acceleration is somewhat similar between the two groups of mpg01. However, cars with higher mpg01 (more fuel-efficient) tend to have slightly higher acceleration on average. This indicates that acceleration may not be as strong a predictor of mpg01 compared to other features like weight or horsepower, but there is still a slight trend where more fuel-efficient cars have better acceleration. Weight vs mpg01: The boxplot for weight shows a clear distinction between the two categories of mpg01. Cars with lower mpg01 (less fuel-efficient) are significantly heavier, while cars with higher mpg01 (more fuel-efficient) are lighter. This suggests that weight is a strong predictor of mpg01, as heavier cars tend to have lower fuel efficiency. Horsepower vs mpg01: The boxplot for horsepower reveals that cars with lower mpg01 (less fuel-efficient) tend to have much higher horsepower compared to cars with higher mpg01. This suggests that horsepower is another strong predictor of mpg01, as more powerful engines are typically associated with lower fuel efficiency. Conclusion: From the boxplots, it is evident that weight, horsepower, and displacement are the most useful features in predicting mpg01. Heavier cars, those with higher horsepower, and those with larger engine displacement are generally less fuel efficient (mpg01 = 0). While acceleration shows some association, it appears to be a weaker predictor compared to the other features.

(3).[2 points] Split the data into a training set and a test set.

```
Answer:
```

```
set.seed(123)


train_index <- sample(seq_len(nrow(Auto)), size = 0.8 * nrow(Auto))


train_data <- Auto[train_index, ]
test_data <- Auto[-train_index, ]

test_data
```

```
##      mpg cylinders displacement horsepower weight acceleration year orig
in
## 1   18.0         8        307.0        130   3504         12.0   70
1
## 3   18.0         8        318.0        150   3436         11.0   70
1
## 6   15.0         8        429.0        198   4341         10.0   70
1
## 15  24.0         4        113.0         95   2372         15.0   70
3
## 17  18.0         6        199.0         97   2774         15.5   70
1
## 18  21.0         6        200.0         85   2587         16.0   70
1
## 19  27.0         4         97.0         88   2130         14.5   70
3
## 28  11.0         8        318.0        210   4382         13.5   70
1
## 39  14.0         8        350.0        165   4209         12.0   71
1
## 50  23.0         4        122.0         86   2220         14.0   71
1
## 57  26.0         4         91.0         70   1955         20.5   71
1
## 59  25.0         4         97.5         80   2126         17.0   72
1
## 61  20.0         4        140.0         90   2408         19.5   72
1
## 63  13.0         8        350.0        165   4274         12.0   72
1
## 66  14.0         8        351.0        153   4129         13.0   72
1
## 69  13.0         8        350.0        155   4502         13.5   72
1
## 71  13.0         8        400.0        190   4422         12.5   72
1
## 72  19.0         3         70.0         97   2330         13.5   72
3
## 76  14.0         8        318.0        150   4077         14.0   72
1
## 80  26.0         4         96.0         69   2189         18.0   72
2
## 89  14.0         8        302.0        137   4042         14.5   73
1
## 100 18.0         6        232.0        100   2945         16.0   73
1
## 101 18.0         6        250.0         88   3021         16.5   73
```

```
1
## 104 11.0         8        400.0        150   4997         14.0   73
1
## 109 20.0         4         97.0         88   2279         19.0   73
3
## 115 26.0         4         98.0         90   2265         15.5   73
2
## 120 20.0         4        114.0         91   2582         14.0   73
2
## 123 24.0         4        121.0        110   2660         14.0   73
2
## 125 11.0         8        350.0        180   3664         11.0   73
1
## 133 25.0         4        140.0         75   2542         17.0   74
1
## 134 16.0         6        250.0        100   3781         17.0   74
1
## 140 14.0         8        302.0        140   4638         16.0   74
1
## 141 14.0         8        304.0        150   4257         15.5   74
1
## 142 29.0         4         98.0         83   2219         16.5   74
2
## 149 26.0         4        116.0         75   2246         14.0   74
2
## 152 31.0         4         79.0         67   2000         16.0   74
2
## 164 18.0         6        225.0         95   3785         19.0   75
1
## 171 23.0         4        140.0         78   2592         18.5   75
1
## 172 24.0         4        134.0         96   2702         13.5   75
3
## 182 33.0         4         91.0         53   1795         17.5   75
3
## 183 28.0         4        107.0         86   2464         15.5   76
2
## 184 25.0         4        116.0         81   2220         16.9   76
2
## 191 14.5         8        351.0        152   4215         12.8   76
1
## 193 22.0         6        250.0        105   3353         14.5   76
1
## 194 24.0         6        200.0         81   3012         17.6   76
1
## 203 17.5         6        258.0         95   3193         17.8   76
1
## 205 32.0         4         85.0         70   1990         17.0   76
3
## 210 19.0         4        120.0         88   3270         21.9   76
2
## 227 20.5         6        231.0        105   3425         16.9   77
1
## 229 18.5         6        250.0         98   3525         19.0   77
1
## 230 16.0         8        400.0        180   4220         11.1   77
1
## 233 16.0         8        351.0        149   4335         14.5   77
1
```

```
## 247 32.8         4         78.0         52   1985          19.4   78
3
## 249 36.1         4         91.0         60   1800          16.4   78
3
## 252 20.2         8        302.0        139   3570          12.8   78
1
## 260 20.8         6        200.0         85   3070          16.7   78
1
## 263 19.2         8        305.0        145   3425          13.2   78
1
## 271 21.1         4        134.0         95   2515          14.8   78
3
## 281 21.5         6        231.0        115   3245          15.4   79
1
## 284 20.2         6        232.0         90   3265          18.2   79
1
## 294 31.9         4         89.0         71   1925          14.0   79
2
## 298 25.4         5        183.0         77   3530          20.1   79
2
## 299 23.0         8        350.0        125   3900          17.4   79
1
## 302 34.2         4        105.0         70   2200          13.2   79
1
## 305 37.3         4         91.0         69   2130          14.7   79
2
## 317 19.1         6        225.0         90   3381          18.7   80
1
## 324 27.9         4        156.0        105   2800          14.4   80
1
## 327 43.4         4         90.0         48   2335          23.7   80
2
## 334 32.7         6        168.0        132   2910          11.4   80
3
## 346 35.1         4         81.0         60   1760          16.1   81
3
## 357 32.4         4        108.0         75   2350          16.8   81
3
## 366 20.2         6        200.0         88   3060          17.1   81
1
## 367 17.6         6        225.0         85   3465          16.6   81
1
## 372 29.0         4        135.0         84   2525          16.0   82
1
## 373 27.0         4        151.0         90   2735          18.0   82
1
## 381 36.0         4        107.0         75   2205          14.5   82
3
## 384 32.0         4         91.0         67   1965          15.7   82
3
## 390 32.0         4        144.0         96   2665          13.9   82
3
## 393 27.0         4        140.0         86   2790          15.6   82
1
##                                 name mpg01
## 1          chevrolet chevelle malibu     0
## 3                 plymouth satellite     0
## 6                   ford galaxie 500     0
## 15              toyota corona mark ii     1
```

```
## 17                         amc hornet     0
## 18                      ford maverick     0
## 19                       datsun pl510     1
## 28                         dodge d200     0
## 39                   chevrolet impala     0
## 50                 mercury capri 2000     1
## 57                  plymouth cricket     1
## 59                 dodge colt hardtop     1
## 61                     chevrolet vega     0
## 63                   chevrolet impala     0
## 66                  ford galaxie 500     0
## 69                buick lesabre custom     0
## 71              chrysler newport royal     0
## 72                     mazda rx2 coupe     0
## 76       plymouth satellite custom (sw)     0
## 80                   renault 12 (sw)     1
## 89                  ford gran torino     0
## 100                        amc hornet     0
## 101                     ford maverick     0
## 104                  chevrolet impala     0
## 109                     toyota carina     0
## 115               fiat 124 sport coupe     1
## 120                        audi 100ls     0
## 123                         saab 99le     1
## 125                   oldsmobile omega     0
## 133                    chevrolet vega     1
## 134 chevrolet chevelle malibu classic     0
## 140              ford gran torino (sw)     0
## 141                  amc matador (sw)     0
## 142                          audi fox     1
## 149                       fiat 124 tc     1
## 152                         fiat x1.9     1
## 164                     plymouth fury     0
## 171                     pontiac astro     1
## 172                     toyota corona     1
## 182                  honda civic cvcc     1
## 183                          fiat 131     1
## 184                         opel 1900     1
## 191                  ford gran torino     0
## 193                    chevrolet nova     0
## 194                     ford maverick     1
## 203                      amc pacer d/l     0
## 205                      datsun b-210     1
## 210                       peugeot 504     0
## 227                      buick skylark     0
## 229                      ford granada     0
## 230             pontiac grand prix lj     0
## 233                  ford thunderbird     0
## 247                   mazda glc deluxe     1
## 249                  honda civic cvcc     1
## 252               mercury monarch ghia     0
## 260                    mercury zephyr     0
## 263        chevrolet monte carlo landau     0
## 271          toyota celica gt liftback     0
## 281                 pontiac lemans v6     0
## 284                   amc concord dl 6     0
## 294                  vw rabbit custom     1
## 298                mercedes benz 300d     1
## 299                  cadillac eldorado     1
```

```
## 302                 plymouth horizon     1
## 305               fiat strada custom     1
## 317                      dodge aspen     0
## 324                       dodge colt     1
## 327               vw dasher (diesel)     1
## 334                     datsun 280-zx     1
## 346                 honda civic 1300     1
## 357                   toyota corolla     1
## 366                  ford granada gl     0
## 367           chrysler lebaron salon     0
## 372                   dodge aries se     1
## 373                  pontiac phoenix     1
## 381                     honda accord     1
## 384               honda civic (auto)     1
## 390                 toyota celica gt     1
## 393                  ford mustang gl     1
```

```
train_data
```

```
##      mpg cylinders displacement horsepower weight acceleration year orig
in
## 181 25.0         4          121        115   2671         13.5   75
2
## 14  14.0         8          455        225   3086         10.0   70
1
## 197 24.5         4           98         60   2164         22.1   76
1
## 308 26.8         6          173        115   2700         12.9   79
1
## 119 24.0         4          116         75   2158         15.5   73
2
## 301 23.9         8          260         90   3420         22.2   79
1
## 231 15.5         8          350        170   4165         11.4   77
1
## 246 36.1         4           98         66   1800         14.4   78
1
## 396 28.0         4          120         79   2625         18.6   82
1
## 379 36.0         4           98         70   2125         17.3   82
1
## 155 15.0         6          250         72   3432         21.0   75
1
## 91  12.0         8          429        198   4952         11.5   73
1
## 92  13.0         8          400        150   4464         12.0   73
1
## 258 19.4         6          232         90   3210         17.2   78
1
## 199 33.0         4           91         53   1795         17.4   76
3
## 385 38.0         4           91         67   1995         16.2   82
3
## 352 34.4         4           98         65   2045         16.2   81
1
## 139 14.0         8          318        150   4457         13.5   74
1
## 360 28.1         4          141         80   3230         20.4   81
```

```
2
## 330 44.6         4           91          67   1850          13.8   80
3
## 26  10.0         8          360         215   4615          14.0   70
1
## 7   14.0         8          454         220   4354           9.0   70
1
## 380 36.0         4          120          88   2160          14.5   82
3
## 256 25.1         4          140          88   2720          15.4   78
1
## 213 16.5         8          350         180   4380          12.1   76
1
## 79  21.0         4          120          87   2979          19.5   72
2
## 82  28.0         4           97          92   2288          17.0   72
3
## 44  13.0         8          400         170   4746          12.0   71
1
## 364 22.4         6          231         110   3415          15.8   81
1
## 335 23.7         3           70         100   2420          12.5   80
3
## 145 31.0         4           76          52   1649          16.5   74
3
## 32  25.0         4          113          95   2228          14.0   71
3
## 110 21.0         4          140          72   2401          19.5   73
1
## 265 18.1         8          302         139   3205          11.2   78
1
## 333 29.8         4           89          62   1845          15.3   80
2
## 23  25.0         4          104          95   2375          17.5   70
2
## 311 38.1         4           89          60   1968          18.8   80
3
## 137 16.0         8          302         140   4141          14.0   74
1
## 361 30.7         6          145          76   3160          19.6   81
2
## 226 17.5         6          250         110   3520          16.4   77
1
## 168 29.0         4           97          75   2171          16.0   75
3
## 219 36.0         4           79          58   1825          18.6   77
2
## 292 19.2         8          267         125   3605          15.0   79
1
## 70  12.0         8          350         160   4456          13.5   72
1
## 73  15.0         8          304         150   3892          12.5   72
1
## 77  18.0         4          121         112   2933          14.5   72
2
## 64  14.0         8          400         175   4385          12.0   72
1
## 143 26.0         4           79          67   1963          15.5   74
2
```

```
## 212 16.5         6           168         120   3820           16.7   76
2
## 387 38.0         6           262          85   3015           17.0   82
1
## 296 35.7         4            98          80   1915           14.4   79
1
## 279 31.5         4            89          71   1990           14.9   78
2
## 42  14.0         8           318         150   4096           13.0   71
1
## 386 25.0         6           181         110   2945           16.4   82
1
## 318 34.3         4            97          78   2188           15.8   80
2
## 225 15.0         8           302         130   4295           14.9   77
1
## 16  22.0         6           198          95   2833           15.5   70
1
## 117 16.0         8           400         230   4278            9.5   73
1
## 95  13.0         8           440         215   4735           11.0   73
1
## 264 17.7         6           231         165   3445           13.4   78
1
## 237 25.5         4           140          89   2755           15.8   77
1
## 87  14.0         8           304         150   3672           11.5   73
1
## 40  14.0         8           400         175   4464           11.5   71
1
## 161 17.0         6           231         110   3907           21.0   75
1
## 242 22.0         6           146          97   2815           14.5   77
3
## 211 19.0         6           156         108   2930           15.5   76
3
## 394 44.0         4            97          52   2130           24.6   82
2
## 35  16.0         6           225         105   3439           15.5   71
1
## 4   16.0         8           304         150   3433           12.0   70
1
## 13  15.0         8           400         150   3761            9.5   70
1
## 353 29.9         4            98          65   2380           20.7   81
1
## 245 43.1         4            90          48   1985           21.5   78
2
## 310 41.5         4            98          76   2144           14.7   80
2
## 280 29.5         4            98          68   2135           16.6   78
3
## 90  15.0         8           318         150   3777           12.5   73
1
## 25  21.0         6           199          90   2648           15.0   70
1
## 293 18.5         8           360         150   3940           13.0   79
1
## 288 16.5         8           351         138   3955           13.2   79
```

```
1
## 332 33.8         4           97          67   2145          18.0   80
3
## 122 15.0         8          318         150   3399          11.0   73
1
## 111 22.0         4          108          94   2379          16.5   73
3
## 160 14.0         8          351         148   4657          13.5   75
1
## 65  15.0         8          318         150   4135          13.5   72
1
## 201 18.0         6          250          78   3574          21.0   76
1
## 68  11.0         8          429         208   4633          11.0   72
1
## 153 19.0         6          225          95   3264          16.0   75
1
## 86  13.0         8          350         175   4100          13.0   73
1
## 167 13.0         8          302         129   3169          12.0   75
1
## 138 13.0         8          350         150   4699          14.5   74
1
## 52  30.0         4           79          70   2074          19.5   71
2
## 75  13.0         8          302         140   4294          16.0   72
1
## 180 22.0         4          121          98   2945          14.5   75
2
## 238 30.5         4           98          63   2051          17.0   77
1
## 99  16.0         6          250         100   3278          18.0   73
1
## 216 13.0         8          318         150   3755          14.0   76
1
## 129 15.0         6          250         100   3336          17.0   74
1
## 214 13.0         8          350         145   4055          12.0   76
1
## 176 29.0         4           90          70   1937          14.0   75
2
## 275 20.3         5          131         103   2830          15.9   78
2
## 234 29.0         4           97          78   1940          14.5   77
2
## 323 46.6         4           86          65   2110          17.9   80
3
## 282 19.8         6          200          85   2990          18.2   79
1
## 114 21.0         6          155         107   2472          14.0   73
1
## 108 18.0         6          232         100   2789          15.0   73
1
## 309 33.5         4          151          90   2556          13.2   79
1
## 156 15.0         6          250          72   3158          19.5   75
1
## 103 26.0         4           97          46   1950          21.0   73
2
```

```
## 257 20.5          6           225         100   3430          17.2   78
1
## 162 16.0          6           250         105   3897          18.5   75
1
## 157 16.0          8           400         170   4668          11.5   75
1
## 5   17.0          8           302         140   3449          10.5   70
1
## 274 23.9          4           119          97   2405          14.9   78
3
## 342 23.5          6           173         110   2725          12.6   81
1
## 345 39.0          4            86          64   1875          16.4   81
1
## 56  27.0          4            97          60   1834          19.0   71
2
## 240 30.0          4            97          67   1985          16.4   77
3
## 254 20.5          6           200          95   3155          18.2   78
1
## 363 24.2          6           146         120   2930          13.8   81
3
## 375 36.0          4           105          74   1980          15.3   82
2
## 228 19.0          6           225         100   3630          17.7   77
1
## 49  18.0          6           250          88   3139          14.5   71
1
## 78  22.0          4           121          76   2511          18.0   72
2
## 84  28.0          4            98          80   2164          15.0   72
1
## 186 26.0          4            98          79   2255          17.7   76
1
## 277 21.6          4           121         115   2795          15.7   78
2
## 198 29.0          4            90          70   1937          14.2   76
2
## 259 20.6          6           231         105   3380          15.8   78
1
## 170 20.0          6           232         100   2914          16.0   75
1
## 338 32.4          4           107          72   2290          17.0   80
3
## 20  26.0          4            97          46   1835          20.5   70
2
## 395 32.0          4           135          84   2295          11.6   82
1
## 166 20.0          8           262         110   3221          13.5   75
1
## 53  30.0          4            88          76   2065          14.5   71
2
## 22  24.0          4           107          90   2430          14.5   70
2
## 179 23.0          4           120          88   2957          17.0   75
2
## 43  12.0          8           383         180   4955          11.5   71
1
## 60  23.0          4            97          54   2254          23.5   72
```

```
2
## 85  27.0          4            97          88   2100          16.5   72
3
## 11  15.0          8           383         170   3563          10.0   70
1
## 315 26.4          4           140          88   2870          18.1   80
1
## 185 25.0          4           140          92   2572          14.9   76
1
## 306 28.4          4           151          90   2670          16.0   79
1
## 47  22.0          4           140          72   2408          19.0   71
1
## 328 36.4          5           121          67   2950          19.9   80
2
## 365 26.6          8           350         105   3725          19.0   81
1
## 196 29.0          4            85          52   2035          22.2   76
1
## 291 15.5          8           351         142   4054          14.3   79
1
## 273 23.8          4           151          85   2855          17.6   78
1
## 297 27.4          4           121          80   2670          15.0   79
1
## 200 20.0          6           225         100   3651          17.7   76
1
## 202 18.5          6           250         110   3645          16.2   76
1
## 174 24.0          4           119          97   2545          17.0   75
3
## 286 17.0          8           305         130   3840          15.4   79
1
## 37  19.0          6           250          88   3302          15.5   71
1
## 175 18.0          6           171          97   2984          14.5   75
1
## 144 26.0          4            97          78   2300          14.5   74
2
## 340 26.6          4           151          84   2635          16.4   81
1
## 217 31.5          4            98          68   2045          18.5   77
3
## 126 20.0          6           198          95   3102          16.5   74
1
## 34  19.0          6           232         100   2634          13.0   71
1
## 41  14.0          8           351         153   4154          13.5   71
1
## 267 30.0          4            98          68   2155          16.5   78
1
## 10  15.0          8           390         190   3850           8.5   70
1
## 356 33.7          4           107          75   2210          14.4   81
3
## 244 21.5          3            80         110   2720          13.5   77
3
## 348 37.0          4            85          65   1975          19.4   81
3
```

```
## 236 26.0          4          97          75   2265          18.2   77
3
## 9   14.0          8         455         225   4425          10.0   70
1
## 376 37.0          4          91          68   2025          18.2   82
3
## 388 26.0          4         156          92   2585          14.5   82
1
## 188 17.5          8         305         140   4215          13.0   76
1
## 62  21.0          4         122          86   2226          16.5   72
1
## 204 29.5          4          97          71   1825          12.2   76
2
## 154 18.0          6         250         105   3459          16.0   75
1
## 350 34.1          4          91          68   1985          16.0   81
3
## 55  35.0          4          72          69   1613          18.0   71
3
## 290 16.9          8         350         155   4360          14.9   79
1
## 377 31.0          4          91          68   1970          17.6   82
3
## 235 24.5          4         151          88   2740          16.0   77
1
## 187 27.0          4         101          83   2202          15.3   76
2
## 159 16.0          8         318         150   4498          14.5   75
1
## 116 15.0          8         350         145   4082          13.0   73
1
## 232 15.5          8         400         190   4325          12.2   77
1
## 54  31.0          4          71          65   1773          19.0   71
3
## 391 36.0          4         135          84   2370          13.0   82
1
## 207 26.5          4         140          72   2565          13.6   76
1
## 248 39.4          4          85          70   2070          18.6   78
3
## 374 24.0          4         140          92   2865          16.4   82
1
## 319 29.8          4         134          90   2711          15.5   80
3
## 262 18.1          6         258         120   3410          15.1   78
1
## 368 28.0          4         112          88   2605          19.6   82
1
## 341 25.8          4         156          92   2620          14.4   81
1
## 347 32.3          4          97          67   2065          17.8   81
3
## 343 30.0          4         135          84   2385          12.9   81
1
## 378 38.0          4         105          63   2125          14.7   82
1
## 221 33.5          4          85          70   1945          16.8   77
```

```
3
## 58  24.0         4          113         95   2278          15.5   72
3
## 106 13.0         8          360        170   4654          13.0   73
1
## 102 23.0         6          198         95   2904          16.0   73
1
## 362 25.4         6          168        116   2900          12.6   81
3
## 389 22.0         6          232        112   2835          14.7   82
1
## 314 28.0         4          151         90   2678          16.5   80
1
## 136 18.0         6          225        105   3613          16.5   74
1
## 382 34.0         4          108         70   2245          16.9   82
3
## 131 26.0         4          122         80   2451          16.5   74
1
## 107 12.0         8          350        180   4499          12.5   73
1
## 354 33.0         4          105         74   2190          14.2   81
2
## 371 31.0         4          112         85   2575          16.2   82
1
## 29   9.0         8          304        193   4732          18.5   70
1
## 222 17.5         8          305        145   3880          12.5   77
1
## 27  10.0         8          307        200   4376          15.0   70
1
## 358 32.9         4          119        100   2615          14.8   81
3
## 289 18.2         8          318        135   3830          15.2   79
1
## 344 39.1         4           79         58   1755          16.9   81
3
## 192 22.0         6          225        100   3233          15.4   76
1
## 316 24.3         4          151         90   3003          20.1   80
1
## 94  14.0         8          318        150   4237          14.5   73
1
## 147 28.0         4           90         75   2125          14.5   74
1
## 307 28.8         6          173        115   2595          11.3   79
1
## 269 27.2         4          119         97   2300          14.7   78
3
## 150 24.0         4          120         97   2489          15.0   74
3
## 165 21.0         6          231        110   3039          15.0   75
1
## 163 15.0         6          258        110   3730          19.0   75
1
## 67  17.0         8          304        150   3672          11.5   72
1
## 326 44.3         4           90         48   2085          21.7   80
2
```

```
## 304 31.8         4          85          65   2020          19.2   79
3
## 135 16.0         6         258         110   3632          18.0   74
1
## 118 29.0         4          68          49   1867          19.5   73
2
## 206 28.0         4          97          75   2155          16.4   76
3
## 253 19.2         6         231         105   3535          19.2   78
1
## 321 37.0         4         119          92   2434          15.0   80
3
## 251 19.4         8         318         140   3735          13.2   78
1
## 46  18.0         6         258         110   2962          13.5   71
1
## 148 24.0         4          90          75   2108          15.5   74
2
## 303 34.5         4         105          70   2150          14.9   79
1
## 336 35.0         4         122          88   2500          15.1   80
2
## 177 19.0         6         232          90   3211          17.0   75
1
## 329 30.0         4         146          67   3250          21.8   80
2
## 105 12.0         8         400         167   4906          12.5   73
1
## 285 20.6         6         225         110   3360          16.6   79
1
## 220 25.5         4         122          96   2300          15.5   77
1
## 178 23.0         4         115          95   2694          15.0   75
2
## 158 15.0         8         350         145   4440          14.0   75
1
## 2   15.0         8         350         165   3693          11.5   70
1
## 132 32.0         4          71          65   1836          21.0   74
3
## 349 37.7         4          89          62   2050          17.3   81
3
## 24  26.0         4         121         113   2234          12.5   70
2
## 243 21.5         4         121         110   2600          12.8   77
2
## 351 34.7         4         105          63   2215          14.9   81
1
## 255 20.2         6         200          85   2965          15.8   78
1
## 325 40.8         4          85          65   2110          19.2   80
3
## 189 16.0         8         318         150   4190          13.0   76
1
## 113 19.0         4         122          85   2310          18.5   73
1
## 312 32.1         4          98          70   2120          15.5   80
1
## 81  22.0         4         122          86   2395          16.0   72
```

```
1
## 241 30.5         4           97          78   2190           14.1   77
2
## 36  17.0         6          250         100   3329           15.5   71
1
## 88  13.0         8          350         145   3988           13.0   73
1
## 322 32.2         4          108          75   2265           15.2   80
3
## 283 22.3         4          140          88   2890           17.3   79
1
## 112 18.0         3           70          90   2124           13.5   73
3
## 195 22.5         6          232          90   3085           17.6   76
1
## 223 17.0         8          260         110   4060           19.0   77
1
## 270 30.9         4          105          75   2230           14.5   78
1
## 266 17.5         8          318         140   4080           13.7   78
1
## 31  28.0         4          140          90   2264           15.5   71
1
## 130 31.0         4           79          67   1950           19.0   74
3
## 74  13.0         8          307         130   4098           14.0   72
1
## 276 17.0         6          163         125   3140           13.6   78
2
## 392 27.0         4          151          90   2950           17.3   82
1
## 261 18.6         6          225         110   3620           18.7   78
1
## 45  13.0         8          400         175   5140           12.0   71
1
## 397 31.0         4          119          82   2720           19.4   82
1
## 93  13.0         8          351         158   4363           13.0   73
1
## 218 30.0         4          111          80   2155           14.8   77
1
## 370 34.0         4          112          88   2395           18.0   82
1
## 190 15.5         8          304         120   3962           13.9   76
1
## 250 19.9         8          260         110   3365           15.5   78
1
## 38  18.0         6          232         100   3288           15.5   71
1
## 208 20.0         4          130         102   3150           15.7   76
2
## 300 27.2         4          141          71   3190           24.8   79
2
## 272 23.2         4          156         105   2745           16.7   78
1
## 12  14.0         8          340         160   3609            8.0   70
1
## 124 20.0         6          156         122   2807           13.5   73
3
```

```
## 21  25.0         4           110          87   2672          17.5   70
2
## 215 13.0         8           302         130   3870          15.0   76
1
## 146 32.0         4            83          61   2003          19.0   74
3
## 128 19.0         6           232         100   2901          16.0   74
1
## 173 25.0         4            90          71   2223          16.5   75
2
## 30  27.0         4            97          88   2130          14.5   71
3
## 320 31.3         4           120          75   2542          17.5   80
3
## 51  28.0         4           116          90   2123          14.0   71
2
## 48  19.0         6           250         100   3282          15.0   71
1
## 313 37.2         4            86          65   2019          16.4   80
3
## 83  23.0         4           120          97   2506          14.5   72
3
## 97  13.0         8           360         175   3821          11.0   73
1
## 224 15.5         8           318         145   4140          13.7   77
1
## 268 27.5         4           134          95   2560          14.2   78
3
## 295 34.1         4            86          65   1975          15.2   79
3
## 96  12.0         8           455         225   4951          11.0   73
1
## 278 16.2         6           163         133   3410          15.8   78
2
## 239 33.5         4            98          83   2075          15.9   77
1
## 369 27.0         4           112          88   2640          18.6   82
1
## 359 31.6         4           120          74   2635          18.3   81
3
## 287 17.6         8           302         129   3725          13.4   79
1
## 121 19.0         4           121         112   2868          15.5   73
2
## 8   14.0         8           440         215   4312           8.5   70
1
## 339 27.2         4           135          84   2490          15.7   81
1
## 169 23.0         4           140          83   2639          17.0   75
1
## 98  18.0         6           225         105   3121          16.5   73
1
## 383 38.0         4            91          67   1965          15.0   82
3
## 209 13.0         8           318         150   3940          13.2   76
1
## 151 26.0         4           108          93   2391          15.5   74
3
##                                      name mpg01
```

```
## 181                         saab 99le     1
## 14             buick estate wagon (sw)     0
## 197                    chevrolet woody     1
## 308           oldsmobile omega brougham     1
## 119                         opel manta     1
## 301    oldsmobile cutlass salon brougham     1
## 231        chevrolet monte carlo landau     0
## 246                        ford fiesta     1
## 396                        ford ranger     1
## 379                     mercury lynx l     1
## 155                    mercury monarch     0
## 91            mercury marquis brougham     0
## 92            chevrolet caprice classic     0
## 258                       amc concord     0
## 199                        honda civic     1
## 385                      datsun 310 gx     1
## 352                     ford escort 4w     1
## 139           dodge coronet custom (sw)     0
## 360          peugeot 505s turbo diesel     1
## 330                honda civic 1500 gl     1
## 26                          ford f250     0
## 7                     chevrolet impala     0
## 380                   nissan stanza xe     1
## 256                 ford fairmont (man)     1
## 213                   cadillac seville     0
## 79                   peugeot 504 (sw)     0
## 82                    datsun 510 (sw)     1
## 44             ford country squire (sw)     0
## 364                      buick century     0
## 335                      mazda rx-7 gs     1
## 145                      toyota corona     1
## 32                       toyota corona     1
## 110                     chevrolet vega     0
## 265                        ford futura     0
## 333                   vokswagen rabbit     1
## 23                           saab 99e     1
## 311               toyota corolla tercel     1
## 137                    ford gran torino     0
## 361                       volvo diesel     1
## 226                  chevrolet concours     0
## 168                     toyota corolla     1
## 219                      renault 5 gtl     1
## 292        chevrolet malibu classic (sw)     0
## 70           oldsmobile delta 88 royale     0
## 73                   amc matador (sw)     0
## 77                    volvo 145e (sw)     0
## 64                    pontiac catalina     0
## 143                   volkswagen dasher     1
## 212                  mercedes-benz 280s     0
## 387    oldsmobile cutlass ciera (diesel)     1
## 296          dodge colt hatchback custom     1
## 279                 volkswagen scirocco     1
## 42                    plymouth fury iii     0
## 386              buick century limited     1
## 318                          audi 4000     1
## 225             mercury cougar brougham     0
## 16                     plymouth duster     0
## 117                  pontiac grand prix     0
## 95          chrysler new yorker brougham     0
```

```
## 264      buick regal sport coupe (turbo)     0
## 237                 ford mustang ii 2+2     1
## 87                          amc matador     0
## 40            pontiac catalina brougham     0
## 161                       buick century     0
## 242                          datsun 810     0
## 211                      toyota mark ii     0
## 394                           vw pickup     1
## 35            plymouth satellite custom     0
## 4                          amc rebel sst     0
## 13              chevrolet monte carlo     0
## 353                       ford escort 2h     1
## 245      volkswagen rabbit custom diesel     1
## 310                           vw rabbit     1
## 280                      honda accord lx     1
## 90                 dodge coronet custom     0
## 25                           amc gremlin     0
## 293 chrysler lebaron town @ country (sw)     0
## 288              mercury grand marquis     0
## 332                           subaru dl     1
## 122                  dodge dart custom     0
## 111                          datsun 610     0
## 160                            ford ltd     0
## 65                    plymouth fury iii     0
## 201                  ford granada ghia     0
## 68                     mercury marquis     0
## 153            plymouth valiant custom     0
## 86                   buick century 350     0
## 167                     ford mustang ii     0
## 138           buick century luxus (sw)     0
## 52                         peugeot 304     1
## 75                ford gran torino (sw)     0
## 180                         volvo 244dl     0
## 238                  chevrolet chevette     1
## 99                 chevrolet nova custom     0
## 216                          dodge d100     0
## 129                      chevrolet nova     0
## 214                          chevy c10     0
## 176                  volkswagen rabbit     1
## 275                          audi 5000     0
## 234           volkswagen rabbit custom     1
## 323                           mazda glc     1
## 282                    mercury zephyr 6     0
## 114                    mercury capri v6     0
## 108                         amc gremlin     0
## 309                     pontiac phoenix     1
## 156                       ford maverick     0
## 103            volkswagen super beetle     1
## 257                     plymouth volare     0
## 162          chevroelt chevelle malibu     0
## 157                    pontiac catalina     0
## 5                          ford torino     0
## 274                      datsun 200-sx     1
## 342                  chevrolet citation     1
## 345                      plymouth champ     1
## 56                volkswagen model 111     1
## 240                           subaru dl     1
## 254                    chevrolet malibu     0
## 363                   datsun 810 maxima     1
```

```
## 375                 volkswagen rabbit l      1
## 228              plymouth volare custom      0
## 49                         ford mustang     0
## 78                  volkswagen 411 (sw)     0
## 84                      dodge colt (sw)     1
## 186                          dodge colt     1
## 277                          saab 99gle     0
## 198                           vw rabbit     1
## 259               buick century special     0
## 170                          amc gremlin    0
## 338                        honda accord     1
## 20          volkswagen 1131 deluxe sedan    1
## 395                       dodge rampage     1
## 166                 chevrolet monza 2+2     0
## 53                            fiat 124b     1
## 22                          audi 100 ls     1
## 179                         peugeot 504     1
## 43                    dodge monaco (sw)     0
## 60                    volkswagen type 3     1
## 85              toyota corolla 1600 (sw)    1
## 11                  dodge challenger se     0
## 315                       ford fairmont     1
## 185                            capri ii     1
## 306               buick skylark limited     1
## 47                  chevrolet vega (sw)     0
## 328                 audi 5000s (diesel)     1
## 365               oldsmobile cutlass ls     1
## 196                  chevrolet chevette     1
## 291             ford country squire (sw)    0
## 273              oldsmobile starfire sx     1
## 297                        amc spirit dl    1
## 200                       dodge aspen se    0
## 202                   pontiac ventura sj    0
## 174                          datsun 710     1
## 286              chevrolet caprice classic  0
## 37                       ford torino 500    0
## 175                          ford pinto     0
## 144                          opel manta     1
## 340                       buick skylark     1
## 217                    honda accord cvcc    1
## 126                      plymouth duster    0
## 34                           amc gremlin    0
## 41                      ford galaxie 500    0
## 267                  chevrolet chevette     1
## 10                    amc ambassador dpl    0
## 356                       honda prelude     1
## 244                          mazda rx-4     0
## 348                      datsun 210 mpg     1
## 236              toyota corolla liftback    1
## 9                       pontiac catalina    0
## 376                   mazda glc custom l    1
## 388             chrysler lebaron medallion  1
## 188    chevrolet chevelle malibu classic    0
## 62                    ford pinto runabout   0
## 204                    volkswagen rabbit    1
## 154                       chevrolet nova    0
## 350                          mazda glc 4    1
## 55                           datsun 1200    1
## 290               buick estate wagon (sw)   0
```

```
## 377                 mazda glc custom     1
## 235             pontiac sunbird coupe     1
## 187                    renault 12tl     1
## 159              plymouth grand fury     0
## 116          chevrolet monte carlo s     0
## 232                 chrysler cordoba     0
## 54               toyota corolla 1200     1
## 391                dodge charger 2.2     1
## 207                       ford pinto     1
## 248                   datsun b210 gx     1
## 374             ford fairmont futura     1
## 319            toyota corona liftback     1
## 262                  amc concord d/l     0
## 368               chevrolet cavalier     1
## 341            dodge aries wagon (sw)     1
## 347                           subaru     1
## 343                 plymouth reliant     1
## 378            plymouth horizon miser     1
## 221             datsun f-10 hatchback     1
## 58              toyota corona hardtop     1
## 106            plymouth custom suburb     0
## 102                  plymouth duster     1
## 362                  toyota cressida     1
## 389                   ford granada l     0
## 314               chevrolet citation     1
## 136        plymouth satellite sebring     0
## 382                   toyota corolla     1
## 131                       ford pinto     1
## 107          oldsmobile vista cruiser     0
## 354                volkswagen jetta     1
## 371        pontiac j2000 se hatchback     1
## 29                          hi 1200d     0
## 222          chevrolet caprice classic     0
## 27                         chevy c20     0
## 358                      datsun 200sx     1
## 289                   dodge st. regis     0
## 344                   toyota starlet     1
## 192                 plymouth valiant     0
## 316                      amc concord     1
## 94           plymouth fury gran sedan     0
## 147                       dodge colt     1
## 307               chevrolet citation     1
## 269                       datsun 510     1
## 150                      honda civic     1
## 165                    buick skyhawk     0
## 163                      amc matador     0
## 67                amc ambassador sst     0
## 326              vw rabbit c (diesel)     1
## 304                       datsun 210     1
## 135                      amc matador     0
## 118                         fiat 128     1
## 206                   toyota corolla     1
## 253                pontiac phoenix lj     0
## 321               datsun 510 hatchback     1
## 251                   dodge diplomat     0
## 46         amc hornet sportabout (sw)     0
## 148                         fiat 128     1
## 303               plymouth horizon tc3     1
## 336                 triumph tr7 coupe     1
```

```
## 177                          amc pacer     0
## 329                   mercedes-benz 240d     1
## 105                        ford country     0
## 285                        dodge aspen 6     0
## 220                    plymouth arrow gs     1
## 178                           audi 100ls     1
## 158                    chevrolet bel air     0
## 2                     buick skylark 320     0
## 132                  toyota corolla 1200     1
## 349                        toyota tercel     1
## 24                             bmw 2002     1
## 243                            bmw 320i     0
## 351                   plymouth horizon 4     1
## 255                 ford fairmont (auto)     0
## 325                           datsun 210     1
## 189                dodge coronet brougham     0
## 113                           ford pinto     0
## 312                   chevrolet chevette     1
## 81                      ford pinto (sw)     0
## 241                    volkswagen dasher     1
## 36              chevrolet chevelle malibu     0
## 88                     chevrolet malibu     0
## 322                       toyota corolla     1
## 283                      ford fairmont 4     0
## 112                           maxda rx3     0
## 195                           amc hornet     0
## 223           oldsmobile cutlass supreme     0
## 270                           dodge omni     1
## 266                      dodge magnum xe     0
## 31                    chevrolet vega 2300     1
## 130                          datsun b210     1
## 74       chevrolet chevelle concours (sw)     0
## 276                          volvo 264gl     0
## 392                     chevrolet camaro     1
## 261                          dodge aspen     0
## 45                    pontiac safari (sw)     0
## 397                          chevy s-10     1
## 93                             ford ltd     0
## 218               buick opel isuzu deluxe     1
## 370             chevrolet cavalier 2-door     1
## 190                          amc matador     0
## 250      oldsmobile cutlass salon brougham     0
## 38                          amc matador     0
## 208                            volvo 245     0
## 300                          peugeot 504     1
## 272                     plymouth sapporo     1
## 12                    plymouth 'cuda 340     0
## 124                       toyota mark ii     0
## 21                          peugeot 504     1
## 215                            ford f108     0
## 146                           datsun 710     1
## 128                           amc hornet     0
## 173                    volkswagen dasher     1
## 30                          datsun pl510     1
## 320                            mazda 626     1
## 51                            opel 1900     1
## 48                      pontiac firebird     0
## 313                           datsun 310     1
## 83           toyouta corona mark ii (sw)     1
```

```
## 97                amc ambassador brougham     0
## 224                 dodge monaco brougham     0
## 268                         toyota corona     1
## 295                      maxda glc deluxe     1
## 96              buick electra 225 custom     0
## 278                        peugeot 604sl     0
## 239                       dodge colt m/m     1
## 369              chevrolet cavalier wagon     1
## 359                            mazda 626     1
## 287                      ford ltd landau     0
## 121                          volvo 144ea     0
## 8                      plymouth fury iii     0
## 339                     plymouth reliant     1
## 169                           ford pinto     1
## 98                      plymouth valiant     0
## 383                          honda civic     1
## 209            plymouth volare premier v8     0
## 151                               subaru     1
```

Findiings:- The exploratory analysis reveals that certain features are strongly associated with whether a car's MPG is above or below the median (mpg01). Specifically, Weight appears to be a key predictor, as heavier cars generally have lower MPG values (mpg01 = 0), while lighter cars are more likely to achieve higher MPG (mpg01 = 1). This negative relationship suggests that the weight of the vehicle impacts fuel efficiency significantly.

Horsepower is another important predictor, with cars that have higher horsepower typically falling into the lower MPG category. This indicates that more powerful engines, while boosting performance, tend to consume more fuel, reducing efficiency. Similarly, Displacement shows a strong association with mpg01, as cars with larger engine displacement generally exhibit lower MPG, further suggesting that engine size and fuel efficiency are inversely related.

Together, these findings suggest that Weight, Horsepower, and Displacement are likely the most useful features for predicting mpg01. The clear separation in these features between high and low MPG groups indicates they could serve as strong predictors in a classification model aimed at identifying whether a car will have high or low fuel efficiency.

(4).[3 points] Perform LDA on the training data in order to predict `mpg01` using the variables that seemed most associated with `mpg01` in (2). What is the test error of the model obtained?

```
Answer:
```

```
library(MASS)


lda_model <- lda(mpg01 ~ weight + horsepower + displacement, data = train_d
ata)
```

```
lda_pred <- predict(lda_model, test_data)$class
lda_error <- mean(lda_pred != test_data$mpg01)
cat("lda Error:",lda_error)
```

```
## lda Error: 0.1392405
```

(5).[3 points] Perform QDA on the training data in order to predict `mpg01` using the variables that seemed most associated with `mpg01` in (2). What is the test error of the model obtained?

```
Answer:
```

```
qda_model <- qda(mpg01 ~ weight + horsepower + displacement, data = train_d
ata)
```

```
qda_pred <- predict(qda_model, test_data)$class
qda_error <- mean(qda_pred != test_data$mpg01)
cat("qda Error:",qda_error)
```

```
## qda Error: 0.1139241
```

(6). [3 points] Perform logistic regression on the training data in order to predict `mpg01` using the variables that seemed most associated with `mpg01` in (2). What is the test error of the model obtained?

```
Answer:
```

```
logit_model <- glm(mpg01 ~ weight + horsepower + displacement, data = train
_data, family = binomial)
```

```
logit_pred <- predict(logit_model, test_data, type = "response")
logit_class <- ifelse(logit_pred > 0.5, 1, 0)
logit_error <- mean(logit_class != test_data$mpg01)

cat("Logit Error:",logit_error)
```

```
## Logit Error: 0.1518987
```

(7). [3 points] Perform naive Bayes on the training data in order to predict `mpg01` using the variables that seemed most associated with `mpg01` in (2). What is the test error of the model obtained?

```
Answer:
```

```
library(e1071)

# Perform Naive Bayes on training data
nb_model <- naiveBayes(mpg01 ~ weight + horsepower + displacement, data = t
rain_data)
nb_pred <- predict(nb_model, test_data)
nb_error <- mean(nb_pred != test_data$mpg01)
cat("NB Error:",nb_error)
```

```
## NB Error: 0.1392405
```

(8). [5 points] Perform KNN on the training data in order to predict `mpg01` using the variables that seemed most associated with `mpg01` in (2). What is the test error of the model obtained? Which value of K seems to perform the best on this data set?

```
Answer:
```

```
library(class)


train_X <- train_data[, c("weight", "horsepower", "displacement")]
test_X <- test_data[, c("weight", "horsepower", "displacement")]
train_y <- train_data$mpg01


k_values <- c(1, 3, 5, 7, 9)
knn_errors <- sapply(k_values, function(k) {
  knn_pred <- knn(train_X, test_X, train_y, k = k)
  mean(knn_pred != test_data$mpg01)
})
best_k <- k_values[which.min(knn_errors)]
knn_error <- min(knn_errors)
cat("KNN ERROR:",knn_error,"\n")
```

```
## KNN ERROR: 0.1265823
```

```
cat("Best K value:",best_k)
```

```
## Best K value: 9
```

(9).[3 points] Compare the above models, which models do you think is the best, why?

```
Answer: Compare Models:
```

The best model for predicting mpg01 in this case is QDA (Quadratic Discriminant Analysis), as it achieved the lowest test error (0.1139) among all models tested. QDA's strength lies in its ability to capture non-linear relationships, which seems to suit this dataset better than models like LDA and logistic regression, which assume linear boundaries. Although KNN also performed well with a test error of 0.1266 at K=9 QDA is more efficient and interpretable as a parametric model. Therefore, Best Mode is QDA because of its Lowest test error, indicating it most effectively captures the structure in the data and it stands out as the most effective and accurate choice for this task.