Q33

Rahul Atre

2023-11-15

Q33 [Classification]

In this problem, we develop a model to predict whether a given car gets high or low gas mileage based on the Auto data set. Start by loading the data and removing all instances with missing values.

```
auto_df = read.csv("auto.csv")
auto_df = na.omit(auto_df)
```

1. 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.

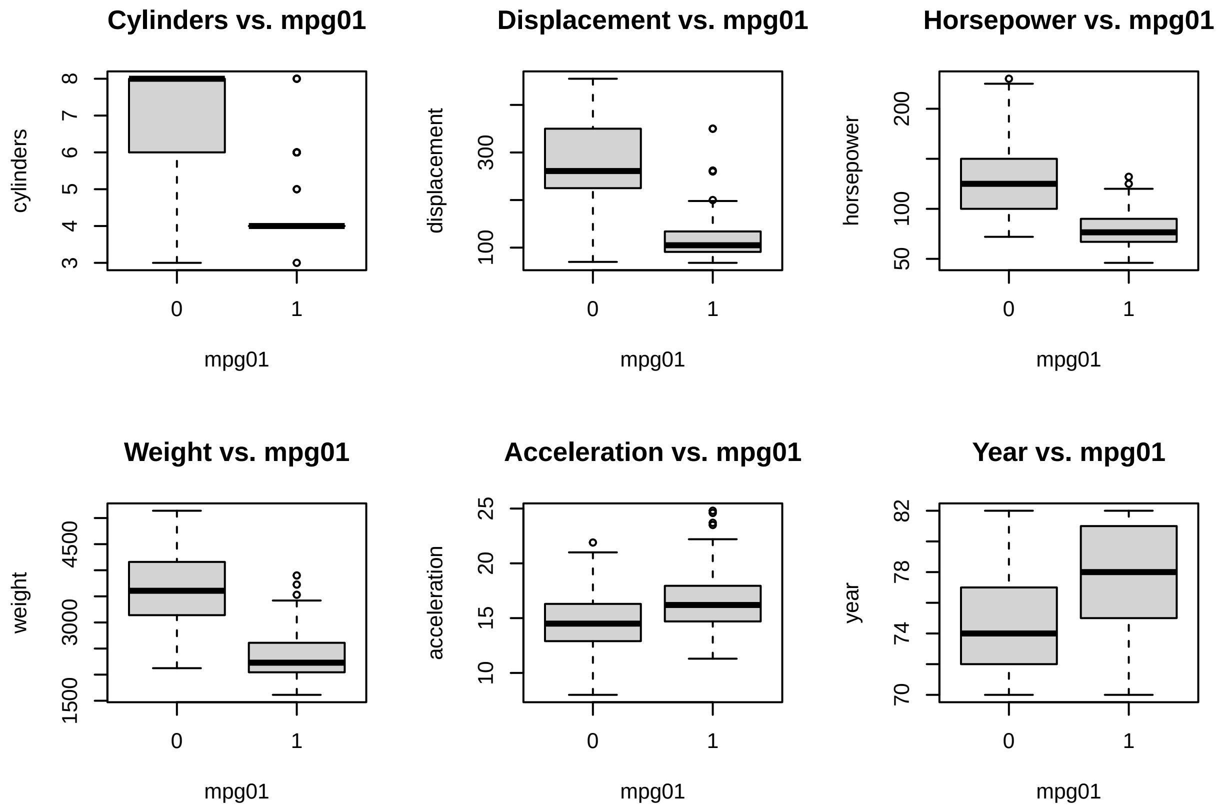
```
mpg_median <- median(auto_df$mpg)
auto_df$mpg01 <- ifelse(auto_df$mpg > mpg_median, 1, 0)
```

2. 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.

```
cor(subset(auto_df, select = -name))
```

```
##                     mpg  cylinders displacement horsepower     weight
## mpg           1.0000000 -0.7776175   -0.8051269 -0.7784268 -0.8322442
## cylinders    -0.7776175  1.0000000    0.9508233  0.8429834  0.8975273
## displacement -0.8051269  0.9508233    1.0000000  0.8972570  0.9329944
## horsepower   -0.7784268  0.8429834    0.8972570  1.0000000  0.8645377
## weight       -0.8322442  0.8975273    0.9329944  0.8645377  1.0000000
## acceleration  0.4233285 -0.5046834   -0.5438005 -0.6891955 -0.4168392
## year          0.5805410 -0.3456474   -0.3698552 -0.4163615 -0.3091199
## origin        0.5652088 -0.5689316   -0.6145351 -0.4551715 -0.5850054
## mpg01         0.8369392 -0.7591939   -0.7534766 -0.6670526 -0.7577566
##              acceleration       year     origin      mpg01
## mpg             0.4233285  0.5805410  0.5652088  0.8369392
## cylinders      -0.5046834 -0.3456474 -0.5689316 -0.7591939
## displacement   -0.5438005 -0.3698552 -0.6145351 -0.7534766
## horsepower     -0.6891955 -0.4163615 -0.4551715 -0.6670526
## weight         -0.4168392 -0.3091199 -0.5850054 -0.7577566
## acceleration    1.0000000  0.2903161  0.2127458  0.3468215
## year            0.2903161  1.0000000  0.1815277  0.4299042
## origin          0.2127458  0.1815277  1.0000000  0.5136984
## mpg01           0.3468215  0.4299042  0.5136984  1.0000000
```

```r
par(mfrow=c(2,3))
boxplot(cylinders ~ mpg01, main = "Cylinders vs. mpg01", data = auto_df)
boxplot(displacement ~ mpg01, main = "Displacement vs. mpg01", data = auto_df)
boxplot(horsepower ~ mpg01, main = "Horsepower vs. mpg01", data = auto_df)
boxplot(weight ~ mpg01, main = "Weight vs. mpg01", data = auto_df)
boxplot(acceleration ~ mpg01, main = "Acceleration vs. mpg01", data = auto_df)
boxplot(year ~ mpg01, main = "Year vs. mpg01", data = auto_df)
```



As we can see from the above boxplots and the correlation matrix, the features that seem the most likely in predicting mpg01 are: cylinders, displacement, horsepower, and weight. It is important to note from the correlation matrix that all of these predictors are negatively correlated to mpg01.

3. Split the data into a training set and a test set.

```r
set.seed(144)

training_set = auto_df[sample(nrow(auto_df), 0.75 * nrow(auto_df)), ]
test_set = auto_df[setdiff(1:nrow(auto_df), rownames(training_set)), ]
```

6. 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?

```r
set.seed(144)

training_set = training_set[, c("mpg01", "cylinders", "displacement", "horsepower", "weight")]
```

```
test_set = test_set[, c("mpg01", "cylinders", "displacement", "horsepower", "weight")]
logistic_model <- glm(mpg01 ~ .,data = test_set, family = binomial)

accuracy = mean((predict(logistic_model, test_set, type = "response") > 0.5) ==  test_set$mpg01)

1 - accuracy #Test error rate
```

```
## [1] 0.08737864
```

The test error for the model is about 11.76%.

7. Perform kNN on the training data, with several values of k, in order to predict mpg01. Use only the variables that seemed most associated with mpg01 in 2. What test errors do you obtain? Which value of k seems to perform the best on this data set?

```
set.seed(144)

k_values = c(1, 2, 3, 4, 5, 10, 20, 50, 100)

test_errors = matrix(0, nrow = length(k_values), ncol = 2)
predictors = c("cylinders", "displacement", "horsepower", "weight")

training_pred = training_set[, predictors]
test_pred = test_set[, predictors]

for (i in 1:length(k_values)) {
knn_model = knn(train = training_pred, test = test_pred, cl = training_set$mpg01, k = k_values[i])
test_errors[i, ] = c(k_values[i], mean(knn_model != test_set$mpg01))
}

test_errors
```

```
##       [,1]       [,2]
##  [1,]    1 0.04854369
##  [2,]    2 0.13592233
##  [3,]    3 0.08737864
##  [4,]    4 0.06796117
##  [5,]    5 0.08737864
##  [6,]   10 0.11650485
##  [7,]   20 0.11650485
##  [8,]   50 0.10679612
##  [9,]  100 0.13592233
```

The values of k that perform the best is k = 4, with a test error of 0.06796117. Although k=1 has the lowest, it would cause the data to be underfit. k=4 is the "sweet-spot" in terms of fitting the data accurately with any bias.