HW3_P2_RMD

Naveen Nagarajan

4/4/2021

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

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

```
summary(auto.ds)
```

```
##       mpg        cylinders  displacement     horsepower        weight
##  Min.   : 9.00   3:  4     Min.   : 68.0   Min.   : 46.0   Min.   :1613
##  1st Qu.:17.00   4:199     1st Qu.:105.0   1st Qu.: 75.0   1st Qu.:2225
##  Median :22.75   5:  3     Median :151.0   Median : 93.5   Median :2804
##  Mean   :23.45   6: 83     Mean   :194.4   Mean   :104.5   Mean   :2978
##  3rd Qu.:29.00   8:103     3rd Qu.:275.8   3rd Qu.:126.0   3rd Qu.:3615
##  Max.   :46.60             Max.   :455.0   Max.   :230.0   Max.   :5140
##
##   acceleration        year       origin                  name     mpg01
##  Min.   : 8.00   Min.   :70.00   1:245   amc matador       :  5   0:196
##  1st Qu.:13.78   1st Qu.:73.00   2: 68   ford pinto        :  5   1:196
##  Median :15.50   Median :76.00   3: 79   toyota corolla    :  5
##  Mean   :15.54   Mean   :75.98           amc gremlin       :  4
##  3rd Qu.:17.02   3rd Qu.:79.00           amc hornet        :  4
##  Max.   :24.80   Max.   :82.00           chevrolet chevette:  4
##                                          (Other)           :365
```

```
auto.ds %>%
    tibble %>%
    head
```

```
## # A tibble: 6 x 10
##     mpg cylinders displacement horsepower weight acceleration  year origin name
##   <dbl> <fct>            <dbl>      <int>  <int>        <dbl> <int> <fct>  <fct>
## 1    18 8                  307        130   3504         12      70 1      chev~
## 2    15 8                  350        165   3693         11.5    70 1      buic~
## 3    18 8                  318        150   3436         11      70 1      plym~
## 4    16 8                  304        150   3433         12      70 1      amc ~
## 5    17 8                  302        140   3449         10.5    70 1      ford~
```

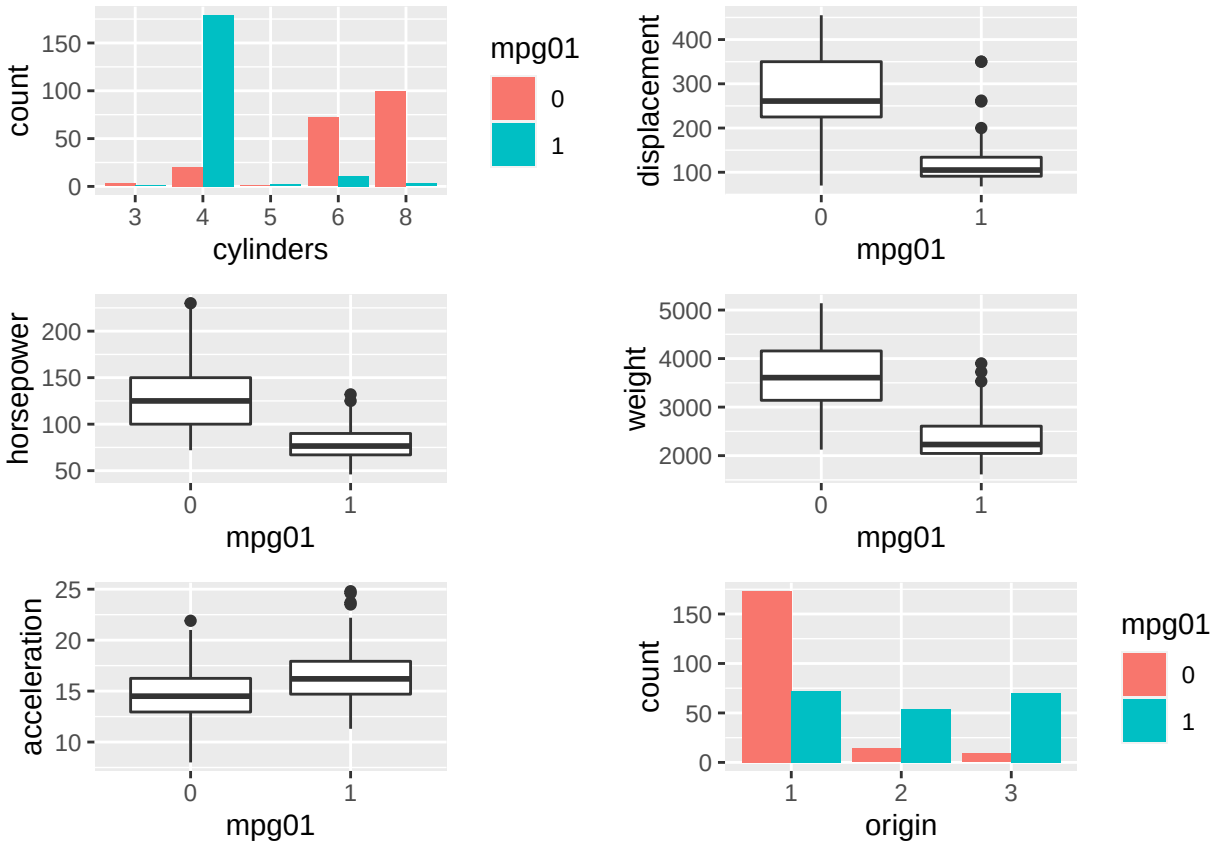
```
## 6    15 8                  429        198   4341         10      70 1      ford~
## # ... with 1 more variable: mpg01 <fct>
```

Median mpg is 22.75

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

You can also embed plots, for example:

```
egg::ggarrange(auto.ds %>%
    ggplot(mapping = aes(x = cylinders, fill = mpg01)) + geom_bar(position = "dodge"),
    auto.ds %>%
        ggplot(mapping = aes(y = displacement, x = mpg01)) + geom_boxplot(), auto.ds %>%
        ggplot(mapping = aes(y = horsepower, x = mpg01)) + geom_boxplot(), auto.ds %>%
        ggplot(mapping = aes(y = weight, x = mpg01)) + geom_boxplot(), auto.ds %>%
        ggplot(mapping = aes(y = acceleration, x = mpg01)) + geom_boxplot(), auto.ds %>%
        ggplot(mapping = aes(x = origin, fill = mpg01)) + geom_bar(position = "dodge"))
```



From the chart displacement, horsepower, weight, cylinders, origin is closely correlated with the mpg01, acceleration doesn't as the decision boundary is overlaps.

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

```
auto.ds = auto.ds %>%
    mutate(ID = row_number())
# Create training set
auto.ds.train <- auto.ds %>%
    sample_frac(0.7)
# Create test set
auto.ds.test <- anti_join(auto.ds, auto.ds.train, by = "ID")

auto.ds.train <- auto.ds.train %>%
    dplyr::select(mpg:mpg01)
auto.ds.test <- auto.ds.test %>%
    dplyr::select(mpg:mpg01)
auto.ds.train$cylinders <- as.numeric(auto.ds.train$cylinders)
auto.ds.test$cylinders <- as.numeric(auto.ds.test$cylinders)

nrow(auto.ds.train)
```

```
## [1] 274
```

```
nrow(auto.ds.test)
```

```
## [1] 118
```

d) Perform LDA on the training data in order to predict mpg01 using the variables that seemed most associated with mpg01 in b). What is the test error of the model obtained?

Error rate and confusion matrix

```
##   Sensitivity Specificity Error rate
## 1      0.9375   0.8714286  0.1016949
```

```
##
##      0  1
##   0 61  9
##   1  3 45
```

e) Perform QDA on the training data in order to predict mpg01 using the variables that seemed most associated with mpg01 in b). What is the test error of the model obtained?

Error rate and confusion matrix

```
##   Sensitivity Specificity Error rate
## 1   0.9583333   0.8714286 0.09322034
```

```
##
##      0  1
##   0 61  9
##   1  2 46
```

f) Perform logistic regression on the training data in order to predict mpg01 using the variables that seemed most associated with mpg01 in b). What is the test error of the model obtained?

Error rate and confusion matrix

```
##   Sensitivity Specificity Error rate
## 1      0.9375   0.8714286  0.1016949
```

```
##
##      0  1
##   0 61  9
##   1  3 45
```

g) 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 (b). What test errors do you obtain? Which value of K seems to perform the best on this data set?

Error rate and confusion matrix for Knn=1

```
##   Sensitivity Specificity Error rate
## 1       0.875   0.8714286  0.1271186
```

```
##
##      0  1
##   0 61  9
##   1  6 42
```

Error rate and confusion matrix for Knn=2

```
##   Sensitivity Specificity Error rate
## 1   0.8333333   0.8428571  0.1610169
```

```
##
##      0  1
##   0 59 11
##   1  8 40
```

Error rate and confusion matrix for Knn=3

```
##   Sensitivity Specificity Error rate
## 1       0.875   0.8857143  0.1186441
```

```
##
##      0  1
##   0 62  8
##   1  6 42
```

Error rate and confusion matrix for Knn=4

```
##   Sensitivity Specificity Error rate
## 1       0.875   0.8857143  0.1186441
```

```
##
##       0  1
##   0 62  8
##   1  6 42
```

Knn with k=3 has less error rate compared to others