Assignment33

John Bute

2024-11-15

R Markdown

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 be loading the data and removing all instances with missing values.

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.

```
Auto <- read.csv("C:/Users/johnb/Desktop/Machine Learning/data/Auto.csv")
Auto <- na.omit(Auto)
median_mpg <- median(Auto$mpg)
Auto$mpg01 <- ifelse(Auto$mpg > median_mpg, 1, 0)

colnames(Auto)
```

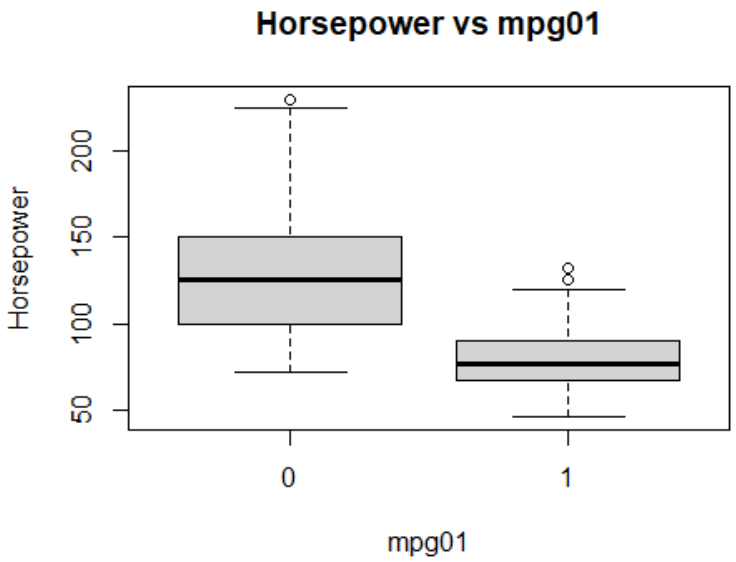
```
##  [1] "mpg"          "cylinders"    "displacement" "horsepower"   "weight"
##  [6] "acceleration" "year"         "origin"       "name"         "mpg01"
```

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.

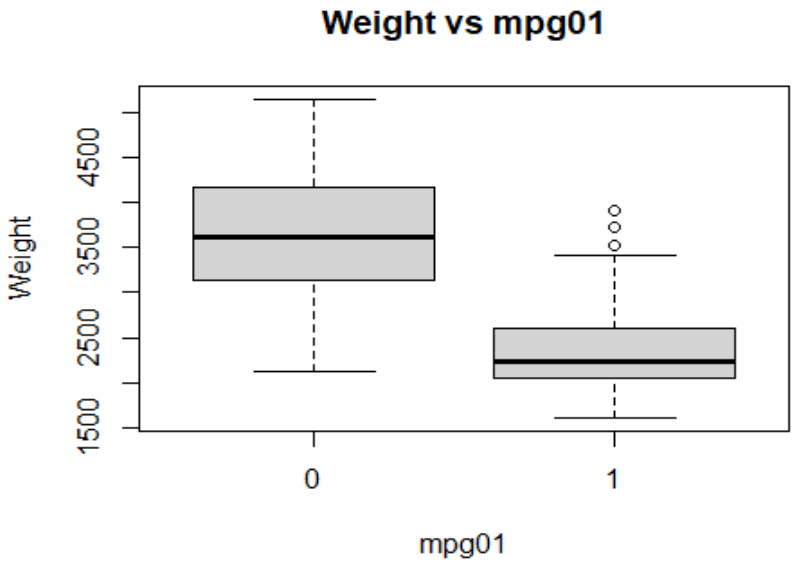
```
library(ggplot2)
library(GGally)
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

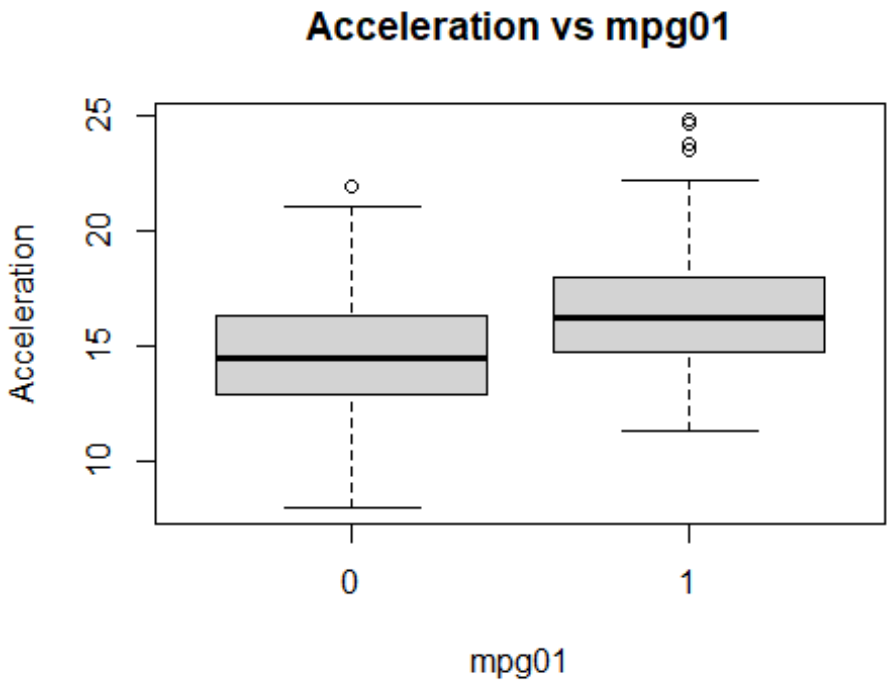
```
boxplot(Auto$horsepower ~ Auto$mpg01, main = "Horsepower vs mpg01", xlab =
"mpg01", ylab = "Horsepower")
```



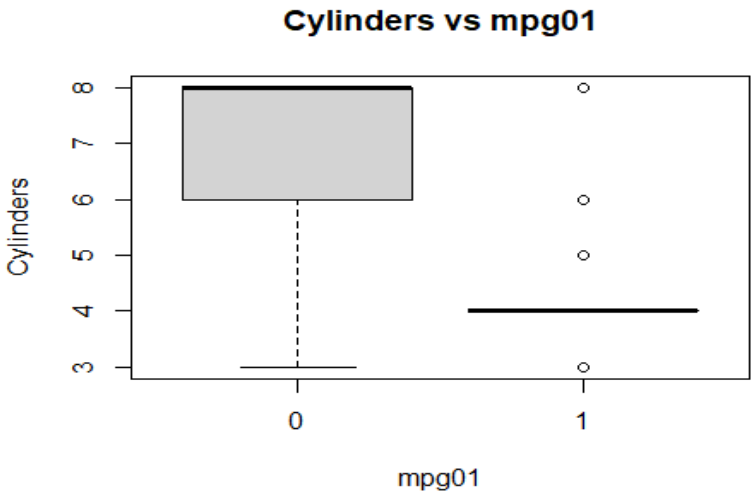

```
boxplot(Auto$weight ~ Auto$mpg01, main = "Weight vs mpg01", xlab = "mpg01",
ylab = "Weight")
```



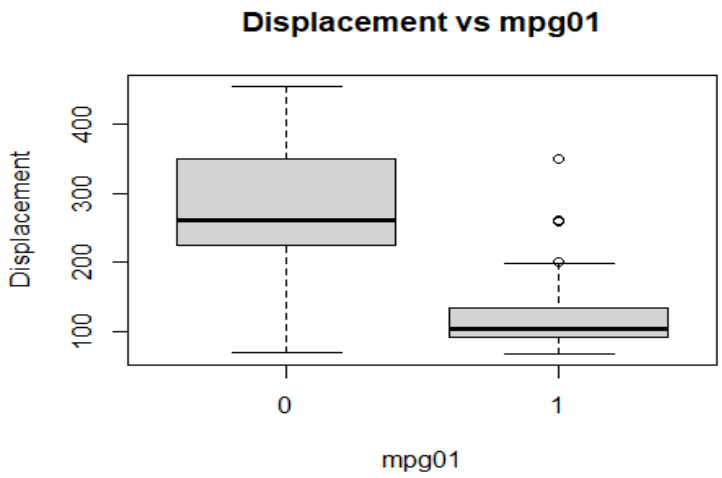

```
boxplot(Auto$acceleration ~ Auto$mpg01, main = "Acceleration vs mpg01", xlab
= "mpg01", ylab = "Acceleration")
```



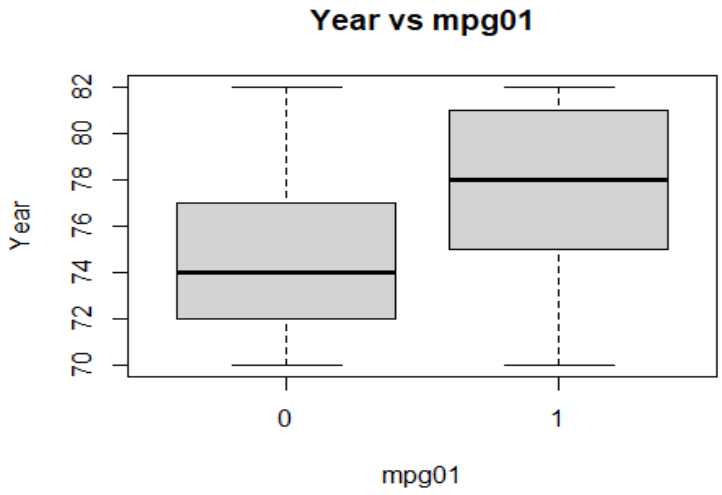

```r
boxplot(Auto$cylinders ~ Auto$mpg01, main = "Cylinders vs mpg01", xlab =
"mpg01", ylab = "Cylinders")
```




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

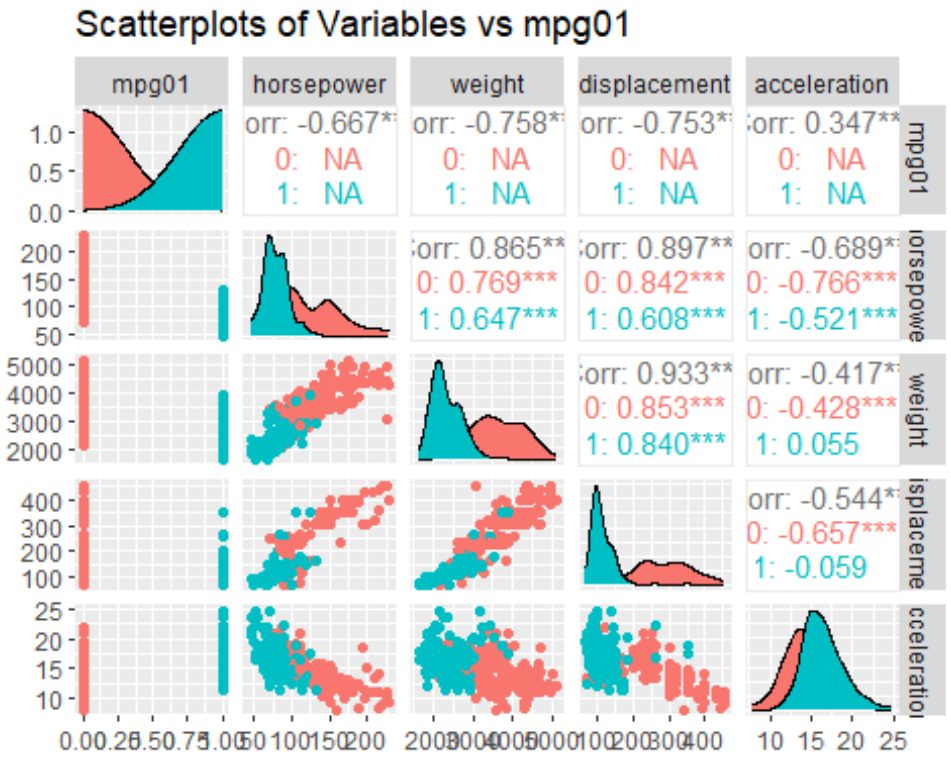



```
boxplot(Auto$year ~ Auto$mpg01, main = "Year vs mpg01", xlab = "mpg01", ylab
= "Year")
```




```
ggpairs(Auto, columns = c("mpg01", "horsepower", "weight", "displacement",
"acceleration"),
        mapping = aes(color = as.factor(mpg01)),
        title = "Scatterplots of Variables vs mpg01")
```

```
## Warning in cor(x, y): the standard deviation is zero
## Warning in cor(x, y): the standard deviation is zero
## Warning in cor(x, y): the standard deviation is zero
## Warning in cor(x, y): the standard deviation is zero
## Warning in cor(x, y): the standard deviation is zero
## Warning in cor(x, y): the standard deviation is zero
## Warning in cor(x, y): the standard deviation is zero
## Warning in cor(x, y): the standard deviation is zero
```



It seems that horsepower, weight, displacement negatively correlate with mpg01. Furthermore, acording to boxplots Displacement vs mpg01 and cylinders vs mpg01, it seems to also correlate, as less cylinders mean a higher mpg (higher than median) and a lower displacement means a higher mpg too. Finally, the only slightly positive correlation we see so far is acceleration. Finally, year seems to positively impact our mpg, as the more recent the make, the better miles per gallon it has.

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

```
set.seed(1)
index <- sample(1:nrow(Auto), 0.7 * nrow(Auto))
train_data <- Auto[index, ]
test_data <- Auto[-index, ]
```

4. 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? (optional)

5. 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? (optional)

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
model <- glm(mpg01 ~ horsepower + weight + displacement + cylinders + year,
data = train_data, family = binomial)

summary(model)

##
## Call:
## glm(formula = mpg01 ~ horsepower + weight + displacement + cylinders +
##     year, family = binomial, data = train_data)
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -13.990747   5.524488  -2.532 0.011325 *
## horsepower    -0.039348   0.019693  -1.998 0.045708 *
## weight        -0.003955   0.001108  -3.570 0.000357 ***
## displacement  -0.009207   0.011764  -0.783 0.433836
## cylinders      0.155724   0.475377   0.328 0.743230
## year           0.390736   0.084575   4.620 3.84e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 379.79  on 273  degrees of freedom
## Residual deviance: 114.05  on 268  degrees of freedom
## AIC: 126.05
##
## Number of Fisher Scoring iterations: 8
```

As we see, horsepower, is very close to the confidence interval, but still significant, along with weight and year.

```r
model <- glm(mpg01 ~ weight + year + horsepower, data = train_data, family =
binomial)
logisitc_predictions <- predict(model, test_data, type = 'response')

logisitc_labels <- ifelse(logisitc_predictions > 0.5, 1, 0)

error <- mean(logisitc_labels != test_data$mpg01)

cat("The test error for Logistic Regression is:", error, "\n")

## The test error for Logistic Regression is: 0.1016949

print(mean(logisitc_labels == test_data$mpg01))

## [1] 0.8983051

confusion_matrix <- table(Predicted = logisitc_labels, Actual =
test_data$mpg01)
```

```
print(confusion_matrix)
```

```
##           Actual
## Predicted  0  1
##         0 53  4
##         1  8 53
```

We have an approximately 90% accuracy, with an error rate of 10%.Furthermore, we notice that our model does a pretty good job at predicting whether mpg is above the median or below it based on several factors. However, a good thing to note is that in future uses, year is a continuous numerical feature. As a result, the model might assume a linearity between this and mpg01 (a categorical feature) which does not make sense. For example, I could say that the more recent models have better gas mileage, but a car from 1970 vs 1980, in terms of car technology, might be so different that year just becomes a predictor for technological advancements for cylinders, horsepower, engines, etc. Also, converting year to a categorical variable is difficult, as we need to increase the model complexity as each year will be treated as a category. For the future, if interpretability is key, then year is not the best choice for inclusion (unless its exclusion severely impacts the model).

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?

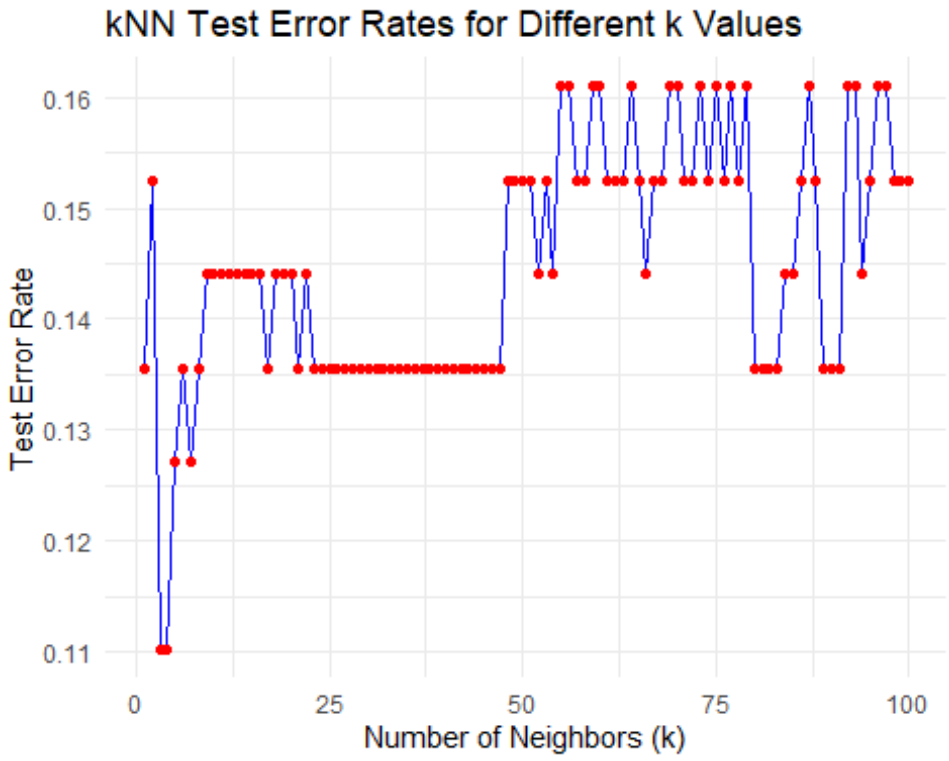
```
library(class)

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

k_values <- 1:100
knn_errors <- sapply(k_values, function(k) {
  knn_pred <- knn(train_X, test_X, train_y, k = k)
  mean(knn_pred != test_y)
})

knn_results <- data.frame(k = k_values, Error = knn_errors)

ggplot(knn_results, aes(x = k, y = Error)) +
  geom_line(color = "blue") +
  geom_point(color = "red") +
  labs(title = "kNN Test Error Rates for Different k Values",
       x = "Number of Neighbors (k)",
       y = "Test Error Rate") +
  theme_minimal()
```




```
min_error <- min(knn_errors)
best_k <- k_values[which.min(knn_errors)]
cat("The minimum error is", min_error, "and it occurs at k =", best_k, "\n")

## The minimum error is 0.1101695 and it occurs at k = 3
```

As we see, the best K is 3, with an error slightly more than our logisitc regression model, with 11% error rate.