STATS 101C - Statistical Models and Data Mining - Homework 2

Darren Tsang, Discussion 4B

Produced on Saturday, Oct. 24 2020 @ 02:42:19 AM

Question 1 (Exercise 4 from Section 4.7)

Part A

On average, you will only use 10% of the available observations to make the prediction.

Part B

On average, you will only use $(10\%)^2$ of the available observations to make the prediction.

Part C

On average, you will only use $(10\%)^{100}$ (which is very, very small) of the available observations to make the prediction.

Part D

As we can see from (a) - (c), the percentage of available observations to make the prediction decreases quite fast, which is a huge drawback for models like KNN.

Part E

We are given the fact that we want to always include 10% of our data. We can think about this wanting the area to be .1 when $p = 2$, the volume to be .1 when $p = 3$, and so forth. We can develop a formula for the length of each side l as a function of p by the following:

$$l = (.1)^{1/p}$$

When $p = 1, l = .1$. When $p = 2, l = .1^{1/2} \approx .3162278$. When $p = 100, l = .1^{1/100} \approx .9772372$.

We notice that as p gets larger, the length also gets larger; this means the cube must get bigger and bigger to make sure that 10% of the training data is always included.

Question 2 (Exercise 8 from Section 4.7)

Since we are given the average error rate using 1-nearest neighbor is 18% and the training error rate for KNN is always 0%, we can naturally derive the fact that the testing error for 1-nearest neighbor is 36%. We are also given that the testing error using logistic regression is 30%. Since the testing error is higher when using 1-nearest neighbors, I would go with logistic regression for classification of new observations.

Question 3 (Exercise 11 from Section 4.7)

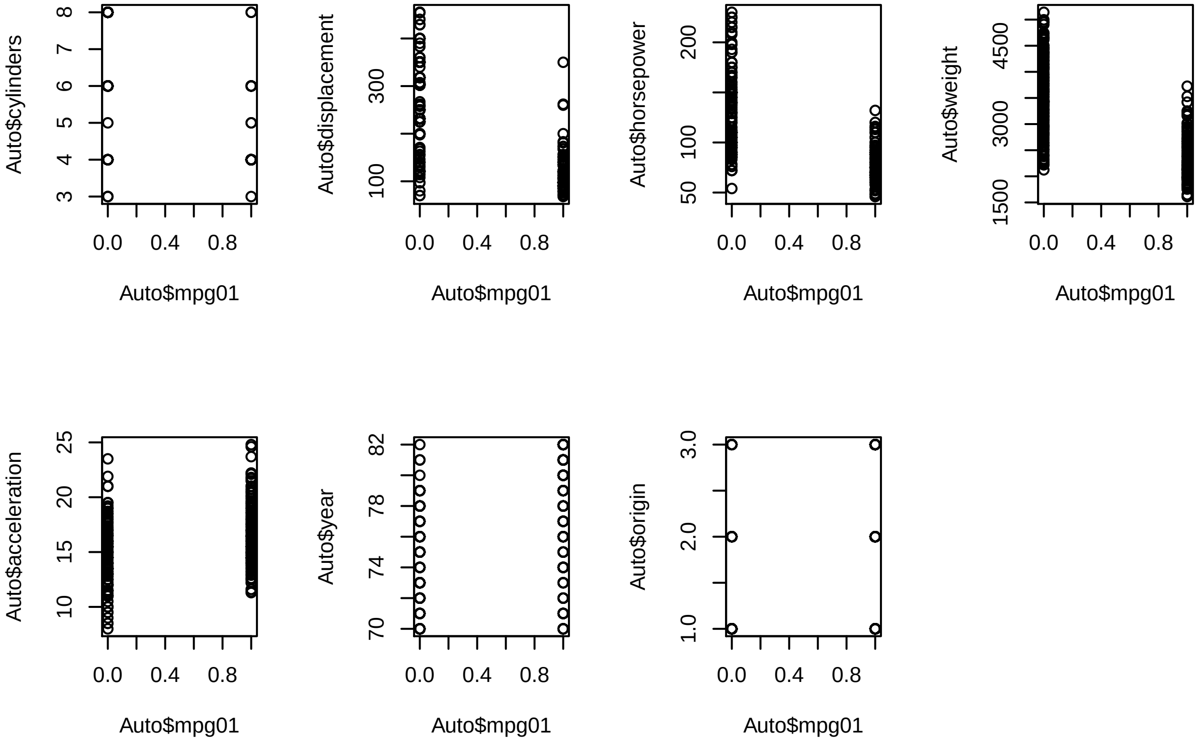
Part A

```
library(ISLR)
data(Auto)
mpg01 <- ifelse(Auto$mpg > mean(Auto$mpg), 1, 0)
Auto <- cbind(Auto, mpg01)
head(Auto)
```

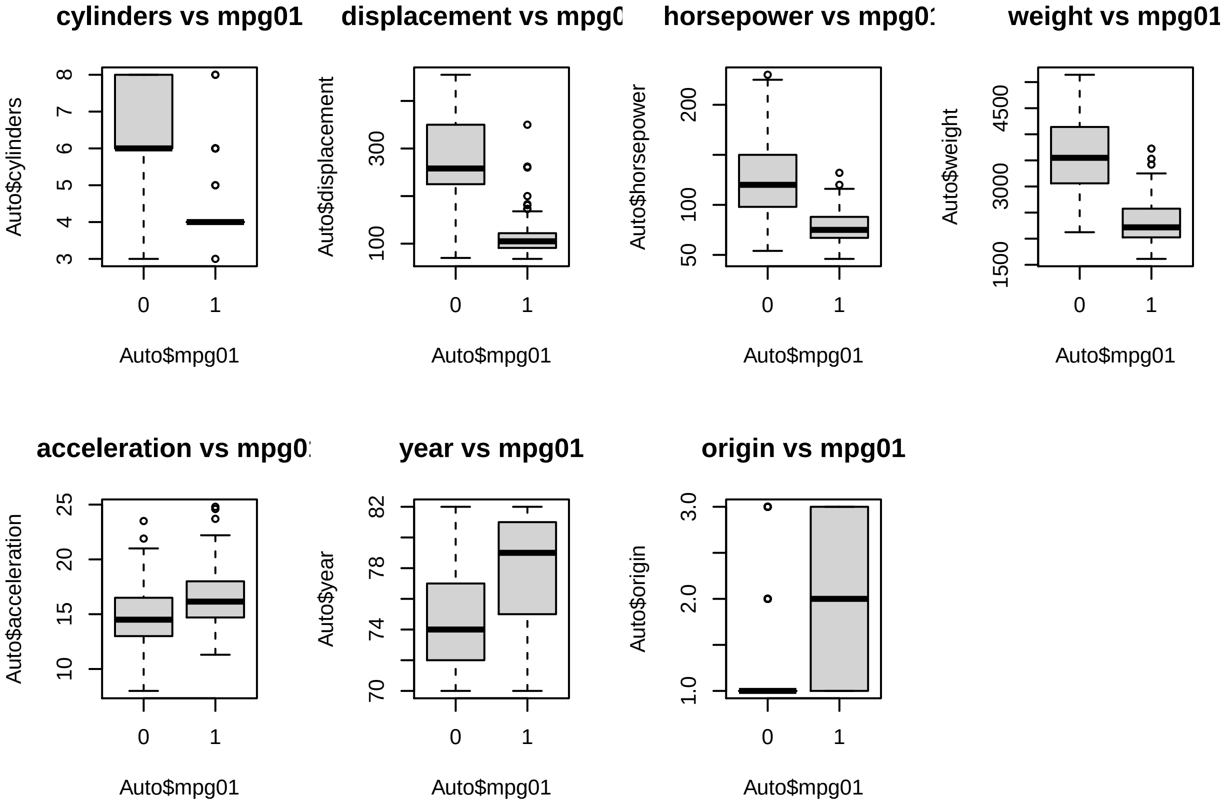
```
##   mpg cylinders displacement horsepower weight acceleration year origin
## 1  18         8          307        130   3504         12.0   70      1
## 2  15         8          350        165   3693         11.5   70      1
## 3  18         8          318        150   3436         11.0   70      1
## 4  16         8          304        150   3433         12.0   70      1
## 5  17         8          302        140   3449         10.5   70      1
## 6  15         8          429        198   4341         10.0   70      1
##                        name mpg01
## 1 chevrolet chevelle malibu     0
## 2         buick skylark 320     0
## 3        plymouth satellite     0
## 4             amc rebel sst     0
## 5               ford torino     0
## 6          ford galaxie 500     0
```

Part B

```r
par(mfrow=c(2,4))
plot(Auto$mpg01, Auto$cylinders)
plot(Auto$mpg01, Auto$displacement)
plot(Auto$mpg01, Auto$horsepower)
plot(Auto$mpg01, Auto$weight)
plot(Auto$mpg01, Auto$acceleration)
plot(Auto$mpg01, Auto$year)
plot(Auto$mpg01, Auto$origin)
```



```r
par(mfrow=c(2,4))
boxplot(Auto$cylinders ~ Auto$mpg01, main = "cylinders vs mpg01")
boxplot(Auto$displacement ~ Auto$mpg01, main = "displacement vs mpg01")
boxplot(Auto$horsepower ~ Auto$mpg01, main = "horsepower vs mpg01")
boxplot(Auto$weight ~ Auto$mpg01, main = "weight vs mpg01")
boxplot(Auto$acceleration ~ Auto$mpg01, main = "acceleration vs mpg01")
boxplot(Auto$year ~ Auto$mpg01, main = "year vs mpg01")
boxplot(Auto$origin ~ Auto$mpg01, main = "origin vs mpg01")
```



As we can see from the plots above, *cylinders*, *displacement*, *horsepower*, *weight*, and *origin* seem most likely to be useful in predicting *mpg01*.

Part C

```
set.seed(2021)
i <- 1:dim(Auto)[1]
count <- round(dim(Auto)[1] * .7)
samp <- sample(i, count, replace = FALSE)
auto_train <- Auto[samp, ]
auto_test <- Auto[-samp, ]
head(auto_train)
```

```
##      mpg cylinders displacement horsepower weight acceleration year origin
## 396 28.0         4          120         79   2625         18.6   82      1
## 168 29.0         4           97         75   2171         16.0   75      3
## 233 16.0         8          351        149   4335         14.5   77      1
## 71  13.0         8          400        190   4422         12.5   72      1
## 194 24.0         6          200         81   3012         17.6   76      1
## 253 19.2         6          231        105   3535         19.2   78      1
##                         name mpg01
## 396              ford ranger     1
## 168           toyota corolla     1
## 233         ford thunderbird     0
## 71  chrysler newport royal     0
## 194            ford maverick     1
## 253       pontiac phoenix lj     0
```

```
head(auto_test)
```

```
##    mpg cylinders displacement horsepower weight acceleration year origin
## 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
## 10  15         8          390        190   3850          8.5   70      1
## 11  15         8          383        170   3563         10.0   70      1
## 12  14         8          340        160   3609          8.0   70      1
##                   name mpg01
## 2    buick skylark 320     0
## 3   plymouth satellite     0
## 4        amc rebel sst     0
## 10  amc ambassador dpl     0
## 11 dodge challenger se     0
## 12  plymouth 'cuda 340     0
```

Part D

```
library(MASS)

lda_model <- lda(mpg01 ~ cylinders +
                   displacement +
                   horsepower +
                   weight +
                   origin,
                 data = auto_train)
lda_predictions <- predict(lda_model, auto_test[, -length(auto_test)])

lda_threshold <- .5
lda_predicted_above <- (lda_predictions$posterior[, '1'] > lda_threshold)
lda_actual <- (auto_test[, "mpg01"] == 1)
table('Actual' = lda_actual, "Predicted" = lda_predicted_above)
```

```
##        Predicted
## Actual  FALSE TRUE
##   FALSE    44   10
##   TRUE      5   59
```

```
mean(lda_actual != lda_predicted_above)
```

```
## [1] 0.1271186
```

The testing error is 0.1271186.

Part E

```
qda_model <- qda(mpg01 ~ cylinders +
                   displacement +
                   horsepower +
                   weight +
                   origin,
                 data = auto_train)
qda_predictions <- predict(qda_model, auto_test[, -length(auto_test)])

qda_threshold <- .5
qda_predicted_above <- qda_predictions$posterior[, '1'] > qda_threshold
qda_actual <- (auto_test[, "mpg01"] == 1)
table('Actual' = qda_actual, "Predicted" = qda_predicted_above)
```

```
##        Predicted
## Actual  FALSE TRUE
##   FALSE    45    9
##   TRUE      5   59
```

```
mean(qda_actual != qda_predicted_above)
```

```
## [1] 0.1186441
```

The testing error is 0.1186441.

Part F

```r
glm_model <- glm(mpg01 ~ cylinders +
                   displacement +
                   horsepower +
                   weight +
                   origin,
                 data = auto_train, family = "binomial")

glm_predictions <- predict(glm_model, auto_test[, -length(auto_test)], type = 'response')

glm_threshold <- .5
glm_predicted_above <- glm_predictions > glm_threshold
glm_actual <- (auto_test[, "mpg01"] == 1)
table('Actual' = glm_actual, "Predicted" = glm_predicted_above)
```

```
##        Predicted
## Actual  FALSE TRUE
##   FALSE    45    9
##   TRUE      8   56
```

```r
mean(glm_actual != glm_predicted_above)
```

```
## [1] 0.1440678
```

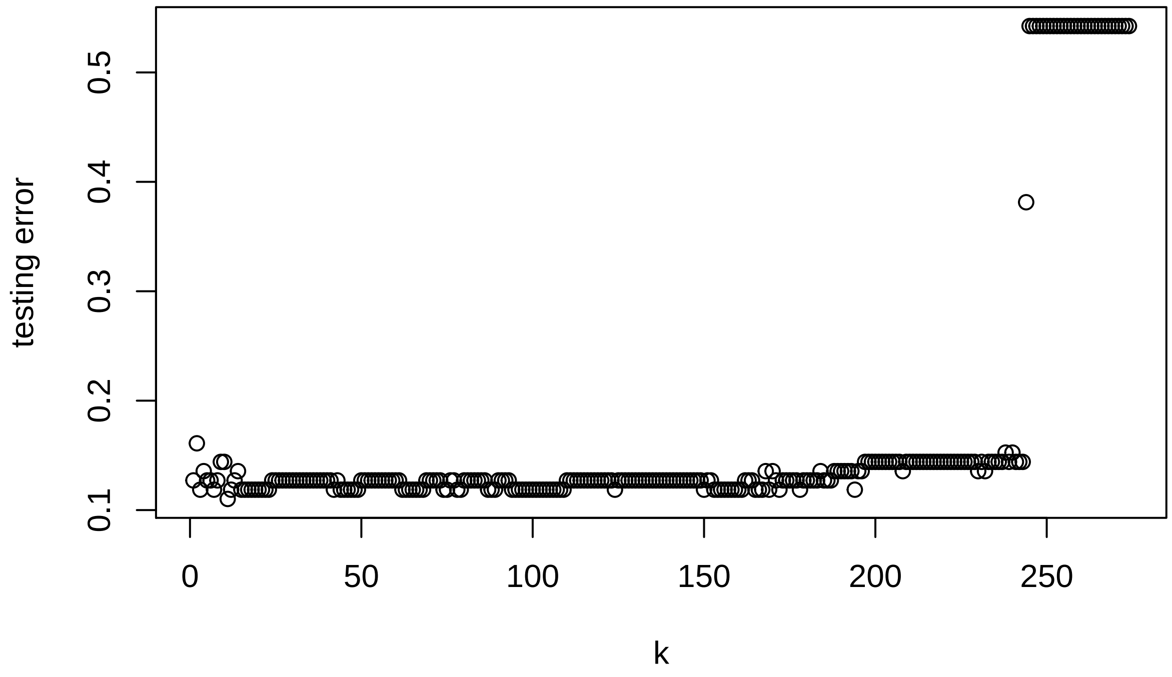
The testing error is 0.1440678.

Part G

```
library(class)

errors_knn <- rep(NA, count)
for (i in seq(1, count)){
  temp <- knn(auto_train[, c(2, 3, 4, 5, 8)],
              auto_test[, c(2, 3, 4, 5, 8)],
              auto_train$mpg01,
              k = i)
  errors_knn[i] <- mean(auto_test$mpg01 != temp)
}

plot(seq(1:count), errors_knn, xlab = "k", ylab="testing error")
```



```
min(errors_knn)
```

```
## [1] 0.1101695
```

```
which(errors_knn == min(errors_knn))
```

```
## [1] 11
```

The lowest testing error, which is 0.1101695, is achieved when K is 11.