Homework of Dataminning, CH4

Zexian Wang

2017-03-12

Q3

If we subtitute 4.11:

$f_k(x) = \frac{1}{\sqrt{2\pi}\sigma_k} exp(-\frac{1}{2\sigma_k^2}(x-\mu_k)^2)$

into 4.10, then we get:

$p_k(x) = \frac{\pi_k \frac{1}{\sqrt{2\pi}\sigma_k} exp(-\frac{1}{2\sigma_k^2}(x-\mu_k)^2)}{\sum_{i=l}^{K} \pi_l \frac{1}{\sqrt{2\pi}\sigma_l} exp(-\frac{1}{2\sigma_l^2}(x-\mu_l)^2)}$

Finding the largest $p_k(x)$ is equivalent to finding the largest $\pi_k \frac{1}{\sqrt{2\pi}\sigma_k} exp(-\frac{1}{2\sigma_k^2}(x-\mu_k)^2)$.

Taking log of above expression, we get:

$log(\pi_k) - log(\sigma_k) - \frac{1}{2\sigma_k^2}(x-\mu_k)^2$

which can be written as:

$log(\pi_k) - log(\sigma_k) - \frac{1}{2\sigma_k^2}(x^2 - 2x\mu_k + \mu_k^2)$

or:

$-\frac{1}{2\sigma_k^2}x^2 + \frac{\mu_k}{\sigma_k^2}x - \frac{\mu_k^2}{2\sigma_k^2} - log(\sigma_k) + log(\pi_k)$

This expression is quadratic.

Q5

(a)

QDA may perform better on the training set because it may overfit. LDA may perform better than QDA on the test set.

(b)

QDA may perform better both on the training set and the test set.

(c)

The prediction accuracy of QDA may improve because it can learn more patterns in the data.

(d)

False. The sample points in training set is not the whole population, QDA may learn some special patterns only appearing in the training set, which may cause wrong prediction in test set.

Q11

```r
library(ISLR)
Auto <- Auto
```

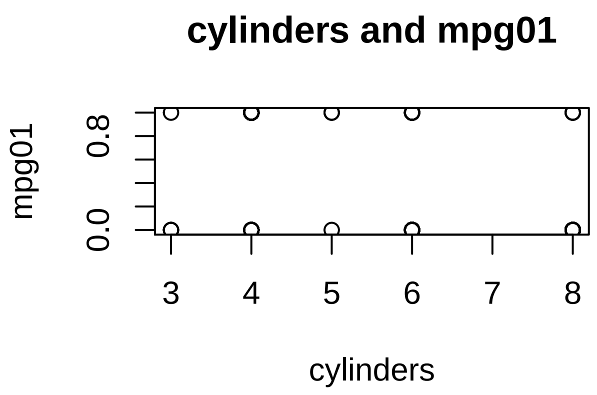
(a)

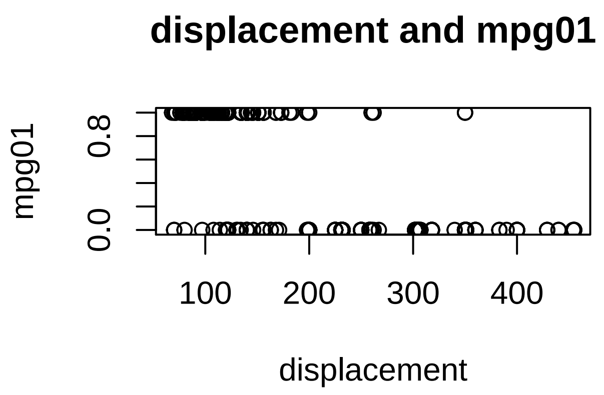
```r
Q11 <- data.frame(mpg01 = rep(0,nrow(Auto)), Auto)
Q11$mpg01[Auto$mpg > median(Auto$mpg)] <- 1
Q11$mpg <- NULL
```

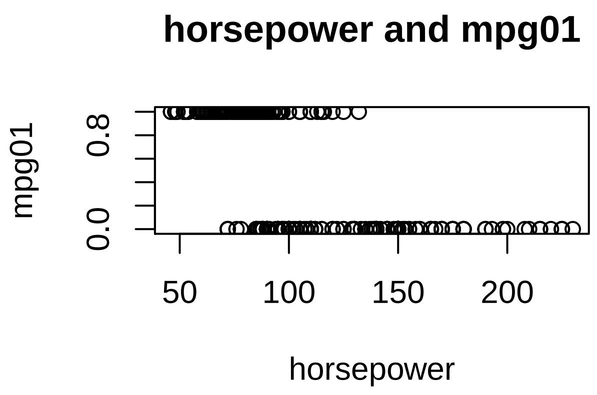
Of course we can't use mpg as a predictor because mpg01 is generalized from it.

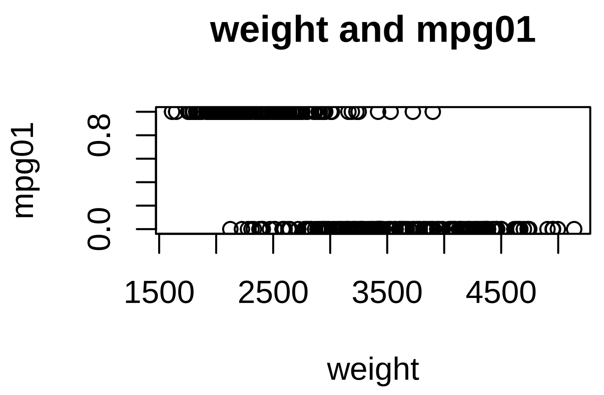
(b)

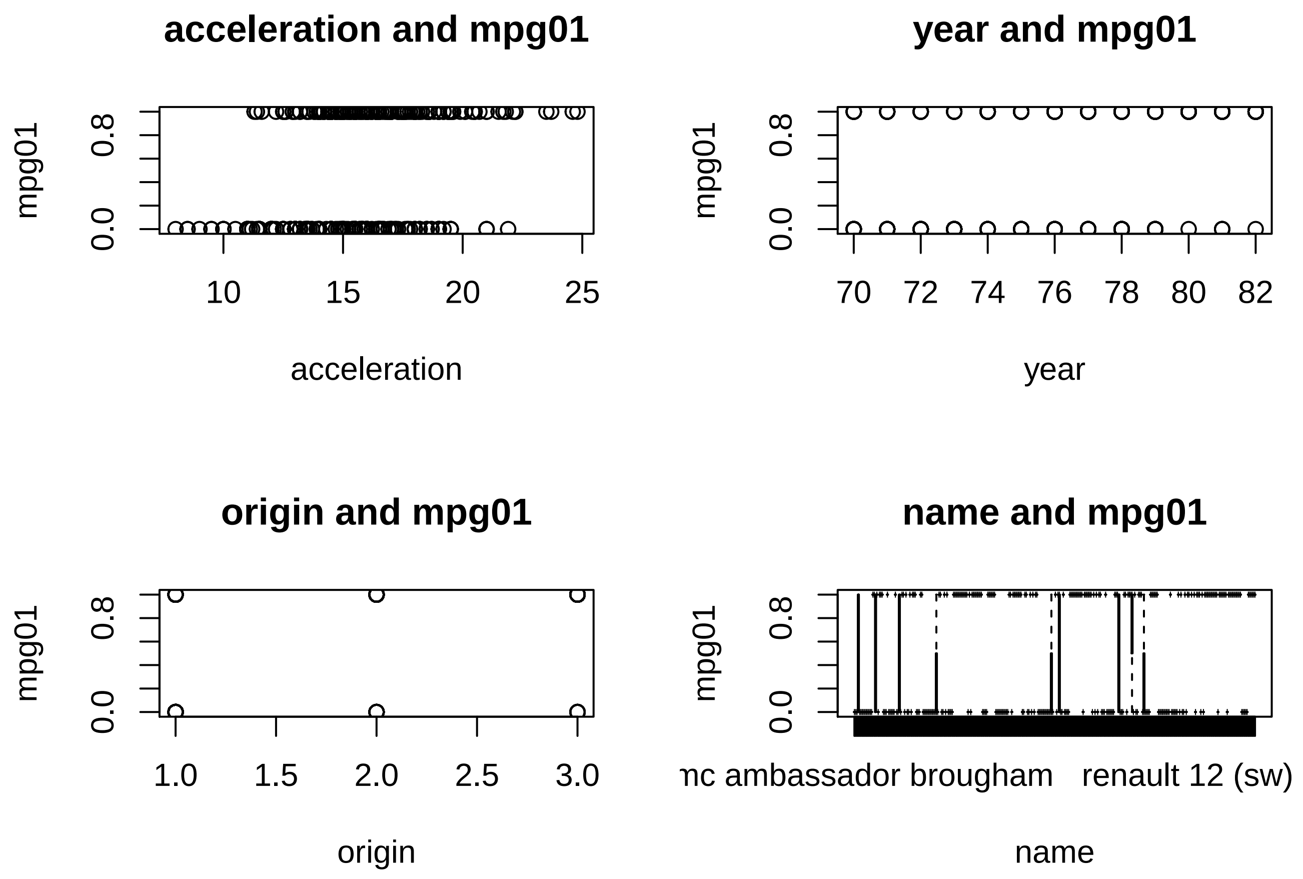
```r
for(i in 2:length(Q11)){
  plot(Q11[,i], Q11$mpg01,xlab = names(Q11)[i],
       ylab = "mpg01",
       main = paste(names(Q11)[i],"and mpg01"))
}
```





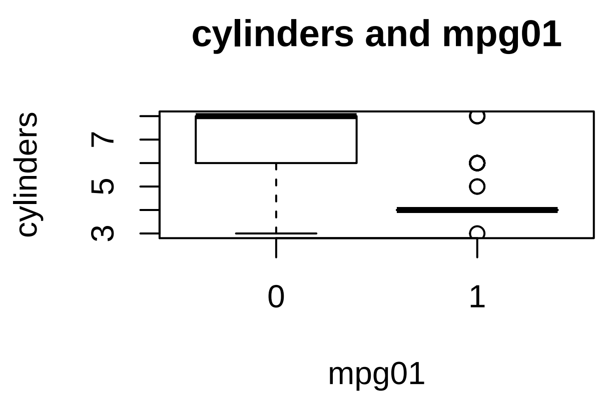


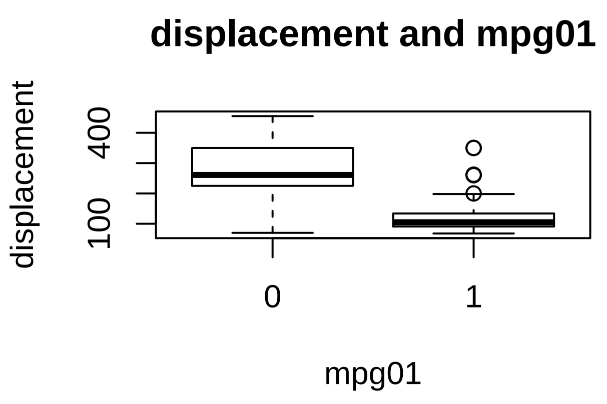


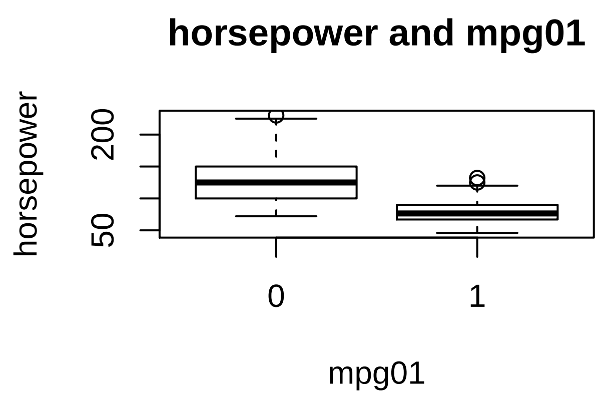


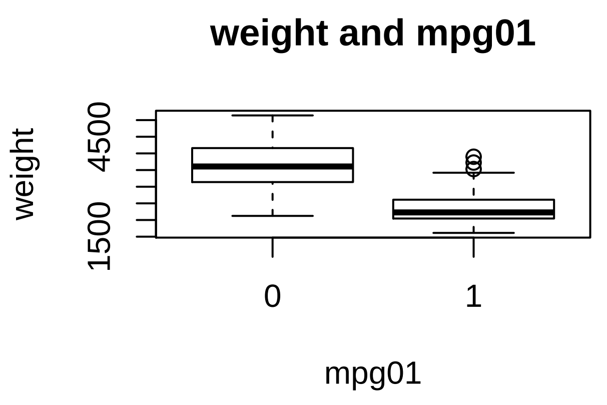
Higher displacement, horsepower, weight may have higher probability that mpg01 is equal to 0. Lower acceleration may have higher probability that mpg01 is equal to 0, but the effect is not as strong as the three variables above. name variable is obviously useless.

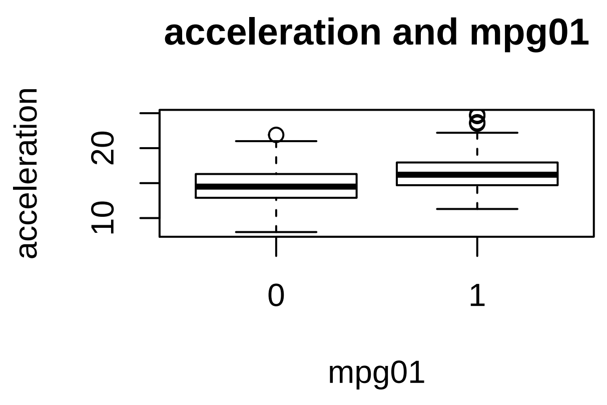
```
for(i in 2:(length(Q11)-1)){
  boxplot(Q11[,i] ~ Q11$mpg01,
          ylab = names(Q11)[i], xlab = "mpg01",
          main = paste(names(Q11)[i],"and mpg01"))
  }
```

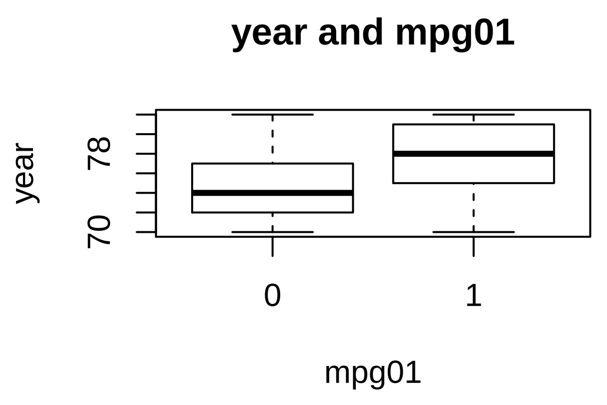

cylinders and mpg01
cylinders
3
5
7
0
1
mpg01

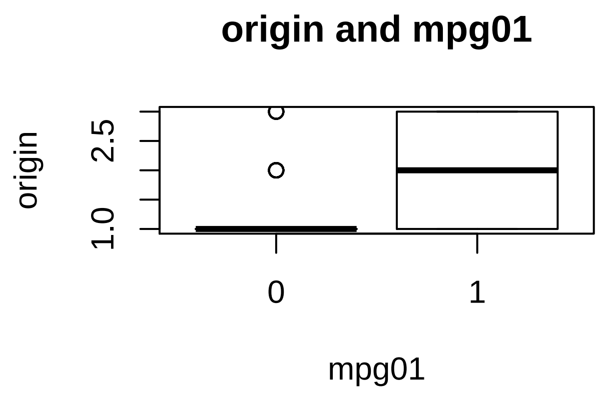

displacement and mpg01
displacement
100
400
0
1
mpg01


horsepower and mpg01
horsepower
50
200
0
1
mpg01


weight and mpg01
weight
1500
4500
0
1
mpg01

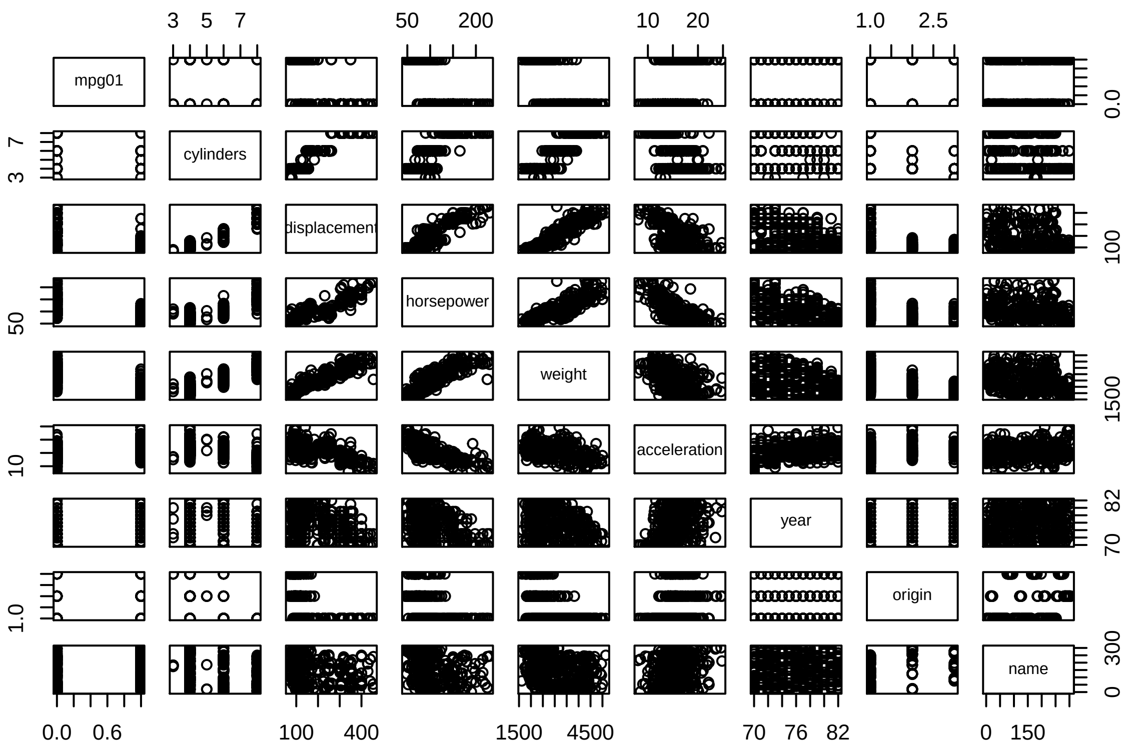

acceleration and mpg01
acceleration
10
20
0
1
mpg01


year and mpg01
year
70
78
0
1
mpg01



In boxplot, we can find that higher cylinders may may have higher probability that mpg01 is equal to 0, while lower year and origin may may have higher probability that mpg01 is equal to 0, but the effect of origin is not as strong as the three variables above.

```
pairs(Q11)
```



We can find that displacement, horsepower and weight have linear relationship.

```
cor(Q11[3:5])
```

```
##              displacement horsepower    weight
## displacement    1.0000000  0.8972570 0.9329944
## horsepower      0.8972570  1.0000000 0.8645377
## weight          0.9329944  0.8645377 1.0000000
```

From the correlation matrix, we find that displacement has high correlation with horsepower and weight(about 0.9). It can be represented by horsepower and weight

So we choose cylinders, horsepower, weight, year as predictors.

(c)

```
set.seed(1111)
train <- sample(392,size = 294)
Q11_train <- Q11[train,-9]
Q11_test <- Q11[-train,-9]
```

(d) LDA

```
library(MASS)
lda.fit <- lda(mpg01 ~ cylinders +  horsepower + weight + year, data = Q11_train)
lda.pred_train <- predict(lda.fit, Q11_train)$class
lda.cfmatrix_train <- table(lda.pred_train ,Q11_train$mpg01)
lda.cfmatrix_train
```

```
##
## lda.pred_train   0   1
##              0 123   6
##              1  22 143
```

```
lda.pred_test <- predict(lda.fit, Q11_test)$class
lda.cfmatrix_test <- table(lda.pred_test ,Q11_test$mpg01)
lda.cfmatrix_test
```

```
##
## lda.pred_test  0  1
##             0 46  1
##             1  5 46
```

```
lda.test_error <- (lda.cfmatrix_test[1,2]+lda.cfmatrix_test[2,1])/nrow(Q11_test)
lda.test_error
```

```
## [1] 0.06122449
```

(e) QDA

```
qda.fit <- qda(mpg01 ~ cylinders +  horsepower + weight + year, data = Q11_train)
qda.pred_train <- predict(qda.fit, Q11_train)$class
qda.cfmatrix_train <- table(qda.pred_train ,Q11_train$mpg01)
qda.cfmatrix_train
```

```
##
## qda.pred_train   0   1
##              0 126  11
##              1  19 138
```

```
qda.pred_test <- predict(qda.fit, Q11_test)$class
qda.cfmatrix_test <- table(qda.pred_test ,Q11_test$mpg01)
qda.cfmatrix_test
```

```
##
## qda.pred_test  0  1
##             0 46  5
##             1  5 42
```

```
qda.test_error <- (qda.cfmatrix_test[1,2]+qda.cfmatrix_test[2,1])/nrow(Q11_test)
qda.test_error
```

```
## [1] 0.1020408
```

(f) Logistic

```
glm.fit <- glm(mpg01 ~ cylinders +  horsepower + weight + year, data = Q11_train,
               family=binomial)
glm.prob_train <- predict(glm.fit,type="response")
glm.pred_train <- rep(0, nrow(Q11_train))
glm.pred_train[glm.prob_train > 0.5] <- 1
glm.cfmatrix_train <- table(glm.pred_train ,Q11_train$mpg01)
glm.cfmatrix_train
```

```
##
## glm.pred_train   0   1
##              0 127  14
##              1  18 135
```

```
glm.prob_test <- predict(glm.fit,newdata = Q11_test, type="response")
glm.pred_test <- rep(0, nrow(Q11_test))
glm.pred_test[glm.prob_test > 0.5] <- 1
glm.cfmatrix_test <- table(glm.pred_test ,Q11_test$mpg01)
glm.cfmatrix_test
```

```
##
## glm.pred_test  0  1
##             0 46  1
##             1  5 46
```

```
glm.test_error <- (glm.cfmatrix_test[1,2]+glm.cfmatrix_test[2,1])/nrow(Q11_test)
glm.test_error
```

```
## [1] 0.06122449
```

(g) KNN

```
library(class)
train_x <- Q11_train[,c("cylinders", "horsepower", "weight", "year")]
test_x <- Q11_test[,c("cylinders", "horsepower", "weight", "year")]
train_y <- Q11_train$mpg01
test_y <- Q11_test$mpg01
knn.test_error <- NULL
for(k in 1:50){
  knn.pred <- knn(train_x,test_x,train_y ,k)
  knn.cfmatrix_train <- table(knn.pred,test_y)
  knn.test_error <- c(knn.test_error,
                      (knn.cfmatrix_train[1,2]+knn.cfmatrix_train[2,1])/nrow(Q11_test))
}
which.min(knn.test_error)
```

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

```
knn.test_error[1]
```

```
## [1] 0.07142857
```