Mehmet Duman

Solution to Homework #3—MTH 522

Problem 1 (Chapter 4 Exercises 7):
Suppose that we wish to predict whether a given stock will issue a dividend this year (Yes or No) based on X , last years percent profit. We examine a large number of companies and discover that the mean value of X for companies that issued a dividend was X = 10, while the mean for those that didnt was X = 0. In addition, the variance of X for these two sets of companies was $\sigma^2 = 36$. Finally, 80% of companies issued dividends. Assuming that X follows a normal distribution, predict the probability that a company will issue a dividend this year given that its percentage return was X = 4 last year.

You will need to use Bayes theorem.

Solution:
From text book 4.11 (Page 139);

$$p_k(x) = \frac{\pi_k \frac{1}{\sqrt{2\pi\sigma}} \exp(-\frac{1}{2\sigma^2}(x-\mu_k)^2)}{\sum \pi_l \frac{1}{\sqrt{2\pi\sigma}} \exp(-\frac{1}{2\sigma^2}(x-\mu_l)^2)} \tag{1}$$

$$p_k(x) = \frac{\cancel{\frac{1}{\sqrt{2\pi\sigma}}} \pi_k \exp(-\frac{1}{2\sigma^2}(x-\mu_k)^2)}{\cancel{\frac{1}{\sqrt{2\pi\sigma}}} \sum \pi_l \exp(-\frac{1}{2\sigma^2}(x-\mu_l)^2)}$$

$$p_{yes}(x) = \frac{\pi_{yes} \exp(-\frac{1}{2\sigma^2}(x-\mu_{yes})^2)}{\sum \pi_l \exp(-\frac{1}{2\sigma^2}(x-\mu_l)^2)}$$

$$p_{yes}(x) = \frac{\pi_{yes} \exp(-\frac{1}{2\sigma^2}(x-\mu_{yes})^2)}{\pi_{yes} \exp(-\frac{1}{2\sigma^2}(x-\mu_{yes})^2) + \pi_{no} \exp(-\frac{1}{2\sigma^2}(x-\mu_{no})^2)}$$

$$p_{yes}(x) = \frac{0.8 \exp(-\frac{1}{2*36}(x-10)^2)}{0.8 \exp(-\frac{1}{2*36}(x-10)^2) + 0.2 \exp(-\frac{1}{2*36}(x-0)^2)}$$

$$p_{yes}(x) = \frac{0.8 \exp(-\frac{1}{2*36}(x-10)^2)}{0.8 \exp(-\frac{1}{2*36}(x-10)^2) + 0.2 \exp(-\frac{1}{2*36}(x-0)^2)}$$

Percentage return X=4 for last year

$$p_{yes}(4) = \frac{0.4852245}{0.4852245 + 0.1601475} = 0.7518524 = \%75.19 \tag{2}$$

From the question 7; The probability that the company will issue a dividend this year given that its percentage return was X = 4 last year : %75.19

Problem 2 (Chapter 4 Exercises 10): This question should be answered using the Weekly data set, which is part of the ISLR package. This data is similar in nature to the Smarket data from this chapters lab, except that it contains 1,089 weekly returns for 21 years, from the beginning of 1990 to the end of 2010.

(a) Produce some numerical and graphical summaries of the Weekly data. Do there appear to be any patterns?

```
> library(MASS)
> library(ISLR)
> summary(Weekly)

      Year           Lag1               Lag2               Lag3
 Min.   :1990   Min.   :-18.1950   Min.   :-18.1950   Min.   :-18.1950
 1st Qu.:1995   1st Qu.: -1.1540   1st Qu.: -1.1540   1st Qu.: -1.1580
 Median :2000   Median :  0.2410   Median :  0.2410   Median :  0.2410
 Mean   :2000   Mean   :  0.1506   Mean   :  0.1511   Mean   :  0.1472
 3rd Qu.:2005   3rd Qu.:  1.4050   3rd Qu.:  1.4090   3rd Qu.:  1.4090
 Max.   :2010   Max.   : 12.0260   Max.   : 12.0260   Max.   : 12.0260
      Lag4               Lag5               Volume            Today              Direction
 Min.   :-18.1950   Min.   :-18.1950   Min.   :0.08747   Min.   :-18.1950   Down:484
 1st Qu.: -1.1580   1st Qu.: -1.1660   1st Qu.:0.33202   1st Qu.: -1.1540   Up  :605
 Median :  0.2380   Median :  0.2340   Median :1.00268   Median :  0.2410
 Mean   :  0.1458   Mean   :  0.1399   Mean   :1.57462   Mean   :  0.1499
 3rd Qu.:  1.4090   3rd Qu.:  1.4050   3rd Qu.:2.05373   3rd Qu.:  1.4050
 Max.   : 12.0260   Max.   : 12.0260   Max.   :9.32821   Max.   : 12.0260
```

```
> pairs(Weekly)
> dev.copy(png,"MTH522_hw3_p2a.png",width=8,height=6,units="in",res=200)
> dev.off()
```

Program 1: The R code generate Figure. 1.

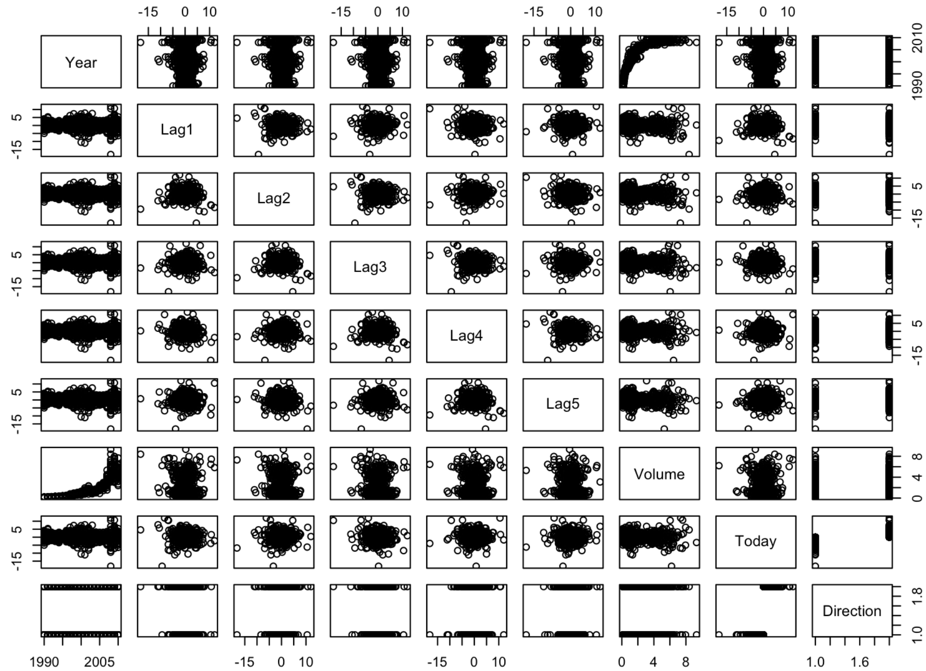

Figure 1: .

```
> library(MASS)
> library(ISLR)
> cor(Weekly[, -9])
Year      Lag1     Lag2    Lag3     Lag4      Lag5  Volume    Today
Year    1.0000 -0.03229 -0.0334 -0.0300 -0.03113 -0.03052  0.8419 -0.03246
Lag1   -0.0323  1.00000 -0.0749  0.0586 -0.07127 -0.00818 -0.0650 -0.07503
Lag2   -0.0334 -0.07485  1.0000 -0.0757  0.05838 -0.07250 -0.0855  0.05917
Lag3   -0.0300  0.05864 -0.0757  1.0000 -0.07540  0.06066 -0.0693 -0.07124
Lag4   -0.0311 -0.07127  0.0584 -0.0754  1.00000 -0.07568 -0.0611 -0.00783
Lag5   -0.0305 -0.00818 -0.0725  0.0607 -0.07568  1.00000 -0.0585  0.01101
Volume  0.8419 -0.06495 -0.0855 -0.0693 -0.06107 -0.05852  1.0000 -0.03308
Today  -0.0325 -0.07503  0.0592 -0.0712 -0.00783  0.01101 -0.0331  1.00000
```

There isn't any corelation between 'Lagx' and 'Today', but we have corealation between 'Year and 'Volume', there is a corelation. Let's plot this two and see;

```
> plot(Year,Volume)
> dev.copy(png,"MTH522_hw3_p2a2.png",width=8,height=6,units="in",res=200)
> dev.off()
```

Program 2: The R code generate Figure. 2.

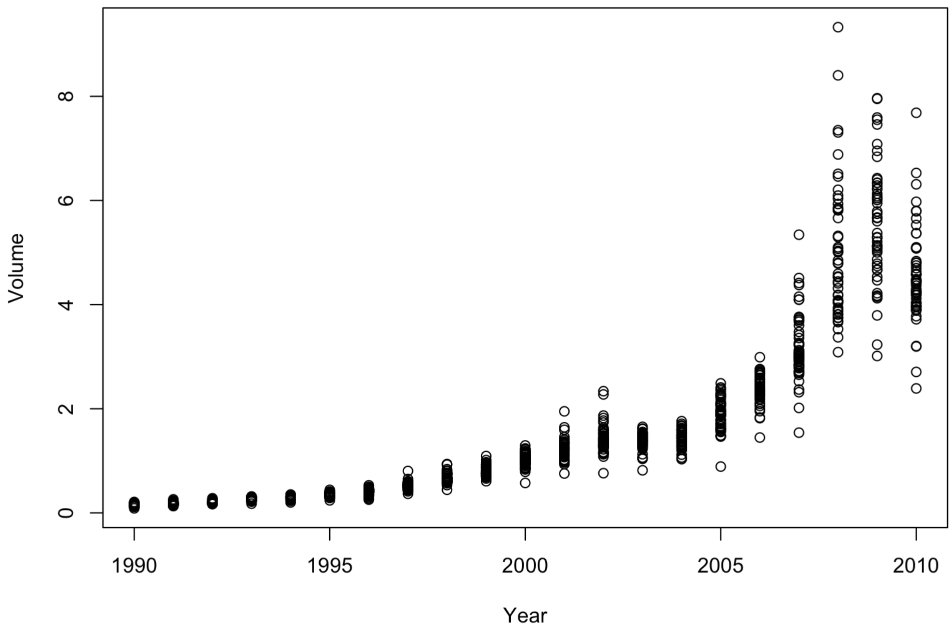

Figure 2: .

(b) Use the full data set to perform a logistic regression with Direction as the response and the five lag variables plus Volume as predictors. Use the summary function to print the results. Do any of the predictors appear to be statistically significant? If so, which ones?

```
> fit.glm <- glm(Direction ~ Lag1 + Lag2 + Lag3 + Lag4 + Lag5 + Volume,
data = Weekly, family = binomial)
> summary(fit.glm)

Call:
glm(formula = Direction ~ Lag1 + Lag2 + Lag3 + Lag4 + Lag5 +
Volume, family = binomial, data = Weekly)

Deviance Residuals:
Min      1Q  Median      3Q     Max
-1.695  -1.256   0.991   1.085   1.458

Coefficients:
            Estimate Std. Error z value  Pr(>|z|)
(Intercept)   0.2669     0.0859    3.11   0.0019 **
Lag1         -0.0413     0.0264   -1.56   0.1181
Lag2          0.0584     0.0269    2.18   0.0296 *
Lag3         -0.0161     0.0267   -0.60   0.5469
Lag4         -0.0278     0.0265   -1.05   0.2937
Lag5         -0.0145     0.0264   -0.55   0.5833
Volume       -0.0227     0.0369   -0.62   0.5377
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 1496.2  on 1088  degrees of freedom
Residual deviance: 1486.4  on 1082  degrees of freedom
AIC: 1500

Number of Fisher Scoring iterations: 4
```

'Lag2' is the only predictor statistically significant with $\Pr(>|z|) = 2.96\%$

(c) Compute the confusion matrix and overall fraction of correct predictions. Explain what the confusion matrix is telling you about the types of mistakes made by logistic regression.

```
> probs = predict(fit.glm, type = "response")
> pred.glm = rep("Down", length(probs))
> pred.glm[probs > 0.5] = "Up"
> table(pred.glm, Direction)

        Direction
pred.glm Down  Up
Down    54  48
Up     430 557
```

The percentage of correct predictions for training data :
(54 + 557) / (54 +48 + 430 +557) = 611 / 1089 = 0.0.5610651974 = 56.1%
So , the training error rate is 1 - 0.0.561065197 = 0.438934803 = 43.89% (From Text Book Page 158) the training error rate is often overly optimistic.

For Weeks,
when the market goes up, the model is right 557 /(557+48) = 92.0661157 = 92.1% most of the time.
when the market goes up, the model is wrong 54/(430+54) = 0.1115702479 = 11.2% most of the time.

(d) Now fit the logistic regression model using a training data period from 1990 to 2008, with Lag2 as the only predictor. Compute the confusion matrix and the overall fraction of correct predictions for the held out data (that is, the data from 2009 and 2010).

```
> train = (Year < 2009)
> Weekly.20092010 = Weekly[!train, ]
> Direction.20092010 = Direction[!train]
> fit.glm2 = glm(Direction ~ Lag2, data = Weekly, family = binomial, subset = train)
> summary(fit.glm2)

Call:
glm(formula = Direction ~ Lag2, family = binomial, data = Weekly,
subset = train)

Deviance Residuals:
Min         1Q     Median         3Q        Max
-1.536167  -1.263542   1.020809   1.090541   1.368474

Coefficients:
Estimate Std. Error z value  Pr(>|z|)
(Intercept) 0.20325743 0.06428036 3.16205 0.0015666 **
Lag2        0.05809527 0.02870446 2.02391 0.0429793 *
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 1354.7097  on 984  degrees of freedom
Residual deviance: 1350.5431  on 983  degrees of freedom
AIC: 1354.5431

Number of Fisher Scoring iterations: 4
```

```
> probs2 = predict(fit.glm2, Weekly.20092010, type = "response")
> pred.glm2 = rep("Down", length(probs2))
> pred.glm2[probs2 > 0.5] = "Up"
> table(pred.glm2, Direction.20092010)

Direction.20092010
pred.glm2 Down Up
Down    9  5
Up     34 56
```

The percentage of correct predictions for data :
(9 + 56) / (9+5+34+56) = 0.625 = 62.5%
So , the error rate is 1 - 0.625 = 0.375 = 37.5% (From Text Book Page 158) the training error rate is often overly optimistic.

For Weeks,
when the market goes up, the model is right 56/(56+5) = 0.9180327869 = 91.8% most of the time.
when the market goes down, the model is right 9/(9+34) =0.2093023256 = 20.9% most of the time.

(e) Repeat (d) using LDA.

```
> library(MASS)
> fit.lda = lda(Direction ~ Lag2, data = Weekly, subset = train)
> fit.lda

Call:
lda(Direction ~ Lag2, data = Weekly, subset = train)

Prior probabilities of groups:
Down         Up
0.447715736 0.552284264

Group means:
Lag2
Down -0.03568253968
Up    0.26036580882

Coefficients of linear discriminants:
LD1
Lag2 0.4414162176
```

```
> pred.lda <- predict(fit.lda, Weekly.20092010)
> table(pred.lda$class, Direction.20092010)

Direction.20092010
     Down Up
Down    9  5
Up     34 56
```

The percentage of correct predictions for test data :
(9 + 56) / (9+5+34+56) = 0.625 = 62.5%
So , the test error rate is 1 - 0.625 = 0.375 = 37.5% (From Text Book Page 158) the training error rate is often overly optimistic.

For Weeks,
when the market goes up, the model is right 56/(56+5) = 0.9180327869 = 91.8% most of the time.
when the market goes down, the model is right 9/(9+34) =0.2093023256 = 20.9% most of the time.

(f) Repeat (d) using QDA.

```
> fit.qda = qda(Direction ~ Lag2, data = Weekly, subset = train)
> fit.qda

Call:
qda(Direction ~ Lag2, data = Weekly, subset = train)

Prior probabilities of groups:
Down         Up
0.447715736 0.552284264

Group means:
Lag2
Down -0.03568253968
Up    0.26036580882
```

```
> pred.qda = predict(fit.qda, Weekly.20092010)
> table(pred.qda$class, Direction.20092010)

Direction.20092010
Down Up
Down    0  0
Up     43 61
```

The percentage of correct predictions for test data :
(61) / (43+61) = 0.5865384615 = 58.65384615%
So , the test error rate is 1 - 0.5865384615 = 0.4134615385 = 41.34615385%
For Weeks;
when the market goes up, the model is right 100% of the time.
when the market goes down, the model is right 0% of the time.
Correct predictions is 58.65384615% even the model is up

(g) Repeat (d) using KNN with K = 1.

```
> library(class)
> train.X = as.matrix(Lag2[train])
> test.X = as.matrix(Lag2[!train])
> train.Direction = Direction[train]
> set.seed(1)
> pred.knn = knn(train.X, test.X, train.Direction, k = 1)
> table(pred.knn, Direction.20092010)


Direction.20092010
pred.knn Down Up
Down   21 30
Up     22 31
```

The percentage of correct predictions for test data : (21+31) / (21+30+22+31) = 0.5 = 50%
So , the test error rate is 50%
For Weeks,
when the market goes up, the model is right 31/(31+30) = 0.5081967213 = 50.81967213% most of the time.
when the market goes down, the model is right 21/(21+22) =0.488372093 = 48.8372093% most of the time.

(h) Which of these methods appears to provide the best results on this data?

Let' compare the test error rates;
The logistic regression model test error rate 37.5%
LDA test error rate 37.5%
QDA test error rate 41.3%
KNN test error rate 50%

So, the logistic regression and LDA have same the minimum error rates.

(i) Experiment with different combinations of predictors, including possible transformations and interactions, for each of the methods. Report the variables, method, and associated confusion matrix that appears to provide the best results on the held out data. Note that you should also experiment with values for K in the KNN classifier.

```
> # Logistic regression with Lag2:Lag1
>  fit.glm3 = glm(Direction ~ Lag2:Lag1, data = Weekly, family = binomial, subset = train)
>  probs3 = predict(fit.glm3, Weekly.20092010, type = "response")
>  pred.glm3 = rep("Down", length(probs3))
>  pred.glm3[probs3 > 0.5] = "Up"
>  table(pred.glm3, Direction.20092010)

Direction.20092010
pred.glm3 Down Up
Down    1  1
Up     42 60
```

```
> mean(pred.glm3 == Direction.20092010)

[1] 0.5865384615
```

```
> # LDA with Lag2 interaction with Lag1
>  fit.lda2 = lda(Direction ~ Lag2:Lag1, data = Weekly, subset = train)
>  pred.lda2 = predict(fit.lda2, Weekly.20092010)
>  mean(pred.lda2$class == Direction.20092010)

[1] 0.5769230769
```

```
> # QDA with sqrt(abs(Lag2))
>  fit.qda2 = qda(Direction ~ Lag2 + sqrt(abs(Lag2)), data = Weekly, subset = train)
>  pred.qda2 = predict(fit.qda2, Weekly.20092010)
>  table(pred.qda2$class, Direction.20092010)

Direction.20092010
Down Up
Down   12 13
Up     31 48
```

```
> mean(pred.qda2$class == Direction.20092010)

[1] 0.5769230769
```

```
>  # KNN k =10
>  pred.knn2 = knn(train.X, test.X, train.Direction, k = 10)
>  table(pred.knn2, Direction.20092010)

Direction.20092010
pred.knn2 Down Up
Down   17 18
Up     26 43
```

```
> mean(pred.knn2 == Direction.20092010)

[1] 0.5769230769
```

```
> # KNN k = 100
>  pred.knn3 = knn(train.X, test.X, train.Direction, k = 100)
>  table(pred.knn3, Direction.20092010)

Direction.20092010
pred.knn3 Down Up
Down    9 12
Up     34 49
```

```
> mean(pred.knn3 == Direction.20092010)

[1] 0.5576923077
```

For the best performance in terms of test error rates, LDA and the original logistic regression model have the best performance.

Problem 3 (Chapter 4 Exercises 11):
In this problem, you will develop a model to predict whether a given car gets high or low gas mileage based on the Auto 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.

```
> attach(Auto)
> mpg01 = rep(0, length(mpg))
> mpg01[mpg > median(mpg)] = 1
> Auto = data.frame(Auto, mpg01)
```

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 usefull tools to answer this question. Describe your findings.

```
> cor(Auto[, -9])
                    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
```

```
> pairs(Auto)
> dev.copy(png,"MTH522_hw3_p3b.png",width=8,height=6,units="in",res=200)
> dev.off()
```

Program 3: The R code generate Figure. 3.

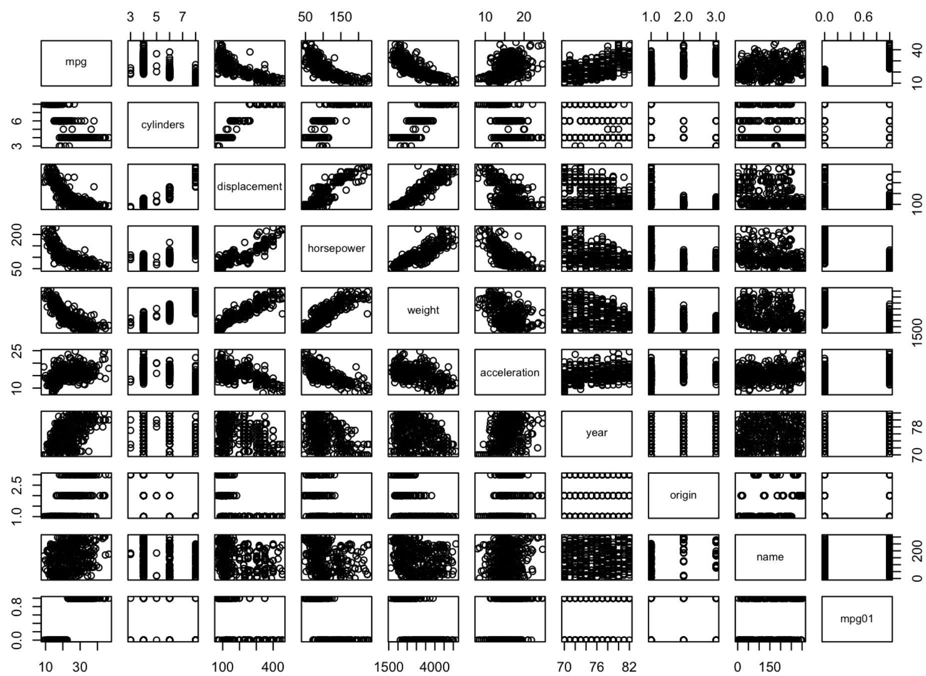

Figure 3: .

```
> boxplot(cylinders ~ mpg01, data = Auto, main = "Cylinders vs mpg01")
>  dev.copy(png,"MTH522_hw3_p3b2.png",width=8,height=6,units="in",res=200)
> dev.off()
```

Program 4: The R code generate Figure. 4.

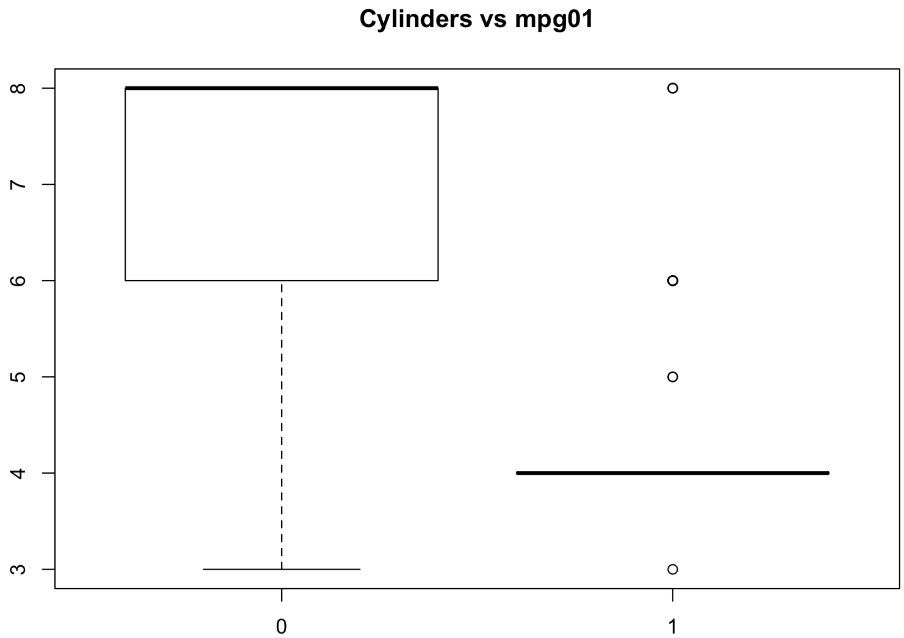

Figure 4: .

```
> boxplot(displacement ~ mpg01, data = Auto, main = "Displacement vs mpg01")
> dev.copy(png,"MTH522_hw3_p3b3.png",width=8,height=6,units="in",res=200)
> dev.off()
```

Program 5: The R code generate Figure. 5.

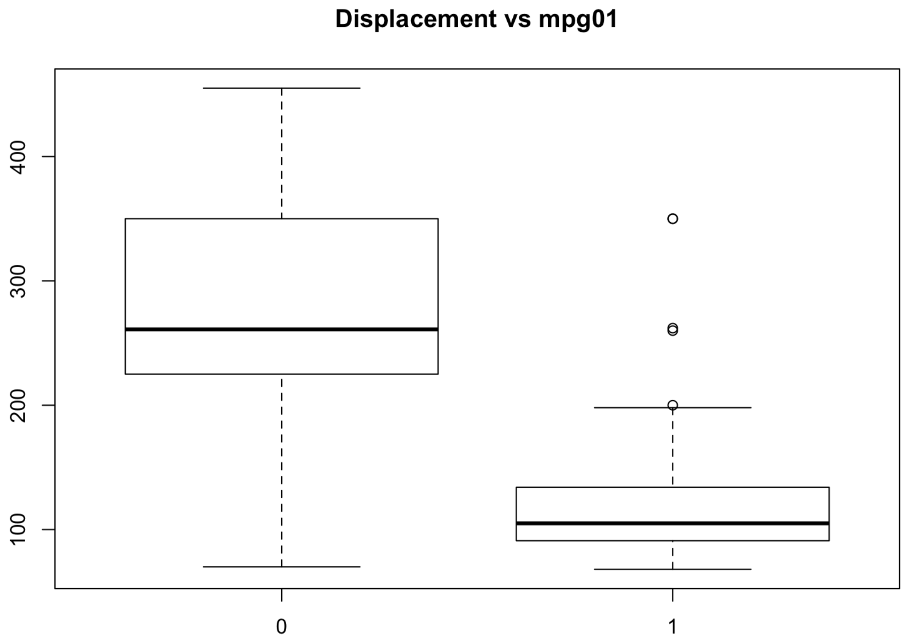

Figure 5: .

```
> boxplot(horsepower ~ mpg01, data = Auto, main = "Horsepower vs mpg01")
> dev.copy(png,"MTH522_hw3_p3b3.png",width=8,height=6,units="in",res=200)
> dev.off()
```

Program 6: The R code generate Figure. 6.

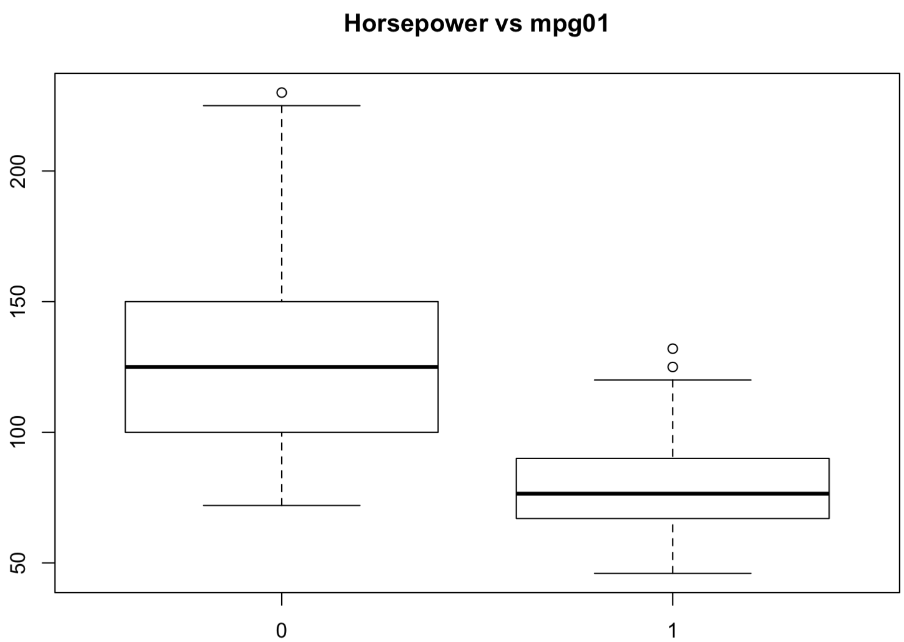

Figure 6: .

```
> boxplot(weight ~ mpg01, data = Auto, main = "Weight vs mpg01")
> dev.copy(png,"MTH522_hw3_p3b5.png",width=8,height=6,units="in",res=200)
> dev.off()
```

Program 7: The R code generate Figure. 7.

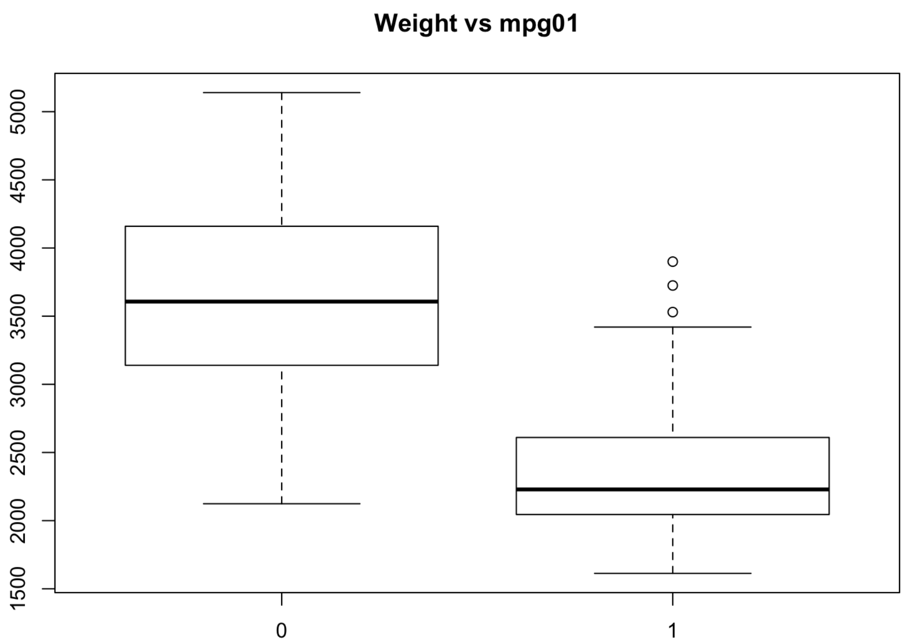

Figure 7: .

```
> boxplot(acceleration ~ mpg01, data = Auto, main = "Acceleration vs mpg01")
> dev.copy(png,"MTH522_hw3_p3b6.png",width=8,height=6,units="in",res=200)
> dev.off()
```

Program 8: The R code generate Figure. 8.

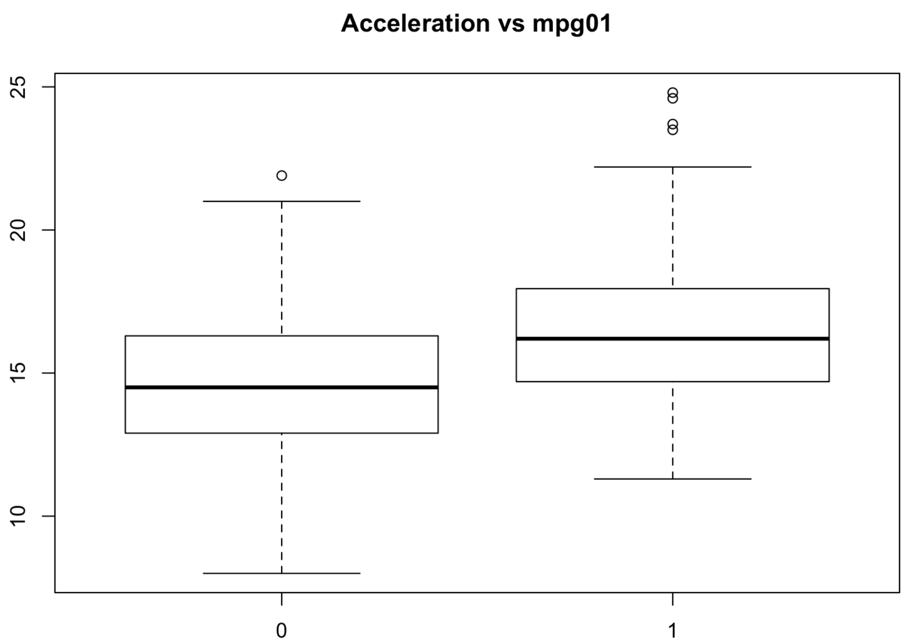

Figure 8: .

```
> boxplot(year ~ mpg01, data = Auto, main = "Year vs mpg01")
> dev.copy(png,"MTH522_hw3_p3b7.png",width=8,height=6,units="in",res=200)
> dev.off()
```

Program 9: The R code generate Figure. 9.

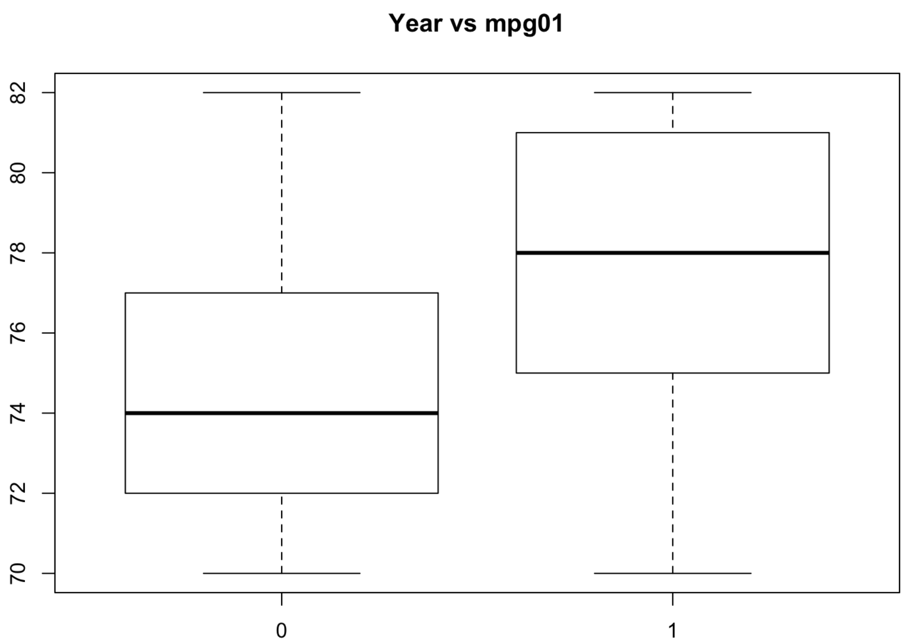

Figure 9: .

As we can see from the figures, there exists some association between mpg01 and cylinders, weight, displ

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

```
> train = (year %% 2 == 0)
> Auto.train = Auto[train, ]
> Auto.test = Auto[!train, ]
> mpg01.test = mpg01[!train]
```

(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?

```
> fit.lda = lda(mpg01 ~ cylinders + weight + displacement + horsepower,data = Auto, subset = train)
> fit.lda
Call:
lda(mpg01 ~ cylinders + weight + displacement + horsepower, data = Auto,
subset = train)

Prior probabilities of groups:
0         1
0.4571429 0.5428571

Group means:
cylinders   weight displacement horsepower
0  6.812500 3604.823     271.7396  133.14583
1  4.070175 2314.763     111.6623   77.92105

Coefficients of linear discriminants:
LD1
cylinders    -0.6741402638
weight       -0.0011465750
displacement  0.0004481325
horsepower    0.0059035377
```

```
> pred.lda = predict(fit.lda, Auto.test)
> table(pred.lda$class, mpg01.test)
mpg01.test
0  1
0 86  9
1 14 73
```

```
> mean(pred.lda$class != mpg01.test)
[1] 0.1263736
```

The test error rate of the model is 12.6373626%.

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?

```
> fit.qda = qda(mpg01 ~ cylinders + weight + displacement + horsepower,data = Auto, subset = train)
> fit.qda

Call:
qda(mpg01 ~ cylinders + weight + displacement + horsepower, data = Auto,
subset = train)

Prior probabilities of groups:
0         1
0.4571429 0.5428571

Group means:
cylinders   weight displacement horsepower
0  6.812500 3604.823     271.7396  133.14583
1  4.070175 2314.763     111.6623   77.92105
```

```
> pred.qda = predict(fit.qda, Auto.test)
> table(pred.qda$class, mpg01.test)

   mpg01.test
   0  1
   0 89 13
   1 11 69
```

```
> mean(pred.qda$class != mpg01.test)
[1] 0.1318681
```

The test error rate of the model is 13.18681%.

(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?

```
> fit.glm = glm(mpg01 ~ cylinders + weight + displacement + horsepower, data = Auto, family = binomial,
>  summary(fit.glm)

Call:
glm(formula = mpg01 ~ cylinders + weight + displacement + horsepower,
family = binomial, data = Auto, subset = train)

Deviance Residuals:
Min        1Q    Median        3Q       Max
-2.48027  -0.03413   0.10583   0.29634   2.57584

Coefficients:
Estimate Std. Error z value Pr(>|z|)
(Intercept)  17.658730   3.409012   5.180 2.22e-07 ***
cylinders    -1.028032   0.653607  -1.573   0.1158
weight       -0.002922   0.001137  -2.569   0.0102 *
displacement  0.002462   0.015030   0.164   0.8699
horsepower   -0.050611   0.025209  -2.008   0.0447 *
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 289.58  on 209  degrees of freedom
Residual deviance:  83.24  on 205  degrees of freedom
AIC: 93.24

Number of Fisher Scoring iterations: 7
```

```
> probs = predict(fit.glm, Auto.test, type = "response")
>  pred.glm = rep(0, length(probs))
>  pred.glm[probs > 0.5] = 1
>  table(pred.glm, mpg01.test)
    mpg01.test
pred.glm  0  1
       0 89 11
       1 11 71
```

```
> mean(pred.glm != mpg01.test)
[1] 0.1208791
```

The test error rate of the model is 12.08791%.

(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?

```
> library(class)
> train.X = cbind(cylinders, weight, displacement, horsepower)[train, ]
> test.X = cbind(cylinders, weight, displacement, horsepower)[!train, ]
> train.mpg01 = mpg01[train]
> set.seed(1)
> pred.knn = knn(train.X, test.X, train.mpg01, k = 1)
> table(pred.knn, mpg01.test)
mpg01.test
pred.knn  0  1
0 83 11
1 17 71
```

```
> mean(pred.knn != mpg01.test)
[1] 0.1538462
```

The test error rate of the model is 15.38462% for k=1.

```
> pred.knn = knn(train.X, test.X, train.mpg01, k = 10)
> table(pred.knn, mpg01.test)
mpg01.test
pred.knn  0  1
0 77  7
1 23 75




> mean(pred.knn != mpg01.test)
[1] 0.1648352
```

The test error rate of the model is 16.48352% for k=10.

```
> pred.knn = knn(train.X, test.X, train.mpg01, k = 100)
>  table(pred.knn, mpg01.test)

mpg01.test
pred.knn  0  1
0 81  7
1 19 75




> mean(pred.knn != mpg01.test)
[1] 0.1428571
```

The test error rate of the model is 14.28571% for k=100.
k =100 is the smallest error rate of model, k=100 seems to perform the best.

Problem 4 (Chapter 4 Exercises 13):
Using the Boston data set, fit classification models in order to predict whether a given suburb has a crime rate above or below the median. Explore logistic regression, LDA, and KNN models using various subsets of the predictors. Describe your findings.

```
> library(MASS)
> summary(Boston)
crim                 zn              indus            chas
Min.   : 0.00632   Min.   :  0.00   Min.   : 0.46   Min.   :0.00000
1st Qu.: 0.08204   1st Qu.:  0.00   1st Qu.: 5.19   1st Qu.:0.00000
Median : 0.25651   Median :  0.00   Median : 9.69   Median :0.00000
Mean   : 3.61352   Mean   : 11.36   Mean   :11.14   Mean   :0.06917
3rd Qu.: 3.67708   3rd Qu.: 12.50   3rd Qu.:18.10   3rd Qu.:0.00000
Max.   :88.97620   Max.   :100.00   Max.   :27.74   Max.   :1.00000
nox                 rm             age              dis
Min.   :0.3850   Min.   :3.561   Min.   :  2.90   Min.   : 1.130
1st Qu.:0.4490   1st Qu.:5.886   1st Qu.: 45.02   1st Qu.: 2.100
Median :0.5380   Median :6.208   Median : 77.50   Median : 3.207
Mean   :0.5547   Mean   :6.285   Mean   : 68.57   Mean   : 3.795
3rd Qu.:0.6240   3rd Qu.:6.623   3rd Qu.: 94.08   3rd Qu.: 5.188
Max.   :0.8710   Max.   :8.780   Max.   :100.00   Max.   :12.127
rad              tax           ptratio          black
Min.   : 1.000   Min.   :187.0   Min.   :12.60   Min.   :  0.32
1st Qu.: 4.000   1st Qu.:279.0   1st Qu.:17.40   1st Qu.:375.38
Median : 5.000   Median :330.0   Median :19.05   Median :391.44
Mean   : 9.549   Mean   :408.2   Mean   :18.46   Mean   :356.67
3rd Qu.:24.000   3rd Qu.:666.0   3rd Qu.:20.20   3rd Qu.:396.23
Max.   :24.000   Max.   :711.0   Max.   :22.00   Max.   :396.90
lstat            medv           crim01
Min.   : 1.73   Min.   : 5.00   Min.   :0.0
1st Qu.: 6.95   1st Qu.:17.02   1st Qu.:0.0
Median :11.36   Median :21.20   Median :0.5
Mean   :12.65   Mean   :22.53   Mean   :0.5
3rd Qu.:16.95   3rd Qu.:25.00   3rd Qu.:1.0
Max.   :37.97   Max.   :50.00   Max.   :1.0
```

```
> library(MASS)
> attach(Boston)
> crim01 = rep(0, length(crim))
> crim01[crim > median(crim)] = 1
> Boston = data.frame(Boston, crim01)
>
> train = 1:(length(crim) / 2)
> test = (length(crim) / 2 + 1):length(crim)
> Boston.train = Boston[train, ]
> Boston.test = Boston[test, ]
> crim01.test = crim01[test]
>
> fit.glm = glm(crim01 ~ . - crim01 - crim, data = Boston, family=binomial, subset=train)

Warning message:
glm.fit: fitted probabilities numerically 0 or 1 occurred




>  probs <- predict(fit.glm, Boston.test, type = "response")
>  pred.glm <- rep(0, length(probs))
>  pred.glm[probs > 0.5] <- 1
>  table(pred.glm, crim01.test)

crim01.test
pred.glm   0   1
       0  68  24
       1  22 139


> mean(pred.glm != crim01.test)

[1] 0.1818182
```

The test error rate of the model is 18.18182%.

```
> fit.glm=glm(crim01 ~ . - crim01 - crim - chas - tax,data=Boston,family=binomial,subset=train)

Warning message:
glm.fit: fitted probabilities numerically 0 or 1 occurred


> probs = predict(fit.glm, Boston.test, type = "response")
> pred.glm = rep(0, length(probs))
> pred.glm[probs > 0.5] = 1
> table(pred.glm, crim01.test)
crim01.test
pred.glm   0   1
      0  67  24
      1  23 139


> mean(pred.glm != crim01.test)

[1] 0.1857708
```

The test error rate of the model is 18.57708%.

```
> fit.lda = lda(crim01 ~ . - crim01 - crim, data = Boston, subset = train)
>  pred.lda = predict(fit.lda, Boston.test)
>  table(pred.lda$class, crim01.test)
crim01.test
    0   1
0  80  24
1  10 139

> mean(pred.lda$class != crim01.test)
[1] 0.1343874
```

The test error rate of the model (LDA) is 13.43874%.

```
> fit.lda = lda(crim01 ~ . - crim01 - crim - chas - tax, data = Boston, subset = train )
>  pred.lda = predict(fit.lda, Boston.test)
>  table(pred.lda$class, crim01.test)

crim01.test
      0    1
0  80   21
1   10 142

> mean(pred.lda$class != crim01.test)
[1] 0.1225296
```

The test error rate of the model (LDA) is 12.25296%.

```
> library(class)
> train.X <- cbind(zn, indus, chas, nox, rm, age, dis, rad, tax, ptratio, black, lstat,medv)[train, ]
> test.X = cbind(zn, indus, chas, nox, rm, age, dis, rad, tax, ptratio, black, lstat,medv)[test, ]
> train.crim01 = crim01[train]
> set.seed(1)
> pred.knn <- knn(train.X, test.X, train.crim01, k = 1)
> table(pred.knn, crim01.test)

crim01.test
pred.knn   0   1
0  85 111
1   5  52

> mean(pred.knn != crim01.test)
[1] 0.458498
```

The test error rate of the model KNN (k=1) is 45.8498%.

```
> pred.knn <- knn(train.X, test.X, train.crim01, k = 10)
>  table(pred.knn, crim01.test)

crim01.test
pred.knn   0   1
0  83  23
1   7 140


> mean(pred.knn != crim01.test)
[1] 0.1185771
```

The test error rate of the model KNN (k=10) is 11.85771%.

```
> pred.knn <- knn(train.X, test.X, train.crim01, k = 100)
>  table(pred.knn, crim01.test)

crim01.test
pred.knn   0   1
0  86 120
1   4  43



> mean(pred.knn != crim01.test)
[1] 0.4901186
```

The test error rate of the model KNN (k=100) is 49.01186%.