SAMBANDH BHUSAN DHAL
ISEN – 613 (ASSIGNMENT 3) *UIN :726006080*

Problem 1

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 chapter's 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?

(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?

(c) Compute the confusion matrix and performance measures (accuracy, error rate, sensitivity, specificity). Explain what the confusion matrix is telling you about the types of mistakes made by logistic regression. Does the error rate represent the performance of logistic regression in prediction? (hint: is it training error rate or test error rate?)

(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 performance measures (accuracy, error rate, sensitivity, specificity) for the held out data (that is, the data from 2009 and 2010).

(e) Repeat (d) using LDA.

(f) Repeat (d) using QDA.

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

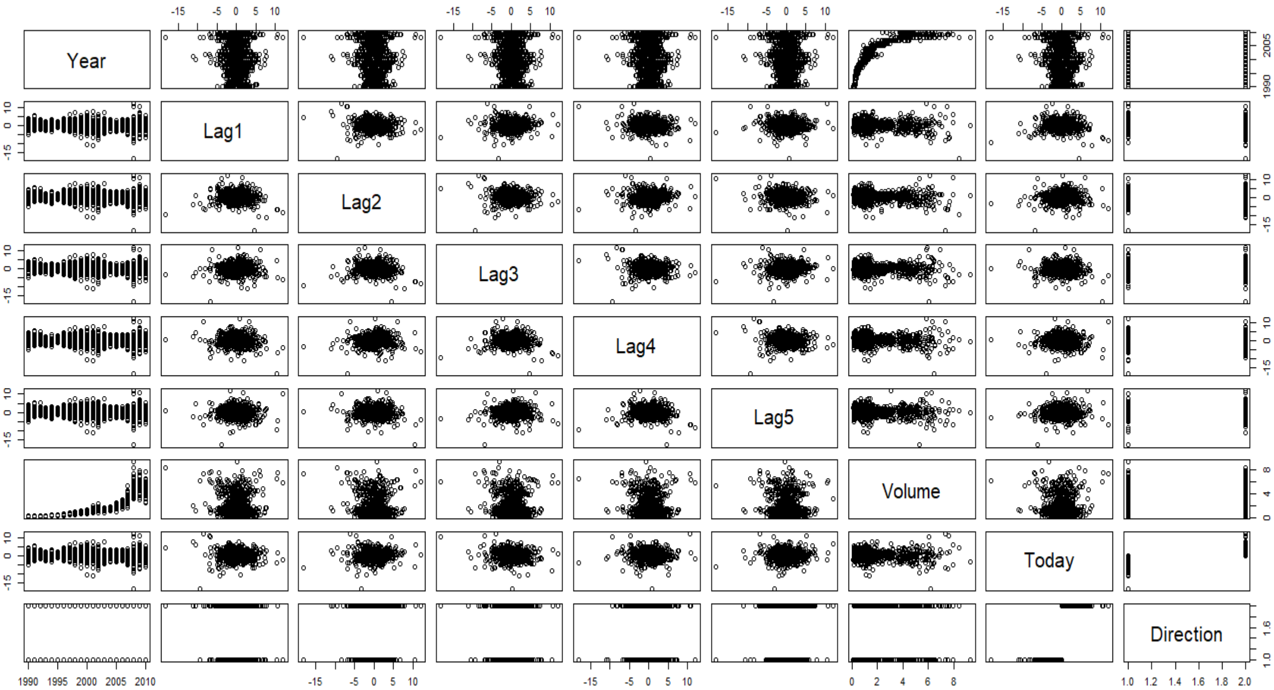
(h) Which of these methods appears to provide the best results on this data?

(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 classifiers.

Ans 1

(a).

```
> library(ISLR)
Warning message:
package 'ISLR' was built under R version 3.4.3
> library(MASS)
> attach(Weekly)
> summary(Weekly)
      Year          Lag1               Lag2               Lag3               Lag4               Lag5              Volume            Today            Direction
 Min.   :1990   Min.   :-18.1950   Min.   :-18.1950   Min.   :-18.1950   Min.   :-18.1950   Min.   :-18.1950   Min.   :0.08747   Min.   :-18.1950   Down:484
 1st Qu.:1995   1st Qu.: -1.1540   1st Qu.: -1.1540   1st Qu.: -1.1580   1st Qu.: -1.1580   1st Qu.: -1.1660   1st Qu.:0.33202   1st Qu.: -1.1540   Up  :605
 Median :2000   Median :  0.2410   Median :  0.2410   Median :  0.2410   Median :  0.2380   Median :  0.2340   Median :1.00268   Median :  0.2410
 Mean   :2000   Mean   :  0.1506   Mean   :  0.1511   Mean   :  0.1472   Mean   :  0.1458   Mean   :  0.1399   Mean   :1.57462   Mean   :  0.1499
 3rd Qu.:2005   3rd Qu.:  1.4050   3rd Qu.:  1.4090   3rd Qu.:  1.4090   3rd Qu.:  1.4090   3rd Qu.:  1.4050   3rd Qu.:2.05373   3rd Qu.:  1.4050
 Max.   :2010   Max.   : 12.0260   Max.   : 12.0260   Max.   : 12.0260   Max.   : 12.0260   Max.   : 12.0260   Max.   :9.32821   Max.   : 12.0260
```



Here, we can see that year and volume have a positive correlation among them.

Volume and year also have a positive correlation between them.

Baring that, no other variables seem to have any correlation among them.

(b).

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

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

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.6949  -1.2565   0.9913   1.0849   1.4579

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  0.26686    0.08593   3.106   0.0019 **
Lag1        -0.04127    0.02641  -1.563   0.1181
Lag2         0.05844    0.02686   2.175   0.0296 *
Lag3        -0.01606    0.02666  -0.602   0.5469
Lag4        -0.02779    0.02646  -1.050   0.2937
Lag5        -0.01447    0.02638  -0.549   0.5833
Volume      -0.02274    0.03690  -0.616   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: 1500.4

Number of Fisher Scoring iterations: 4
```

The p value for lag2 is small (0.02) which shows that there is a relationship between Direction and lag2.

The value of slope is positive which shows that there is a positive relationship among them.

(c).

```
> logistic.probs=predict(logistic.fit,type="response")

> logistic.pred=rep("Down",1089)
> logistic.pred[logistic.probs>0.5]="Up"
> table(logistic.pred,Direction)
             Direction
logistic.pred Down  Up
         Down   54  48
         Up    430 557
> mean(logistic.pred==Direction)
[1] 0.5610652
```

Training accuracy= 56.10%

Error rate=1-Training accuracy= 100-56.10= 43.9%

Sensitivity= TP/P = 92.07%

Specificity= TN/N = 11.16%

Thus, we see that logistic regression predicts one type of class more than the other, and there is always a tradeoff between Sensitivity and Specificity, even though ideally we would like them both to be high. The error rate found here is the training error rate.

(d).

```
> train=(Year<2009)
> Weekly.2009=Weekly[!train,]
> Direction.2009=Direction[!train]
> logistic.fit=glm(Direction~Lag2,data=Weekly,family=binomial,subset=train)
> logistic.probs=predict(logistic.fit,Weekly.2009,data=Weekly,type="response")
> logistic.pred=rep("Down",104)
> logistic.pred[logistic.probs>0.5]="Up"
> table(logistic.pred,Direction.2009)
             Direction.2009
logistic.pred Down Up
         Down    9  5
         Up     34 56
> mean(logistic.pred==Direction.2009)
[1] 0.625
>
```

Training accuracy= 62.5%

Error rate=1-Training accuracy= 100-62.5= 37.5%

Sensitivity= TP/P = 91.8%

Specificity= TN/N = 20.93%

(e).

```
> lda.fit=lda(Direction~Lag2,data=Weekly,subset=train)
> lda.pred=predict(lda.fit,Weekly.2009)
> lda.class=lda.pred$class
> table(lda.class,Direction.2009)
         Direction.2009
lda.class Down Up
     Down    9  5
     Up     34 56
> mean(lda.class==Direction.2009)
[1] 0.625
```

Training accuracy= 62.5%

Error rate=1-Training accuracy= 100-62.5= 37.5%

Sensitivity= TP/P = 87.8%

Specificity= TN/N = 20.93%

(f).

```
> qda.fit=qda(Direction~Lag2,data=Weekly,subset=train)
> qda.pred=predict(qda.fit,Weekly.2009)
> qda.class=qda.pred$class
> table(qda.class,Direction.2009)
         Direction.2009
qda.class Down Up
     Down    0  0
     Up     43 61
> mean(qda.class==Direction.2009)
[1] 0.5865385
```

Training accuracy= 58.65%

Error rate=1-Training accuracy= 100-58.65= 41.35%

Sensitivity= TP/P = 61/61=100%

Specificity= TN/N =0%

(g).

```
> library(class)
> train.X=cbind(Lag2)[train,]
> test.X=cbind(Lag2)[!train,]
> train.Direction=Direction[train]
> set.seed(1)
> knn.pred=knn(data.frame(train.X),data.frame(test.X),train.Direction,k=1)
> table(knn.pred,Direction.2009)
        Direction.2009
knn.pred Down Up
    Down   21 30
    Up     22 31
> mean(knn.pred==Direction.2009)
[1] 0.5
```

Training accuracy= 50%

Error rate=1-Training accuracy= 100-50= 50%

Sensitivity= TP/P = 31/61= 50.81%

Specificity= TN/N =21/43= 48.84%

(h). For the model stated above, both LDA and Logistic regression provide a prediction accuracy of 62.5% on the testing data.

KNN is the worst predictor for this model as it provides a prediction accuracy of just 50% on the testing data.

(i).

```
> logistic_model1=glm(Direction~Lag1:Lag2, data=Weekly, family="binomial",subset=train)
> logistic_probs = predict(logistic_model1,Weekly.2009, type = "response")
> logistic_pred = rep("Down", 104)
> logistic_pred[logistic_probs > 0.5] = "Up"
> table(logistic_pred, Direction.2009)
             Direction.2009
logistic_pred Down Up
         Down    1  1
         Up     42 60
> mean(logistic_pred==Direction.2009)
[1] 0.5865385
```

```
> logistic_model2=glm(Direction~Lag2:Volume, data=Weekly, family="binomial",subset=train
)
> logistic_probs = predict(logistic_model2,Weekly.2009, type = "response")
> logistic_pred = rep("Down", 104)
> logistic_pred[logistic_probs > 0.5] = "Up"
> table(logistic_pred, Direction.2009)
             Direction.2009
logistic_pred Down Up
         Down    9  6
         Up     34 55
> mean(logistic_pred==Direction.2009)
[1] 0.6153846
```

```
> logistic_model3=glm(Direction~Lag1+Lag2, data=Weekly, family="binomial",subset=train)
> logistic_probs = predict(logistic_model3,Weekly.2009, type = "response")
> logistic_pred = rep("Down", 104)
> logistic_pred[logistic_probs > 0.5] = "Up"
> table(logistic_pred, Direction.2009)
             Direction.2009
logistic_pred Down Up
         Down    7  8
         Up     36 53
> mean(logistic_pred==Direction.2009)
[1] 0.5769231
```

```
> lda.fit1 = lda(Direction ~ Lag2:Lag1, data = Weekly, subset = train)
> lda.pred = predict(lda.fit1,Weekly.2009)
> table(lda.pred$class, Direction.2009)
      Direction.2009
       Down Up
  Down    0  1
  Up     43 60
> mean(lda.pred$class == Direction.2009)
[1] 0.5769231
```

```
> lda.fit2 = lda(Direction ~ Lag2:Volume, data = Weekly, subset = train)
> lda.pred = predict(lda.fit2, Weekly.2009)
> table(lda.pred$class, Direction.2009)
      Direction.2009
       Down Up
  Down    8  6
  Up     35 55
> mean(lda.pred$class == Direction.2009)
[1] 0.6057692
```

```
> lda.fit3 = lda(Direction ~ Lag1+Lag2, data = Weekly, subset = train)
> lda.pred = predict(lda.fit3, Weekly.2009)
> table(lda.pred$class, Direction.2009)
      Direction.2009
       Down Up
  Down    7  8
  Up     36 53
> mean(lda.pred$class == Direction.2009)
[1] 0.5769231
```

```
> qda.fit1 = qda(Direction ~ Lag2:Lag1, data = Weekly, subset = train)
> qda.pred = predict(qda.fit1, Weekly.2009)$class
> table(qda.pred, Direction.2009)
        Direction.2009
qda.pred Down Up
    Down   16 32
    Up     27 29
> mean(qda.pred == Direction.2009)
[1] 0.4326923
> qda.fit2 = qda(Direction ~ Lag2:Volume, data = Weekly, subset = train)
> qda.pred = predict(qda.fit2, Weekly.2009)$class
> table(qda.pred, Direction.2009)
        Direction.2009
qda.pred Down Up
    Down   24 28
    Up     19 33
> mean(qda.pred == Direction.2009)
[1] 0.5480769
```

```
> qda.fit3 = qda(Direction ~ Lag1+Lag2, data = Weekly, subset = train)
> qda.pred = predict(qda.fit3, Weekly.2009)$class
> table(qda.pred, Direction.2009)
        Direction.2009
qda.pred Down Up
    Down    7 10
    Up     36 51
> train.X = cbind(Lag2[train])
> test.X = cbind(Lag2[!train])
> train.Direction = Direction[train]
> set.seed(1)
> knn.pred = knn(train.X, test.X, train.Direction, k = 20)
> table(knn.pred, Direction.2009)
        Direction.2009
knn.pred Down Up
    Down   21 21
    Up     22 40
> mean(knn.pred == Direction.2009)
[1] 0.5865385
> train.X = cbind(Lag2[train])
> test.X = cbind(Lag2[!train])
> train.Direction = Direction[train]
> set.seed(1)
> knn.pred = knn(train.X, test.X, train.Direction, k = 100)
> table(knn.pred, Direction.2009)
        Direction.2009
knn.pred Down Up
    Down   10 11
    Up     33 50
> mean(knn.pred == Direction.2009)
[1] 0.5769231
> train.X = cbind(Lag2[train])
> test.X = cbind(Lag2[!train])
> train.Direction = Direction[train]
> set.seed(1)
> knn.pred = knn(train.X, test.X, train.Direction, k = 8)
> table(knn.pred, Direction.2009)
        Direction.2009
knn.pred Down Up
    Down   15 21
    Up     28 40
> mean(knn.pred == Direction.2009)
[1] 0.5288462
```

From the analysis, we can see that the original settings for logistic regression, LDA and QDA is the best.

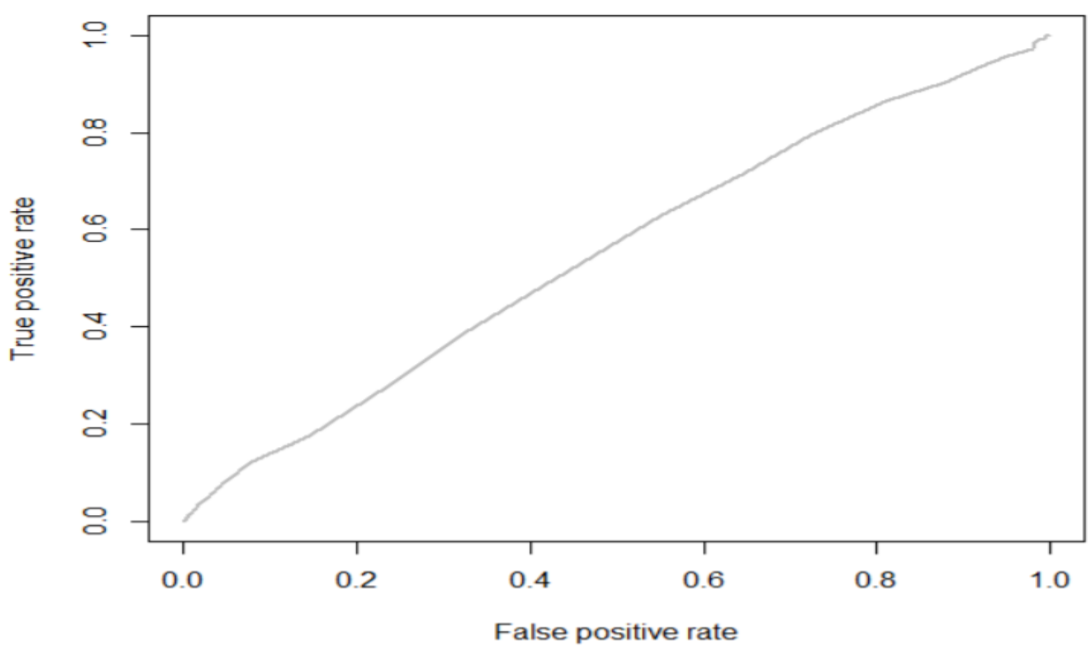
The only exception is that of KNN where the prediction accuracy increases when we increase the value of K.

Problem 2
Perform ROC analysis and present the results for logistic regression and LDA used for the best model chosen in Question 1(i).

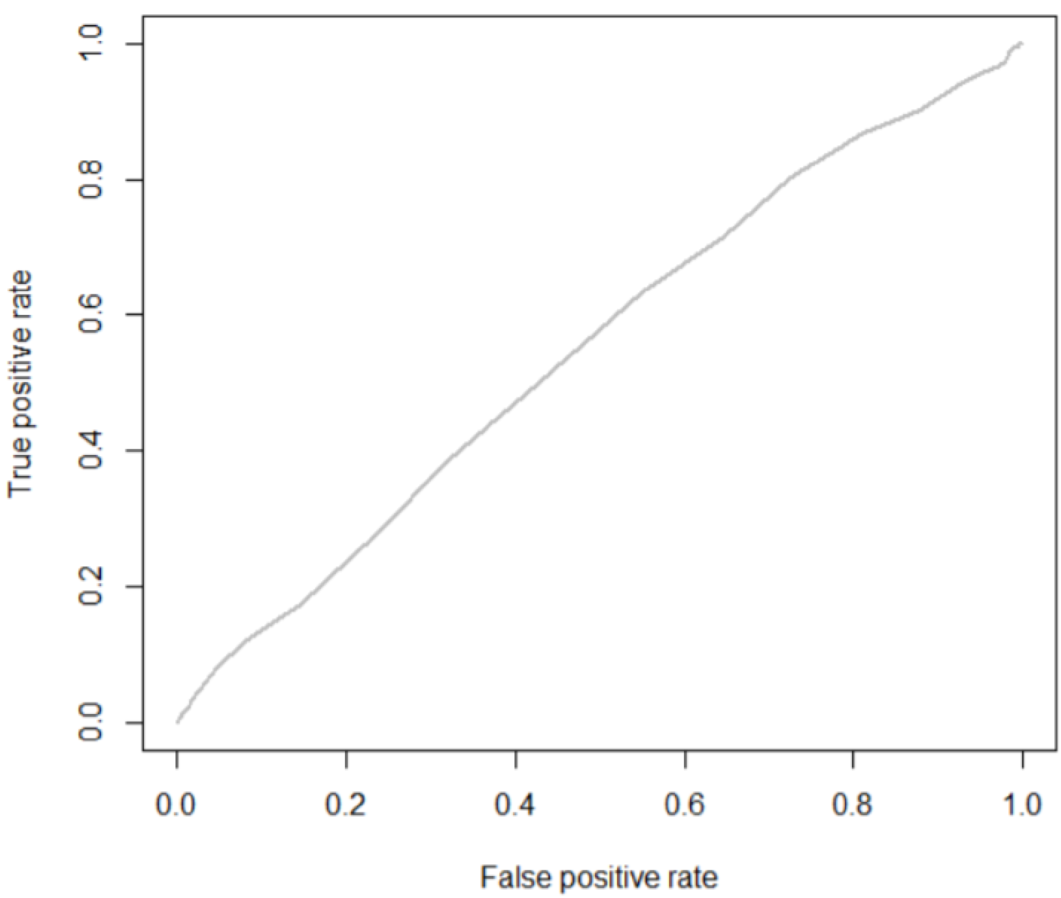
Ans 2.

```
> LR.fit= glm(Direction~Lag1+Lag2,family=binomial,data=Weekly)
> LR.pred= predict(LR.fit,type="response")
```

```
> roc.curve=function(s,print=FALSE){
+   Ps=(LR.pred>s)*1
+   FP=sum((Ps==1)*(Direction=="Down"))/sum(Direction=="Down")
+   TP=sum((Ps==1)*(Direction=="Up"))/sum(Direction=="Up")
+   if(print==TRUE){
+     print(table(Observed=Direction,Predicted=Ps))
+   }
+   vect=c(FP,TP)
+   names(vect)=c("FPR","TPR")
+   return(vect)
+ }
> threshold=0.5
> roc.curve(threshold,print=TRUE)
        Predicted
Observed   0   1
    Down  38 446
    Up    38 567
      FPR       TPR
0.9214876 0.9371901
> ROC.curve=Vectorize(roc.curve)
> M.ROC=ROC.curve(seq(0,1,by=0.01))
> plot(M.ROC[1,],M.ROC[2,],col="grey",lwd=2,type="l",xlab="False positive rate",ylab="Tr
ue positive rate")
```



```
> LDA.fit= lda(Direction~Lag1+Lag2,data=Weekly)
> LDA.pred0 = predict(LDA.fit,type="response")
> LDA.pred= LDA.pred0$posterior[,2]
> roc.curve=function(s,print=FALSE){
+   Ps=(LDA.pred>s)*1
+   FP=sum((Ps==1)*(Direction=="Down"))/sum(Direction=="Down")
+   TP=sum((Ps==1)*(Direction=="Up"))/sum(Direction=="Up")
+   if(print==TRUE){
+     print(table(Observed=Direction,Predicted=Ps))
+   }
+   vect=c(FP,TP)
+   names(vect)=c("FPR","TPR")
+   return(vect)
+ }
> threshold=0.5
> roc.curve(threshold,print=TRUE)
        Predicted
Observed   0   1
    Down  37 447
    Up    37 568
      FPR       TPR
0.9235537 0.9388430
> ROC.curve=Vectorize(roc.curve)
> M.ROC=ROC.curve(seq(0,1,by=0.01))
> plot(M.ROC[1,],M.ROC[2,],col="grey",lwd=2,type="l",xlab="False positive rate",ylab="Tr
ue positive rate")
```



Problem 3

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 that 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.

(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.

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

(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?

(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?

(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?

(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?

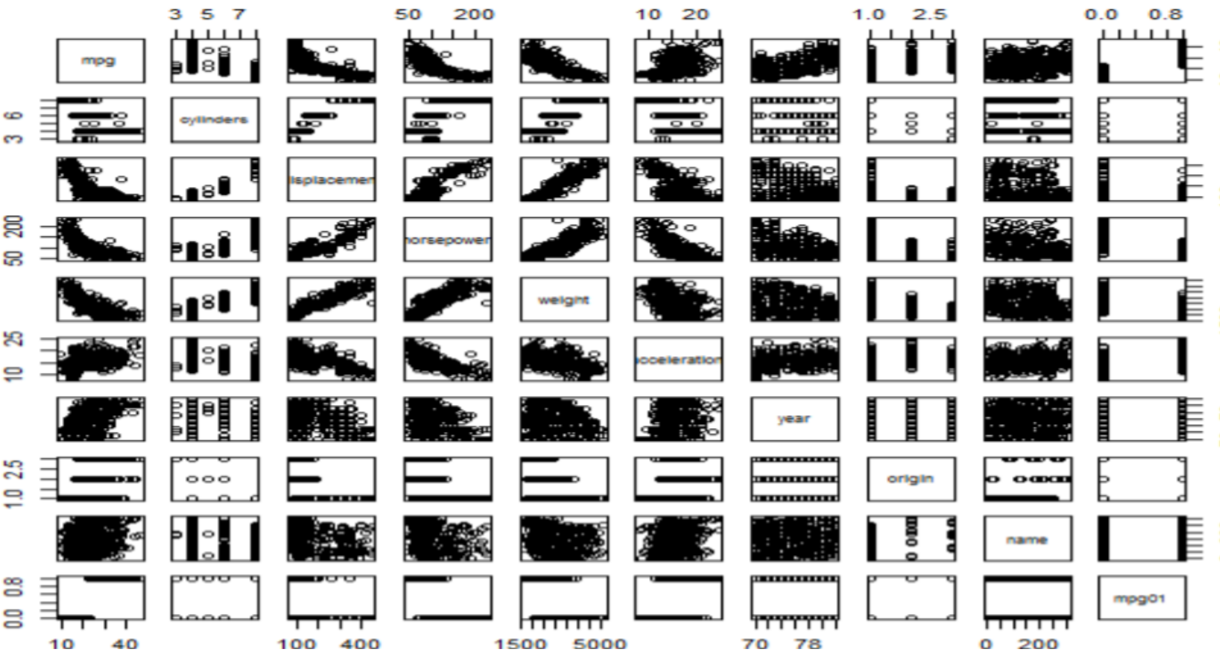
Ans 3.

(a).

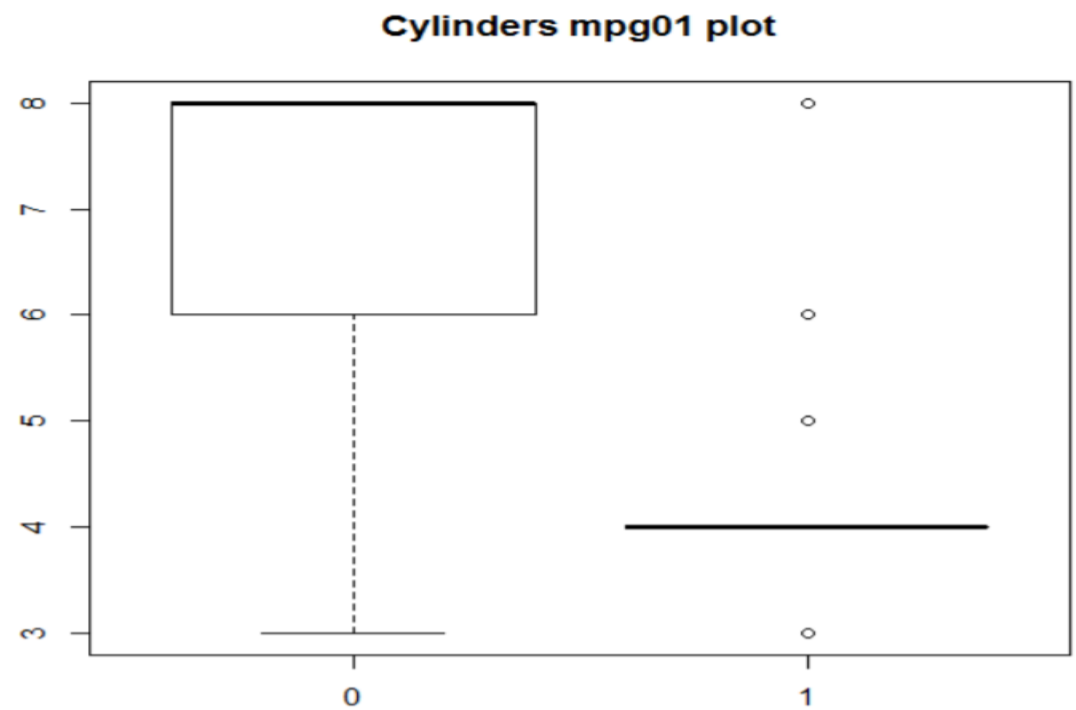
```
> library(ISLR)
> attach(Auto)
> mpg01 = rep(0, length(mpg))
> mpg01[mpg > median(mpg)] = 1
> Auto = data.frame(Auto, mpg01)
```

(b).

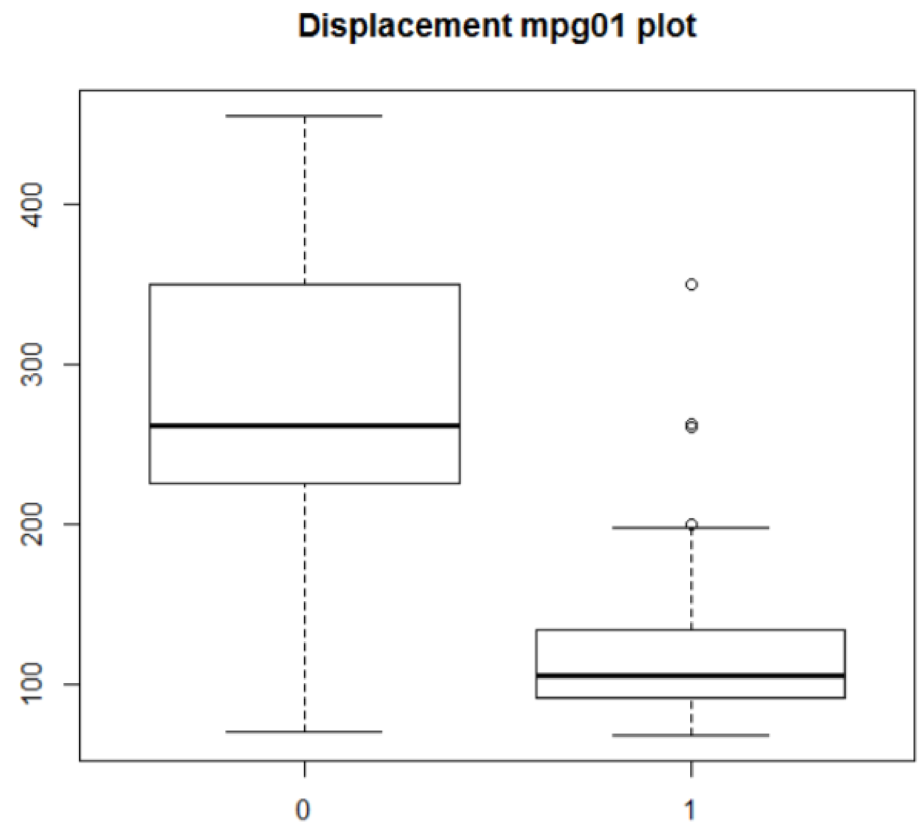
```
> cor(Auto[, -9])
                    mpg  cylinders displacement horsepower     weight acceleration
mpg           1.0000000 -0.7776175   -0.8051269 -0.7784268 -0.8322442    0.4233285
cylinders    -0.7776175  1.0000000    0.9508233  0.8429834  0.8975273   -0.5046834
displacement -0.8051269  0.9508233    1.0000000  0.8972570  0.9329944   -0.5438005
horsepower   -0.7784268  0.8429834    0.8972570  1.0000000  0.8645377   -0.6891955
weight       -0.8322442  0.8975273    0.9329944  0.8645377  1.0000000   -0.4168392
acceleration  0.4233285 -0.5046834   -0.5438005 -0.6891955 -0.4168392    1.0000000
year          0.5805410 -0.3456474   -0.3698552 -0.4163615 -0.3091199    0.2903161
origin        0.5652088 -0.5689316   -0.6145351 -0.4551715 -0.5850054    0.2127458
mpg01         0.8369392 -0.7591939   -0.7534766 -0.6670526 -0.7577566    0.3468215
                   year     origin      mpg01
mpg           0.5805410  0.5652088  0.8369392
cylinders    -0.3456474 -0.5689316 -0.7591939
displacement -0.3698552 -0.6145351 -0.7534766
horsepower   -0.4163615 -0.4551715 -0.6670526
weight       -0.3091199 -0.5850054 -0.7577566
acceleration  0.2903161  0.2127458  0.3468215
year          1.0000000  0.1815277  0.4299042
origin        0.1815277  1.0000000  0.5136984
mpg01         0.4299042  0.5136984  1.0000000
> pairs(Auto)
```



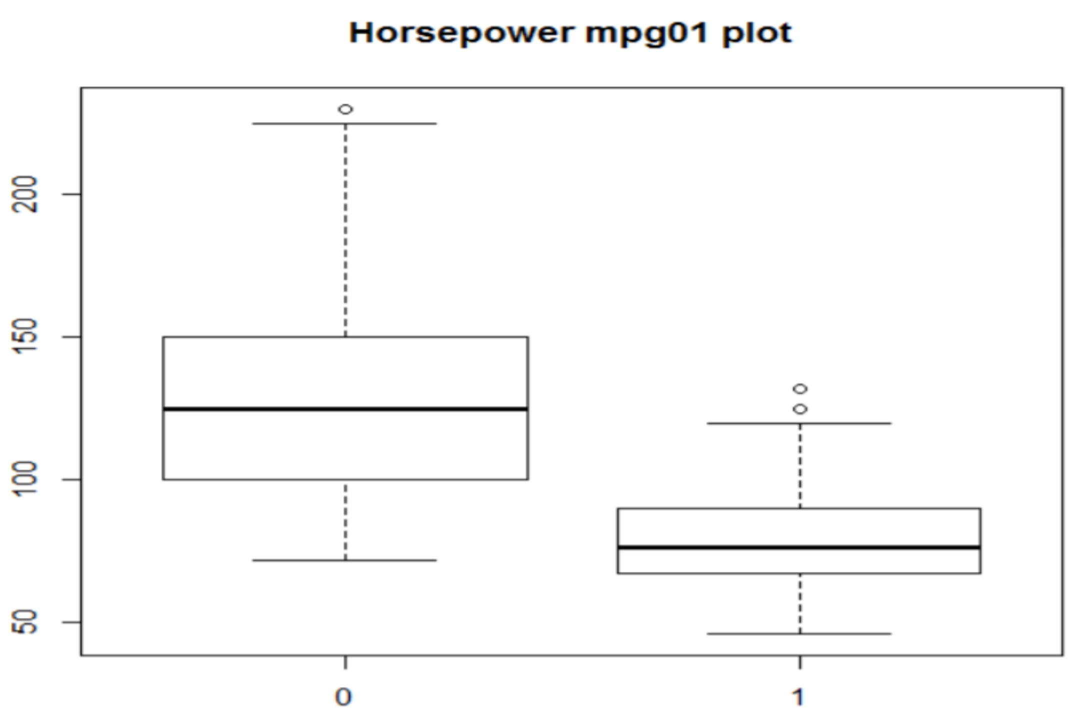

```
> boxplot(cylinders ~ mpg01, data = Auto, main = "Cylinders mpg01 plot")
```



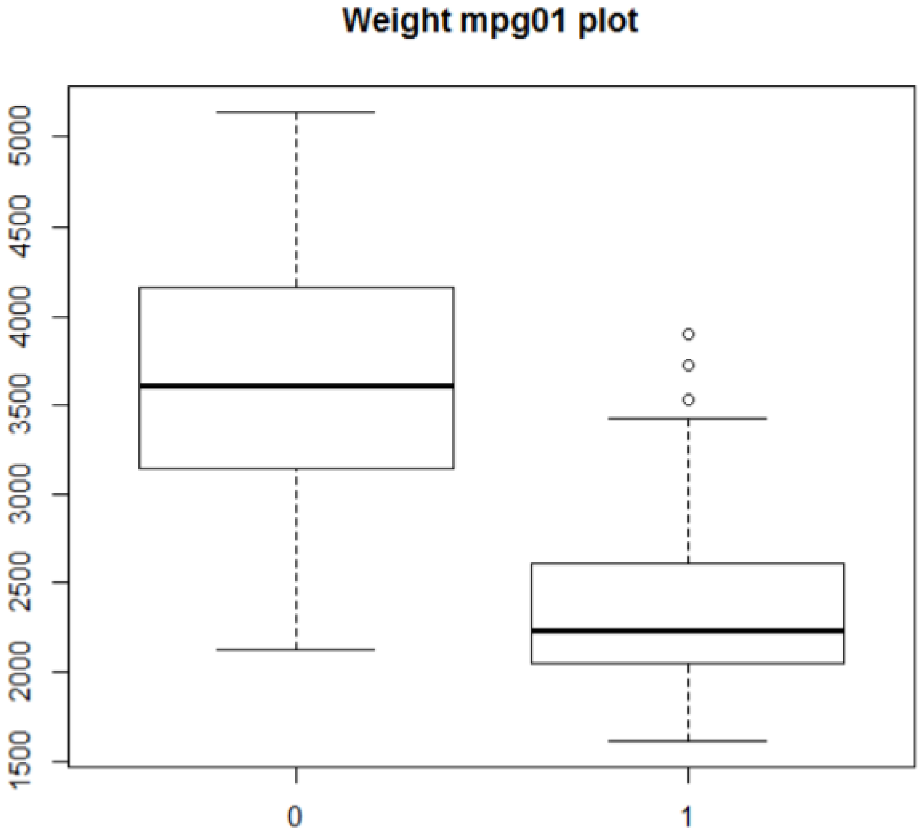
```
> boxplot(displacement ~ mpg01, data = Auto, main = "Displacement mpg01 plot")
```



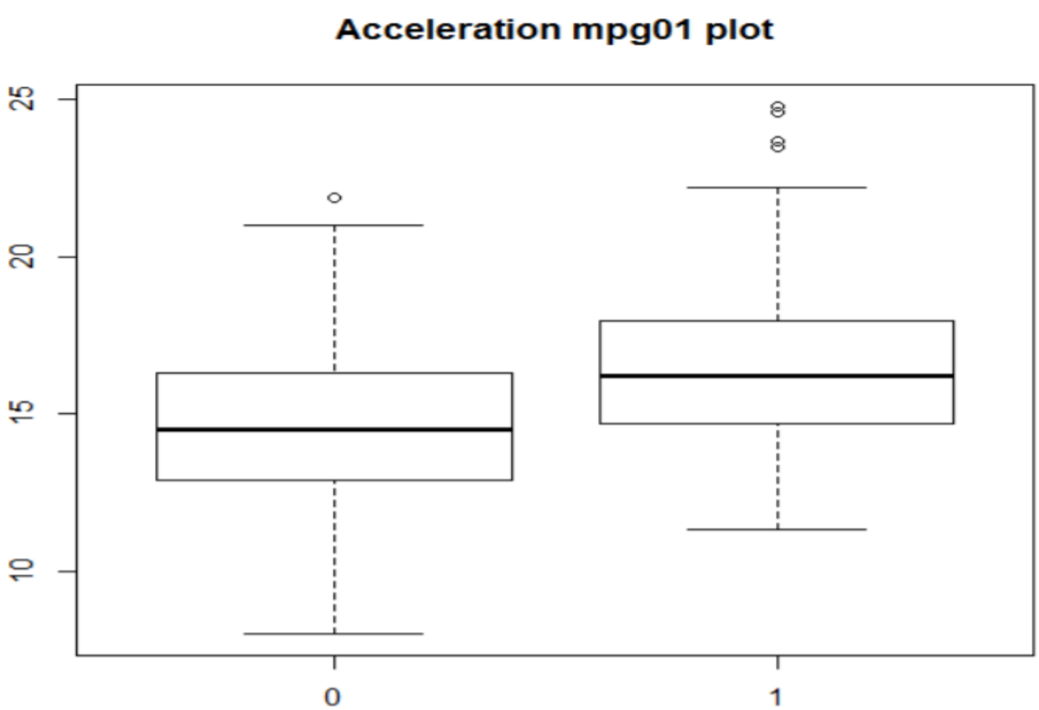

```
> boxplot(horsepower ~ mpg01, data = Auto, main = "Horsepower mpg01 plot")
```



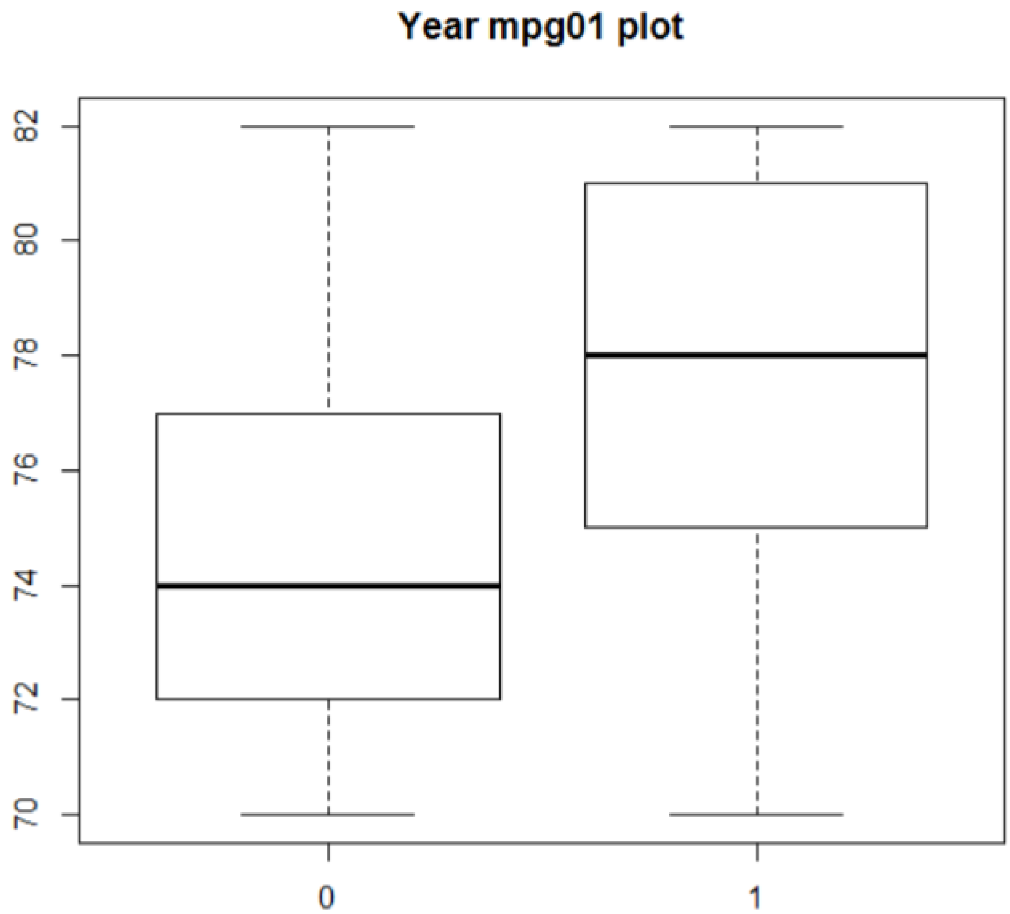
```
> boxplot(weight ~ mpg01, data = Auto, main = "Weight mpg01 plot")
```




```
> boxplot(acceleration ~ mpg01, data = Auto, main = "Acceleration mpg01 plot")
```



```
> boxplot(year ~ mpg01, data = Auto, main = "Year mpg01 plot")
```



By looking at the results of the cor() function, scatterplot, box plots and pairs plot, we can conclude that mpg01 has correlation with cylinders, displacement, horsepower and weight.

(c).

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

(d).

```
> fit.lda = lda(mpg01 ~ cylinders + weight + displacement + horsepower, data = Auto, sub
set = train)
> 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 testing error in case of LDA was found to be 12.637%

(e).

```
> fit.qda = qda(mpg01 ~ cylinders + weight + displacement + horsepower, data = Auto, sub
set = train)
> 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 testing error of QDA was found to be 13.19%

(f).

```
> fit.glm = glm(mpg01 ~ cylinders + weight + displacement + horsepower, data = Auto, fam
ily = binomial, subset = train)
> 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 testing error in case of logistic regression was found to be 12.09%