Apurva Shrivastava
925009508
ISEN 613 - 600

Assignment 3 – Classification

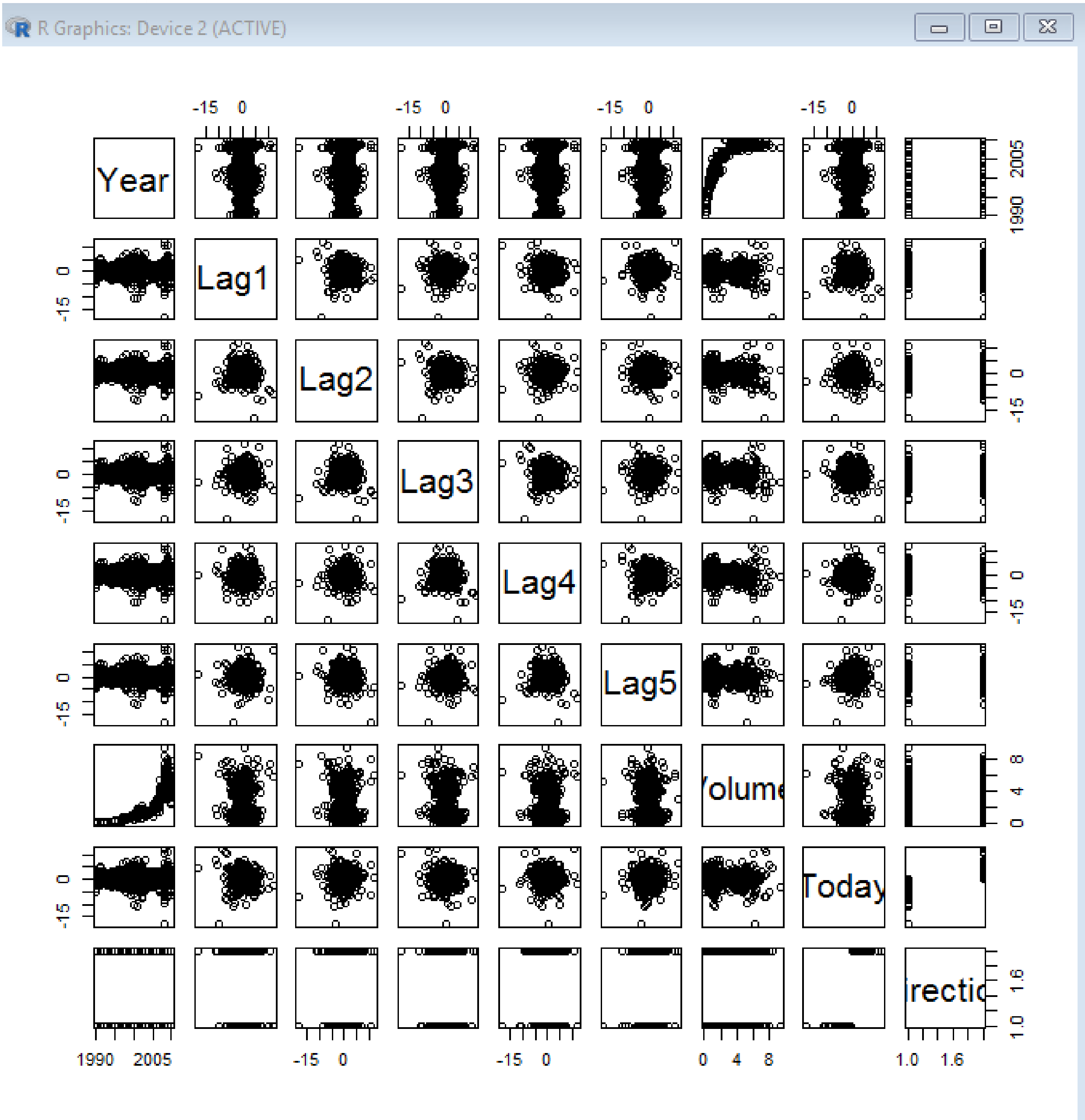
Problem 1

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?

```
> library(ISLR)
> attach(weekly)
Error in attach(weekly) : object 'weekly' not found
> 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
> head(Weekly)
  Year   Lag1   Lag2   Lag3   Lag4   Lag5    Volume  Today Direction
1 1990  0.816  1.572 -3.936 -0.229 -3.484 0.1549760 -0.270      Down
2 1990 -0.270  0.816  1.572 -3.936 -0.229 0.1485740 -2.576      Down
3 1990 -2.576 -0.270  0.816  1.572 -3.936 0.1598375  3.514        Up
4 1990  3.514 -2.576 -0.270  0.816  1.572 0.1616300  0.712        Up
5 1990  0.712  3.514 -2.576 -0.270  0.816 0.1537280  1.178        Up
6 1990  1.178  0.712  3.514 -2.576 -0.270 0.1544440 -1.372      Down
> str(Weekly)
'data.frame':   1089 obs. of  9 variables:
 $ Year     : num  1990 1990 1990 1990 1990 1990 1990 1990 1990 1990 ...
 $ Lag1     : num  0.816 -0.27 -2.576 3.514 0.712 ...
 $ Lag2     : num  1.572 0.816 -0.27 -2.576 3.514 ...
 $ Lag3     : num  -3.936 1.572 0.816 -0.27 -2.576 ...
 $ Lag4     : num  -0.229 -3.936 1.572 0.816 -0.27 ...
 $ Lag5     : num  -3.484 -0.229 -3.936 1.572 0.816 ...
 $ Volume   : num  0.155 0.149 0.16 0.162 0.154 ...
 $ Today    : num  -0.27 -2.576 3.514 0.712 1.178 ...
 $ Direction: Factor w/ 2 levels "Down","Up": 1 1 2 2 2 1 2 2 2 1 ...
> pairs(Weekly)
> 
```

Apurva Shrivastava
925009508
ISEN 613 - 600



By plotting pair(Weekly), we observe that Year and Volume have a strong correlation. Their correlation can be measure by using the cor() command .

Apurva Shrivastava
925009508
ISEN 613 - 600

```
> cor(Weekly[,-9])
              Year         Lag1         Lag2        Lag3         Lag4         Lag5      Volume        Today
Year    1.00000000 -0.032289274 -0.03339001 -0.03000649 -0.031127923 -0.030519101  0.84194162 -0.032459894
Lag1   -0.03228927  1.000000000 -0.07485305  0.05863568 -0.071273876 -0.008183096 -0.06495131 -0.075031842
Lag2   -0.03339001 -0.074853051  1.00000000 -0.07572091  0.058381535 -0.072499482 -0.08551314  0.059166717
Lag3   -0.03000649  0.058635682 -0.07572091  1.00000000 -0.075395865  0.060657175 -0.06928771 -0.071243639
Lag4   -0.03112792 -0.071273876  0.05838153 -0.07539587  1.000000000 -0.075675027 -0.06107462 -0.007825873
Lag5   -0.03051910 -0.008183096 -0.07249948  0.06065717 -0.075675027  1.000000000 -0.05851741  0.011012698
Volume  0.84194162 -0.064951313 -0.08551314 -0.06928771 -0.061074617 -0.058517414  1.00000000 -0.033077783
Today  -0.03245989 -0.075031842  0.05916672 -0.07124364 -0.007825873  0.011012698 -0.03307778  1.000000000
>
```

We observer here that the correlation coefficient is 0.842 for year and volume.

(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?

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

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
```

If we observe the p-values, we see that only predictor Lag2 is statistically significant as its p-value is less than 0.05 (0.0296). The other predictors have high p-value, hence are not significant.

(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?)

Apurva Shrivastava
925009508
ISEN 613 - 600

```
>
> logistic.probs=predict(logistic.fit,type="response")
> logistic.pred=rep("Down",nrow(Weekly))
> logistic.pred[logistic.probs>0.5]="Up"
>
> #Confusion matrix
> table(logistic.pred,Direction)
             Direction
logistic.pred Down  Up
         Down   54  48
         Up    430 557
>
> #Accuracy
> mean(logistic.pred==Direction)
[1] 0.5610652
>
> #Error
> mean(logistic.pred!=Direction)
[1] 0.4389348
>
```

From the confusion matrix, we can compute the following performance measures as below:

Accuracy: (54+557/54+557+48+430) = 0.5610652, meaning we have an accuracy of 56.10652% on the training data for predicted values.

Error Rate: (48+ 430/ 54+557+48+430) = 0.4389348, meaning we have an error rate of 43.89348% on the training data (training error rate) for predicted values.

Sensitivity: (557/557+48) = 0.92066, meaning we have a sensitivity of 92.066% on the training data for predicted values. This means that when the prediction is up, the model is correct 92.066% of the times.

Specificity: (54/54+430) = 0.1115702, meaning we have a specificity of 11.15702% on the training data for predicted values. This means that when the prediction is down, the model is correct 11.15702% of the times.

(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).

Apurva Shrivastava
925009508
ISEN 613 - 600

```
> train=(Year<=2008)
> test=Weekly[!train,]
> Direction.test=Direction[!train]
>
> logistic.fit1d=glm(Direction~Lag2,data=Weekly,family=binomial,subset=train)
> summary(logistic.fit1d)

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

Deviance Residuals:
   Min      1Q  Median      3Q     Max
-1.536  -1.264   1.021   1.091   1.368

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  0.20326    0.06428   3.162  0.00157 **
Lag2         0.05810    0.02870   2.024  0.04298 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1354.7  on 984  degrees of freedom
Residual deviance: 1350.5  on 983  degrees of freedom
AIC: 1354.5

Number of Fisher Scoring iterations: 4
```

```
> logistic.probs1d=predict(logistic.fit1d,test,type="response")
> logistic.pred1d=rep("Down", 104)
> logistic.pred1d[logistic.probs1d>0.5] = "Up"
> table(logistic.pred1d,Direction.test)
                Direction.test
logistic.pred1d Down Up
           Down    9  5
           Up     34 56
> mean(logistic.pred1d==Direction.test) #Accuracy rate
[1] 0.625
> mean(logistic.pred1d!=Direction.test) #Error rate
[1] 0.375
> |
```

Accuracy Rate = (TP + TN) / Total observations = (9+56)/(9+56+34+5) = 62.5%
Error Rate = 1 – Accuracy rate = 100 – 62.5 = 37.5%
Sensitivity = TP/Total Positive = 56/(56+5) = 91.8%
Specificity = TN/Total Negative = 9/(9+34) = 20.9%

(e) Repeat (d) using LDA

Apurva Shrivastava
925009508
ISEN 613 - 600

```
>
> library(MASS)
> lda.fit=lda(Direction~Lag2, data=Weekly,subset=train)
> summary(lda.fit)
        Length Class  Mode
prior   2      -none- numeric
counts  2      -none- numeric
means   2      -none- numeric
scaling 1      -none- numeric
lev     2      -none- character
svd     1      -none- numeric
N       1      -none- numeric
call    4      -none- call
terms   3      terms  call
xlevels 0      -none- list
> lda_test=Weekly[!train,]
> lda_direction=Direction[!train]
> lda.pred = predict(lda.fit,lda_test)
> table(lda.pred$class,lda_direction)
      lda_direction
       Down Up
  Down    9  5
  Up     34 56
> mean(lda.pred$class==lda_direction) #Accuracy rate
[1] 0.625
> mean(lda.pred$class!=lda_direction) #Error rate
[1] 0.375
>
```

Apurva Shrivastava
925009508
ISEN 613 - 600

(f) Repeat (d) using QDA

```
> qda.fit=qda(Direction~Lag2, data=Weekly,subset=train)
> summary(qda.fit)
        Length Class  Mode
prior   2      -none- numeric
counts  2      -none- numeric
means   2      -none- numeric
scaling 2      -none- numeric
ldet    2      -none- numeric
lev     2      -none- character
N       1      -none- numeric
call    4      -none- call
terms   3      terms  call
xlevels 0      -none- list
> qda_test=Weekly[!train,]
> qda_direction=Weekly$Direction[!train]
> qda.pred = predict(qda.fit,lda_test)
> names(qda.pred)
[1] "class"     "posterior"
> qda.class = qda.pred$class
> table(qda.pred$class,qda_direction)
      qda_direction
       Down Up
  Down    0  0
  Up     43 61
> mean(qda.pred$class==qda_direction) #Accuracy rate
[1] 0.5865385
> mean(qda.pred$class!=qda_direction) #Error rate
[1] 0.4134615
>
> |
```

Apurva Shrivastava
925009508
ISEN 613 - 600

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

```
> library(class)
> #Training data (predictors)
> knn_train=as.matrix(Weekly$Lag2[train])
>
> #Test data (predictors)
> knn_test=as.matrix(Lag2[!train])
>
> #Training data (response)
> knn_train_Direction=Direction[train]
> set.seed(1)
> knn.pred=knn(knn_train,knn_test,knn_train_Direction,k=1)
> summary(knn.pred)
Down   Up
  51   53
> knn_direction=Weekly$Direction[!train]
> table(knn.pred,knn_direction)
        knn_direction
knn.pred Down Up
    Down   21 30
    Up     22 31
> mean(knn.pred==knn_direction) #Accuracy rate
[1] 0.5
> mean(knn.pred!=knn_direction) #Error rate
[1] 0.5
>
> |
```

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

From the above four, the accuracy rates are as follows:

Logistic Regression: 62.50 %
LDA: 62.50 %
QDA: 58.65 %
KNN: 50 %

Hence, we can see that Logistic regression and LDA have the same accuracy.

Apurva Shrivastava
925009508
ISEN 613 - 600

(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.

Logistic Regression

Predictor: (Lag 2 + Lag 4)

```
> logmodel_combined <- glm(Direction~Lag2+Lag4, data=Weekly, family= binomial, subset= training)
> summary(logmodel_combined)

Call:
glm(formula = Direction ~ Lag2 + Lag4, family = binomial, data = Weekly,
    subset = training)

Deviance Residuals:
   Min       1Q  Median       3Q      Max
-1.507  -1.261   1.021   1.092   1.346

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  0.20536    0.06439   3.189  0.00143 **
Lag2         0.05942    0.02881   2.062  0.03916 *
Lag4        -0.01740    0.02846  -0.611  0.54092
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1354.7  on 984  degrees of freedom
Residual deviance: 1350.2  on 982  degrees of freedom
AIC: 1356.2

Number of Fisher Scoring iterations: 4

> prbs_combined <- predict(logmodel_combined, Weekly.20092010,  type="response")
> pred.glmcombined[prbs_combined>0.5]<-"Up"
> prbs_combined <- predict(logmodel_combined, Weekly.20092010,  type="response")
> pred.glmcombined <- rep("Down", length(prbs_combined))
> pred.glmcombined[prbs_combined>0.5]<-"Up"
> table(pred.glmcombined, Direction.20092010)
                Direction.20092010
pred.glmcombined Down Up
            Down    8  4
            Up     35 57
>
> mean(pred.glmcombined==Direction.20092010)
[1] 0.625
```

Predictor: (Lag 1 + Lag 2)

Apurva Shrivastava
925009508
ISEN 613 - 600

```
> logmodel_combined2 <- glm(Direction~Lag1+Lag2, data=Weekly, family= binomial, subset= training)
> summary(logmodel_combined2)

Call:
glm(formula = Direction ~ Lag1 + Lag2, family = binomial, data = Weekly,
    subset = training)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.6149  -1.2565   0.9989   1.0875   1.5330

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  0.21109    0.06456   3.269  0.00108 **
Lag1        -0.05421    0.02886  -1.878  0.06034 .
Lag2         0.05384    0.02905   1.854  0.06379 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1354.7  on 984  degrees of freedom
Residual deviance: 1347.0  on 982  degrees of freedom
AIC: 1353

Number of Fisher Scoring iterations: 4

> prbs_combined2 <- predict(logmodel_combined2, Weekly.20092010,  type="response")
> pred.glmcombined2 <- rep("Down", length(prbs_combined2))
> pred.glmcombined2[prbs_combined2>0.5]<-"Up"
> table(pred.glmcombined2, Direction.20092010)
                 Direction.20092010
pred.glmcombined2 Down Up
             Down    7  8
             Up     36 53
> 
> mean(pred.glmcombined2==Direction.20092010)
[1] 0.5769231
```

LDA

Predictor: Lag2 + Lag 4

```
> ldamodel_combined <- lda(Direction~Lag2+Lag4, data= Weekly, subset= training)
> ldamodel_combined
Call:
lda(Direction ~ Lag2 + Lag4, data = Weekly, subset = training)

Prior probabilities of groups:
     Down        Up
0.4477157 0.5522843

Group means:
            Lag2       Lag4
Down -0.03568254 0.15925624
Up    0.26036581 0.09220956

Coefficients of linear discriminants:
            LD1
Lag2  0.4320682
Lag4 -0.1258311
```

Apurva Shrivastava
925009508
ISEN 613 - 600

```
> pred.ldacombined <- predict(ldamodel_combined, Weekly.20092010)
> lda_class_combined <- pred.ldacombined$class
> table(lda_class_combined, Direction.20092010)
                  Direction.20092010
lda_class_combined Down Up
              Down    8  4
              Up     35 57
> mean(lda_class_combined==Direction.20092010)
[1] 0.625
```

Predictor: (Lag1 + Lag2)

```
> ldamodel_combined2 <- lda(Direction~Lag1+Lag2, data= Weekly, subset= training)
> ldamodel_combined2
Call:
lda(Direction ~ Lag1 + Lag2, data = Weekly, subset = training)

Prior probabilities of groups:
     Down        Up
0.4477157 0.5522843

Group means:
             Lag1        Lag2
Down  0.289444444 -0.03568254
Up   -0.009213235  0.26036581

Coefficients of linear discriminants:
            LD1
Lag1 -0.3013148
Lag2  0.2982579
> pred.ldacombined2 <- predict(ldamodel_combined2, Weekly20092010)
Error in is.data.frame(data) : object 'Weekly20092010' not found
> pred.ldacombined2 <- predict(ldamodel_combined2, Weekly.20092010)
> lda_class_combined2 <- pred.ldacombined2$class
> table(lda_class_combined2, Direction.20092010)
                   Direction.20092010
lda_class_combined2 Down Up
               Down    7  8
               Up     36 53
> |
> mean(lda_class_combined2==Direction.20092010)
[1] 0.5769231
```

QDA

Predictor: Log(Lag2)

Apurva Shrivastava
925009508
ISEN 613 - 600

```
> qdalog <- qda(Direction~log(Lag2), data= Weekly, subset= training)
Warning message:
In log(Lag2) : NaNs produced
> qdalog
Call:
qda(Direction ~ log(Lag2), data = Weekly, subset = training)

Prior probabilities of groups:
     Down        Up
0.4422018 0.5577982

Group means:
       log(Lag2)
Down -0.1014035
Up    0.1419702
> pred.qdalog <- predict(qdalog, Weekly.20092010)
There were 45 warnings (use warnings() to see them)
> qda_class_log <- pred.qdalog$class
> table(qda_class_log, Direction.20092010)
             Direction.20092010
qda_class_log Down Up
         Down    1  6
         Up     22 31
>
```

KNN

K =10

```
> pred.knn10 <- knn(training.x, test.x, training.Direction, k=10)
> table( pred.knn10, Direction.20092010)
          Direction.20092010
pred.knn10 Down Up
      Down   17 18
      Up     26 43
>
> mean(pred.knn10==Direction.20092010)
[1] 0.5769231
```

K = 100

```
> pred.knn100 <- knn(training.x, test.x, training.Direction, k=100)
> table( pred.knn100, Direction.20092010)
           Direction.20092010
pred.knn100 Down Up
       Down    9 12
       Up     34 49
> mean(pred.knn100==Direction.20092010)
[1] 0.5576923
```

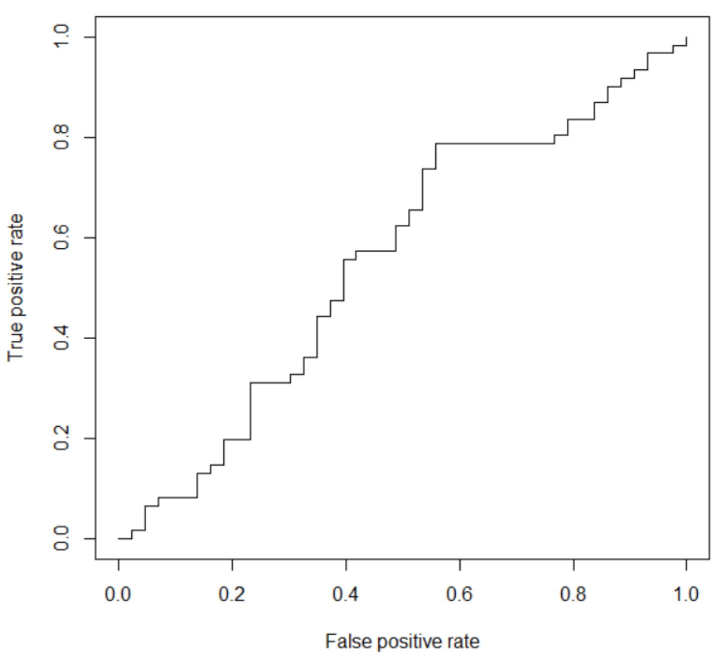
Apurva Shrivastava
925009508
ISEN 613 - 600

Problem 2

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

Logisitc Regression:

```
> library(ROCR)
Loading required package: gplots

Attaching package: 'gplots'

The following object is masked from 'package:stats':

    lowess

Warning messages:
1: package 'ROCR' was built under R version 3.3.3
2: package 'gplots' was built under R version 3.3.3
> glm.probs=predict(logit.fit,Weekly.test,type="response")
> pr = prediction(glm.probs, Weekly.test$Direction)
> prf = performance(pr, measure = "tpr", x.measure = "fpr")
> plot(prf)
> auc <- performance(pr, measure = "auc")
> auc <- auc@y.values[[1]]
> auc
[1] 0.557377
```

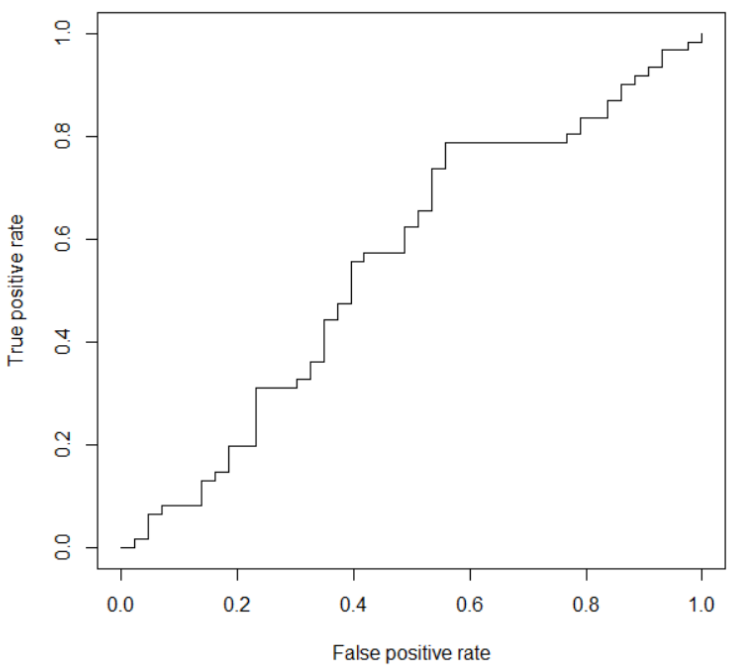


Apurva Shrivastava
925009508
ISEN 613 - 600

The Area Under the Curve(AUC) is very low ~ 0.5, This means that the model has very low predictive ability.

LDA:

```
> lda.pred = predict(fit.lda, Weekly.test)
> names(lda.pred)
[1] "class"     "posterior" "x"
> pr = prediction(lda.pred$posterior[,2], Weekly.test$Direction)
> prf.lda= performance(pr, measure = "tpr", x.measure = "fpr")
> plot(prf.lda)
> auc = performance(pr, measure = "auc")
> auc = auc@y.values[[1]]
> auc
[1] 0.5566146
```



The AUC (Area under the curve) is still very low ~ 0.5 and is almost the same as logistic model. This is due to the fact that the performance measures are the same for both models.

Apurva Shrivastava
925009508
ISEN 613 - 600

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.

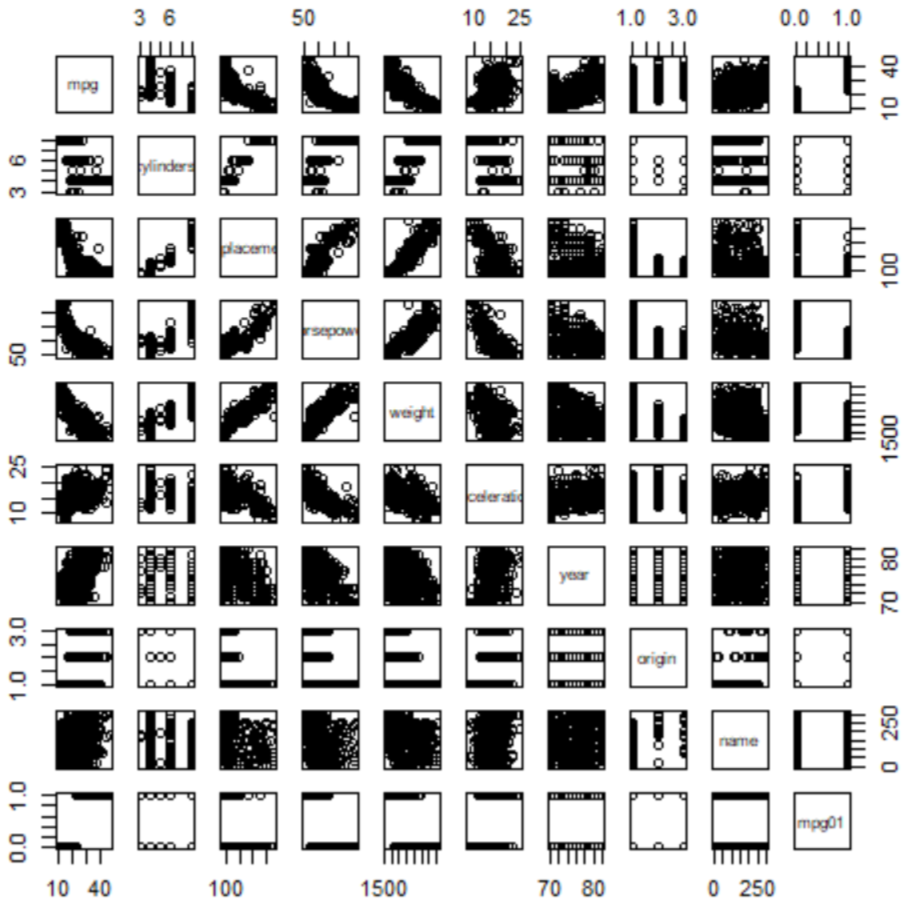
(a) Create a binary variable, mpg01, that contains a 1 if mpg contains a value above its median, and a 0 if mpg contains a value below its median. You can compute the median using the median () function. Note that you may find it helpful to use the data. frame() function to create a single data set containing both mpg01 and the other Auto variables.

```
> summary(Auto)
      mpg          cylinders      displacement     horsepower        weight
 Min.   : 9.00   Min.   :3.000   Min.   : 68.0   Min.   : 46.0   Min.   :1613
 1st Qu.:17.00   1st Qu.:4.000   1st Qu.:105.0   1st Qu.: 75.0   1st Qu.:2225
 Median :22.75   Median :4.000   Median :151.0   Median : 93.5   Median :2804
 Mean   :23.45   Mean   :5.472   Mean   :194.4   Mean   :104.5   Mean   :2978
 3rd Qu.:29.00   3rd Qu.:8.000   3rd Qu.:275.8   3rd Qu.:126.0   3rd Qu.:3615
 Max.   :46.60   Max.   :8.000   Max.   :455.0   Max.   :230.0   Max.   :5140

  acceleration        year           origin                      name
 Min.   : 8.00   Min.   :70.00   Min.   :1.000   amc matador       :  5
 1st Qu.:13.78   1st Qu.:73.00   1st Qu.:1.000   ford pinto        :  5
 Median :15.50   Median :76.00   Median :1.000   toyota corolla    :  5
 Mean   :15.54   Mean   :75.98   Mean   :1.577   amc gremlin       :  4
 3rd Qu.:17.02   3rd Qu.:79.00   3rd Qu.:2.000   amc hornet        :  4
 Max.   :24.80   Max.   :82.00   Max.   :3.000   chevrolet chevette:  4
                                                 (Other)           :365
> mpg01 <- rep(0, length(mpg))
> mpg01[mpg > median(mpg)] <- 1
> Auto1 <- data.frame(Auto, mpg01)
> summary(Auto1)
      mpg          cylinders      displacement     horsepower        weight
 Min.   : 9.00   Min.   :3.000   Min.   : 68.0   Min.   : 46.0   Min.   :1613
 1st Qu.:17.00   1st Qu.:4.000   1st Qu.:105.0   1st Qu.: 75.0   1st Qu.:2225
 Median :22.75   Median :4.000   Median :151.0   Median : 93.5   Median :2804
 Mean   :23.45   Mean   :5.472   Mean   :194.4   Mean   :104.5   Mean   :2978
 3rd Qu.:29.00   3rd Qu.:8.000   3rd Qu.:275.8   3rd Qu.:126.0   3rd Qu.:3615
 Max.   :46.60   Max.   :8.000   Max.   :455.0   Max.   :230.0   Max.   :5140

  acceleration        year           origin                      name
 Min.   : 8.00   Min.   :70.00   Min.   :1.000   amc matador       :  5
 1st Qu.:13.78   1st Qu.:73.00   1st Qu.:1.000   ford pinto        :  5
 Median :15.50   Median :76.00   Median :1.000   toyota corolla    :  5
 Mean   :15.54   Mean   :75.98   Mean   :1.577   amc gremlin       :  4
 3rd Qu.:17.02   3rd Qu.:79.00   3rd Qu.:2.000   amc hornet        :  4
 Max.   :24.80   Max.   :82.00   Max.   :3.000   chevrolet chevette:  4
                                                 (Other)           :365
     mpg01
 Min.   :0.0
 1st Qu.:0.0
 Median :0.5
 Mean   :0.5
 3rd Qu.:1.0
 Max.   :1.0
```

(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.

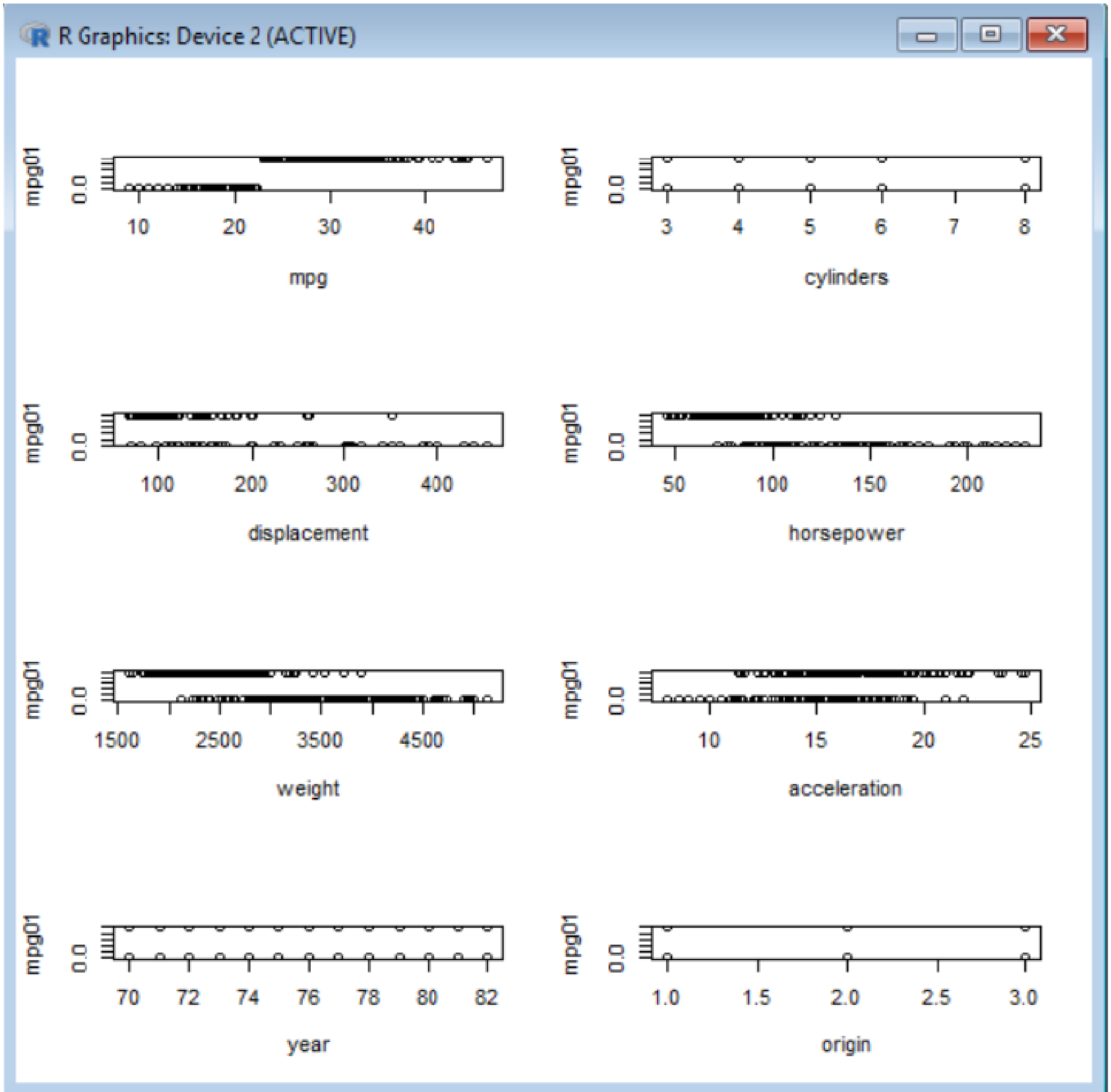
Apurva Shrivastava
925009508
ISEN 613 - 600

```
> pairs(Autol)
```




```
> attach(Auto)
> par(mfrow=c(4,2))
> plot(mpg01~mpg+cylinders+displacement+horsepower+weight+acceleration+year+origin)
```

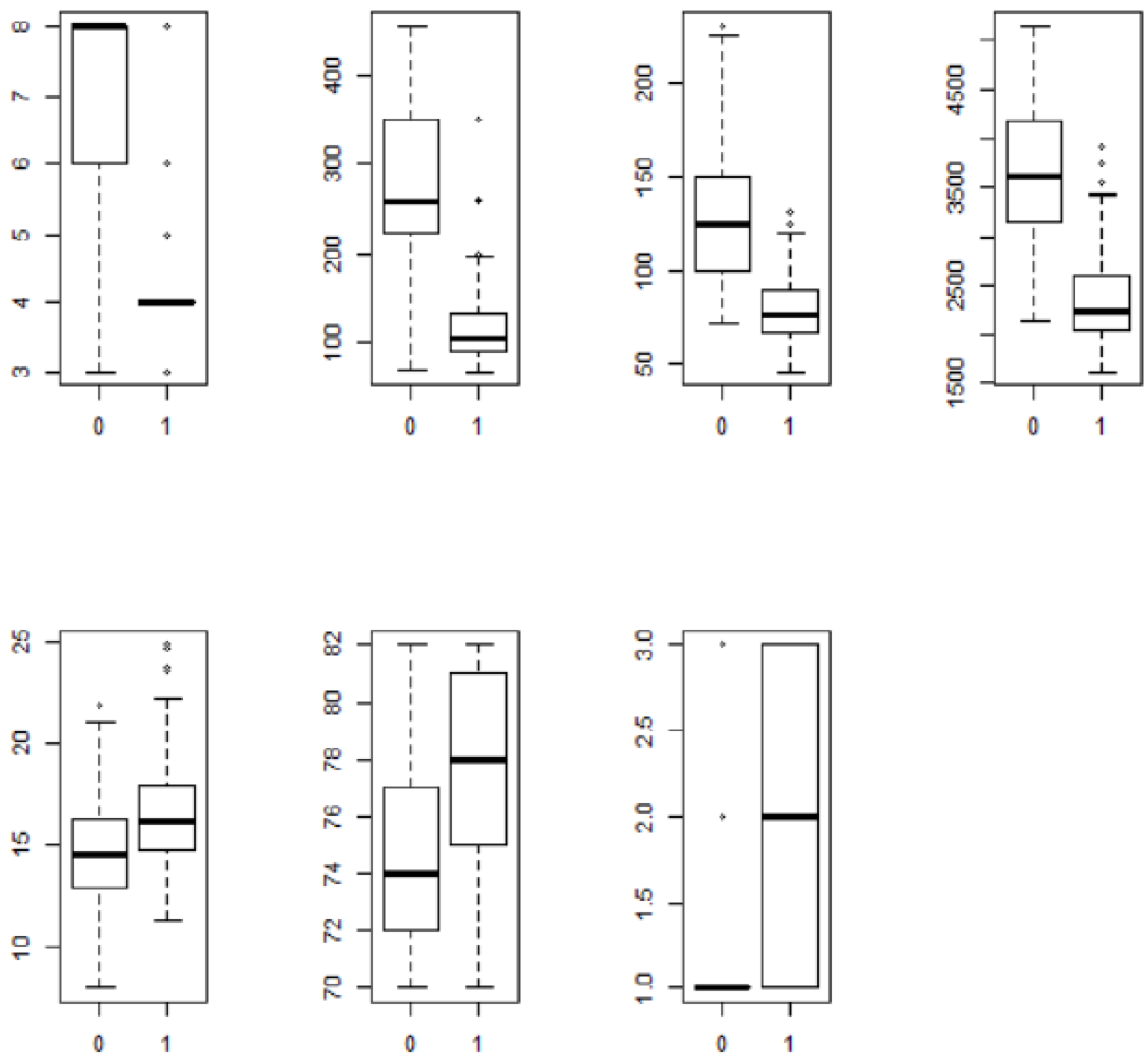
Apurva Shrivastava
925009508
ISEN 613 - 600



Plotting boxplots to better understand the trends:

```
> par(mfrow=c(2,4))
> boxplot(cylinders~mpg01)
> boxplot(displacement~mpg01)
> boxplot(horsepower~mpg01)
> boxplot(weight~mpg01)
> boxplot(acceleration~mpg01)
> boxplot(year~mpg01)
> boxplot(origin~mpg01)
```

Apurva Shrivastava
925009508
ISEN 613 - 600



We observe here that are trends in cylinder, weight, displacement, year and horsepower.
No such trend could be seen for origin and acceleration.

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

I am splitting the data for test, where I am taking training data as years <= 1976 and the data for years > 1976 I will treat them as test data.

Apurva Shrivastava
925009508
ISEN 613 - 600

```
> table(year,mpg01)
    mpg01
year  0  1
  70 22  7
  71 16 11
  72 20  8
  73 34  6
  74 11 15
  75 19 11
  76 20 14
  77 15 13
  78 21 15
  79 13 16
  80  1 26
  81  3 25
  82  1 29
> summary(year)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  70.00   73.00   76.00   75.98   79.00   82.00 
> 
> train<-Auto[year<=76,]
> test<-Auto[year>76,]
> dim(test)
[1] 178  10
> dim(train)
[1] 214  10
> dim(Auto)
[1] 392  10
```

Apurva Shrivastava
925009508
ISEN 613 - 600

(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?

```
> attach(Auto)
The following objects are masked from Auto (pos = 3):

    acceleration, cylinders, displacement, horsepower, mpg, mpg01, name, origin, weight, year

> lda.fit=lda(mpg01~cylinders+weight+displacement+horsepower,data=train)
> summary(lda.fit)
        Length Class  Mode
prior   2      -none- numeric
counts  2      -none- numeric
means   8      -none- numeric
scaling 4      -none- numeric
lev     2      -none- character
svd     1      -none- numeric
N       1      -none- numeric
call    3      -none- call
terms   3      terms  call
xlevels 0      -none- list
> lda.pred = predict(lda.fit, test)
> mean(lda.pred$class == test$mpg01) #Accuracy rate
[1] 0.8932584
> mean(lda.pred$class != test$mpg01) #Error rate
[1] 0.1067416
> |
```

So, we can see that our error rate here is 10.67%

Apurva Shrivastava
925009508
ISEN 613 - 600

(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?

```
> qda.fit=qda(mpg01~cylinders+weight+displacement+horsepower,data=train)
> summary(qda.fit)
        Length Class  Mode
prior    2     -none- numeric
counts   2     -none- numeric
means    8     -none- numeric
scaling 32     -none- numeric
ldet     2     -none- numeric
lev      2     -none- character
N        1     -none- numeric
call     3     -none- call
terms    3     terms  call
xlevels  0     -none- list
> qda.pred = predict(qda.fit, test)
> mean(qda.pred$class == test$mpg01)  #Accuracy rate
[1] 0.8651685
> mean(qda.pred$class != test$mpg01) #Error rate
[1] 0.1348315
> 
```

Error rate is 13.4 %

Apurva Shrivastava
925009508
ISEN 613 - 600

(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?

```
> logistic.fit=glm(mpg01~cylinders+weight+displacement+horsepower,data=train)
> summary(logistic.fit)

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

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-0.92489  -0.17335   0.08104   0.22947   0.71965

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)   1.426e+00  1.395e-01  10.221  < 2e-16 ***
cylinders    -8.179e-02  4.077e-02  -2.006 0.046161 *
weight       -2.202e-04  6.243e-05  -3.526 0.000518 ***
displacement -1.332e-03  7.997e-04  -1.666 0.097205 .
horsepower    3.387e-03  1.141e-03   2.969 0.003340 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for gaussian family taken to be 0.09328779)

    Null deviance: 47.776  on 213  degrees of freedom
Residual deviance: 19.497  on 209  degrees of freedom
AIC: 106.62

Number of Fisher Scoring iterations: 2

> logistic.prob = predict(logistic.fit, test, type="response")
> logistic.pred=rep(0,nrow(test))
> logistic.pred[logistic.prob>0.5]=1
> mean(logistic.pred == test$mpg01) #Accuracy rate
[1] 0.8932584
> mean(logistic.pred != test$mpg01) #Error rate
[1] 0.1067416
>
```

Error rate: 10.67%

Apurva Shrivastava
925009508
ISEN 613 - 600

(g) Perform KNN on the training data, with several values of KK, in order to predict "mpg01" using 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?

```
> train.X = cbind(cylinders, weight, displacement, horsepower)[year<=76, ]
> test.X = cbind(cylinders, weight, displacement, horsepower)[year>76, ]
> train.mpg01 = Auto$mpg01[year<=76]
> set.seed(1)
> # KNN(k=1)
> knn.pred = knn(train.X, test.X,train.mpg01, k = 1)
> summary(knn.pred)
 0  1
81 97
> mean(knn.pred == test$mpg01) #Accuracy rate
[1] 0.8146067
> mean(knn.pred != test$mpg01) #Error rate
[1] 0.1853933
>
> # KNN(k=50)
> knn.pred = knn(train.X, test.X, train.mpg01, k = 50)
> summary(knn.pred)
 0  1
91 87
> mean(knn.pred == test$mpg01) #Accuracy rate
[1] 0.7696629
> mean(knn.pred != test$mpg01) #Error rate
[1] 0.2303371
>
> # KNN(k=100)
> knn.pred = knn(train.X, test.X, train.mpg01, k = 100)
> summary(knn.pred)
 0  1
80 98
> mean(knn.pred == test$mpg01) #Accuracy rate
[1] 0.8314607
> mean(knn.pred != test$mpg01) #Error rate
[1] 0.1685393
>
> Accuracy_Rate=matrix(rep(0,178),178,1)
> for (i in (1:178)){
+ knn.pred_new = knn(train.X, test.X, train.mpg01, k = i)
+ Accuracy_Rate[i,]=mean(knn.pred_new == test$mpg01)
+ i=i+1}
> max(Accuracy_Rate)
[1] 0.8651685
>
```

Apurva Shrivastava
925009508
ISEN 613 - 600

```
> Accuracy_Rate=matrix(rep(0,178),178,1)
> for (i in (1:178)){
+ knn.pred_new = knn(train.X, test.X, train.mpg01, k = i)
+ Accuracy_Rate[i,]=mean(knn.pred_new == test$mpg01)
+ i=i+1}
> max(Accuracy_Rate)
[1] 0.8651685
>
> which.max(Accuracy_Rate)
[1] 122
>
```

Maximum accuracy rate is observed for K = 122, which is 86.61%