Classification of Stock Return Status

Classification using Logistic Regression, LDA and k-NN

Dataset: Weekly
Weekly S&P Stock Market Data
Class: Down – Negative return
Up – Positive return

9 variables and 1089 observations

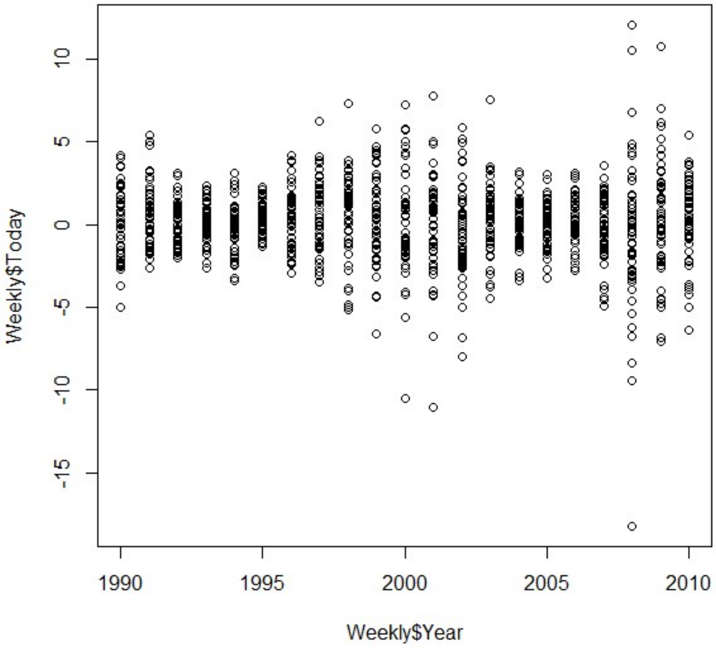
Logistic Regression: It uses linear regression + sigmoid function to classify the data. As sigmoid function returns values from 0 to 1 and values <=0.5 is classified as one class and values>0.5 is classified as other class its binary classification.

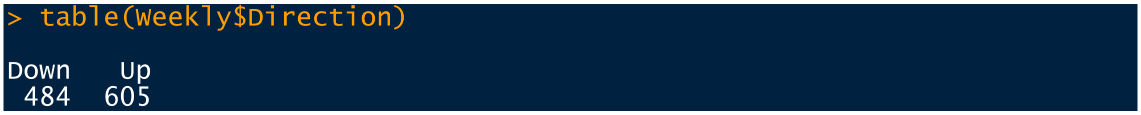
LDA: Linear discriminant analysis – a true decision boundary discovery algorithm which is applied Gaussian data it assumes the class has common covariance and its decision boundary is linear separating the classes.

k-NN: Nearest Neighbor algorithm where k represents numbers of neighbors to be considered to classify the given data point to the closest class out of k neighbors using distance as a metric.

Exploratory Data Analysis:

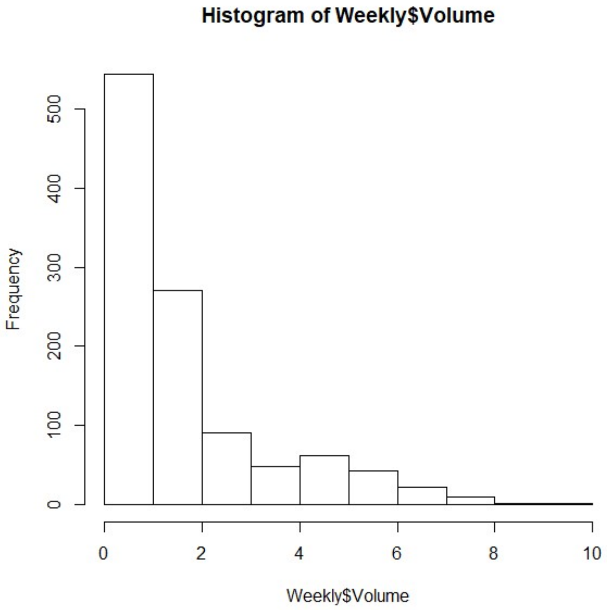
In 2008, the market had extreme positive and negative return.



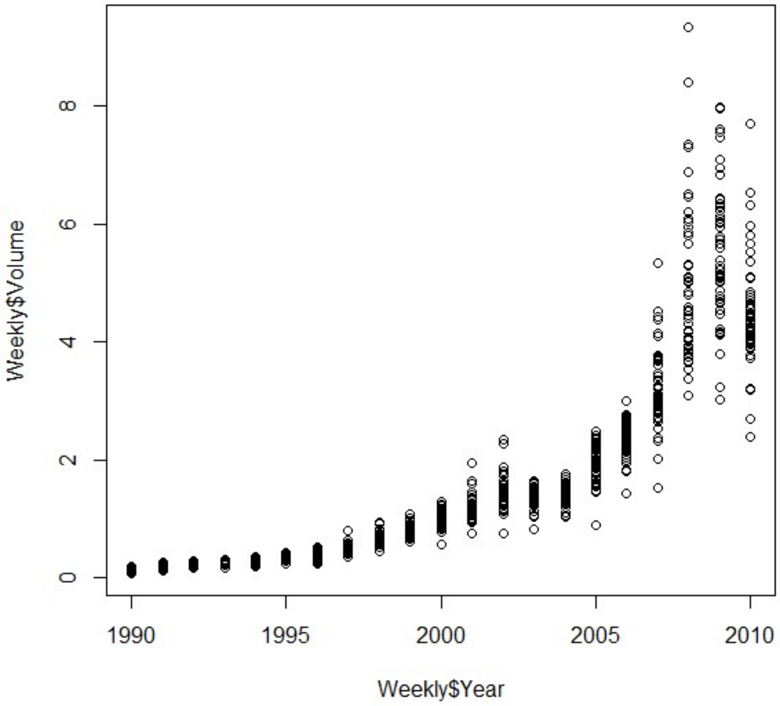


```
> table(Weekly$Direction)

Down   Up
 484  605
```

Market is so volatile that it has so ups and downs.

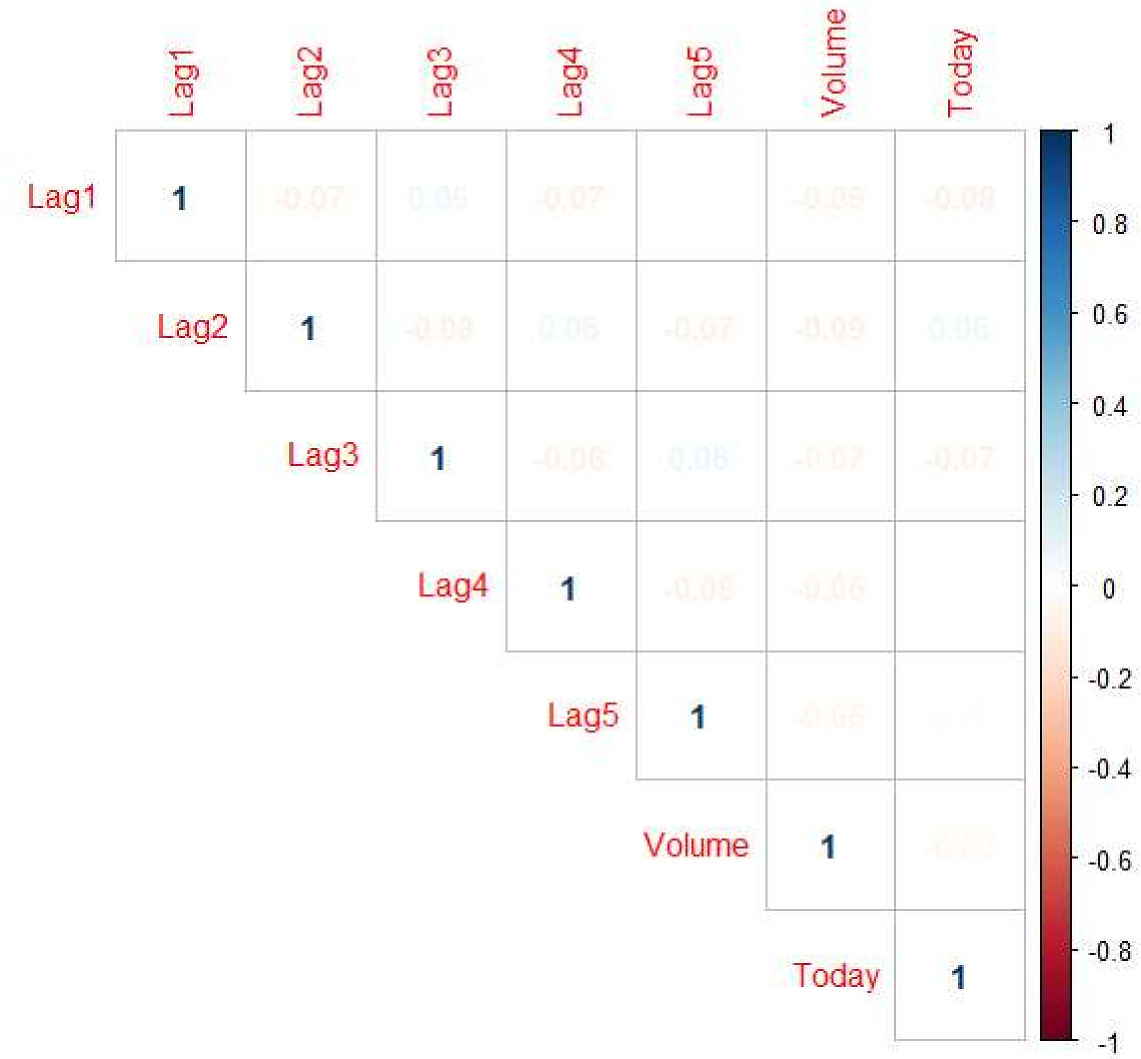


Volume of shares traded in billions



Volume of shares traded per year
Year 2008 in which it has the highest number of shares traded.

Correlation between the variables:

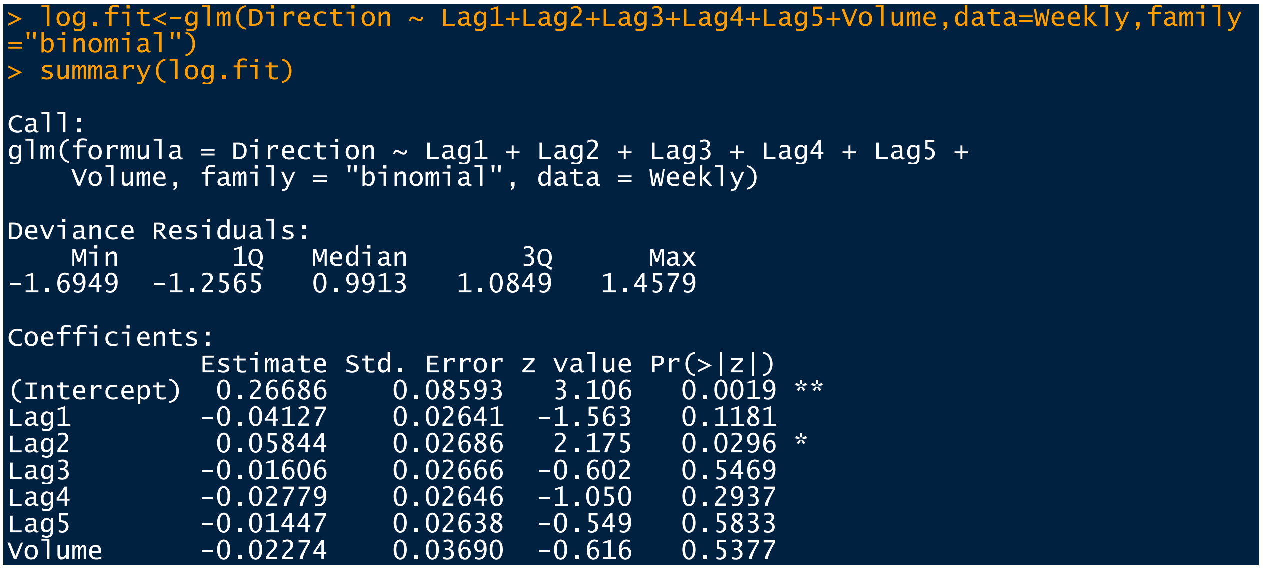


None of the variables are correlated.

<u>Insights from exploratory data analysis:</u>

1. Features are not correlated
2. Volumes of shares traded increased over years
3. Market is highly volatile as the number of positive weekly returns and negative weekly returns are high
4. In 2008 it had highest and lowest weekly returns

Logistic Regression: linear regression combined with sigmoid function for binomial classification

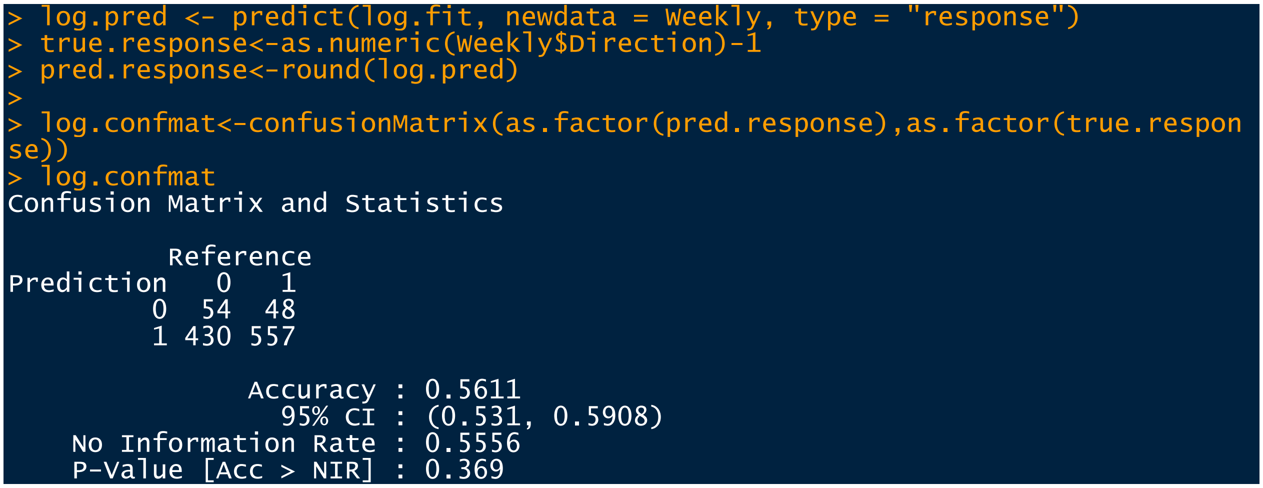


```
> log.fit<-glm(Direction ~ Lag1+Lag2+Lag3+Lag4+Lag5+Volume,data=Weekly,family
="binomial")
> summary(log.fit)

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
```

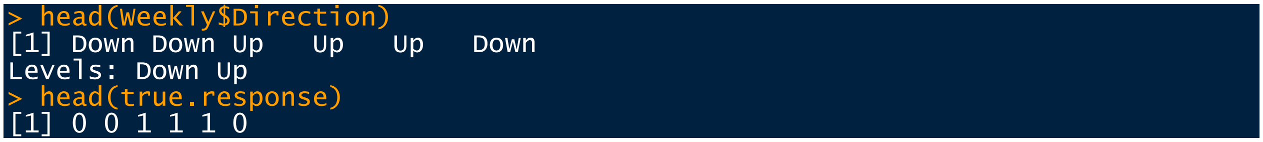
From this we know Lag2 is the most significant feature in prediction of class – direction of positive increase or negative increase in the weekly values.

Prediction of whole data using logistic regression:

Logistic regression fits response variable wrt to predictors passed and resulting value is passed thru sigmoid function which gives value between 0 to 1 and then it is classified as classes 0 and 1 so we have to factor the actual class to compare with the classes predicted by the logistic regression.



```
> log.pred <- predict(log.fit, newdata = Weekly, type = "response")
> true.response<-as.numeric(Weekly$Direction)-1
> pred.response<-round(log.pred)
>
> log.confmat<-confusionMatrix(as.factor(pred.response),as.factor(true.respon
se))
> log.confmat
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0  54  48
         1 430 557

               Accuracy : 0.5611
                 95% CI : (0.531, 0.5908)
    No Information Rate : 0.5556
    P-Value [Acc > NIR] : 0.369
```



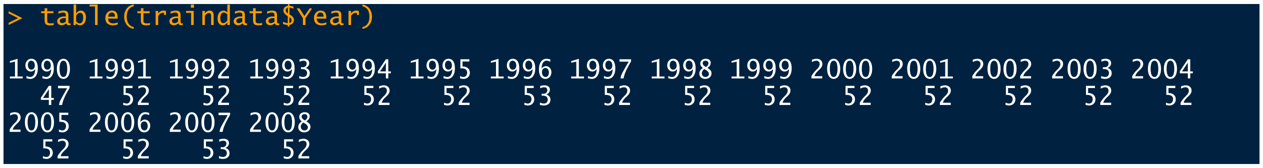
```
> head(Weekly$Direction)
[1] Down Down Up   Up   Up   Down
Levels: Down Up
> head(true.response)
[1] 0 0 1 1 1 0
```

Down – 0
Up- 1

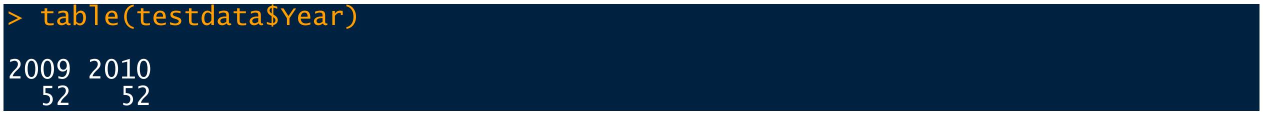
-It has misclassified 48 which are of class UP as DOWN
-It has misclassified 430 which are of class DOWN as UP

Accuracy is 56%.

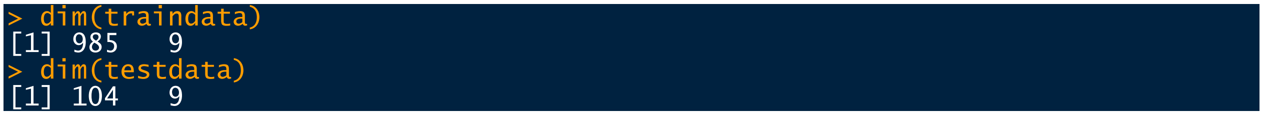
Let’s split data into training – which includes data from 1990 to 2008 and testing – which includes data from 2009 to 2010.



```
> table(traindata$Year)

1990 1991 1992 1993 1994 1995 1996 1997 1998 1999 2000 2001 2002 2003 2004
  47   52   52   52   52   52   53   52   52   52   52   52   52   52   52
2005 2006 2007 2008
  52   52   53   52
```

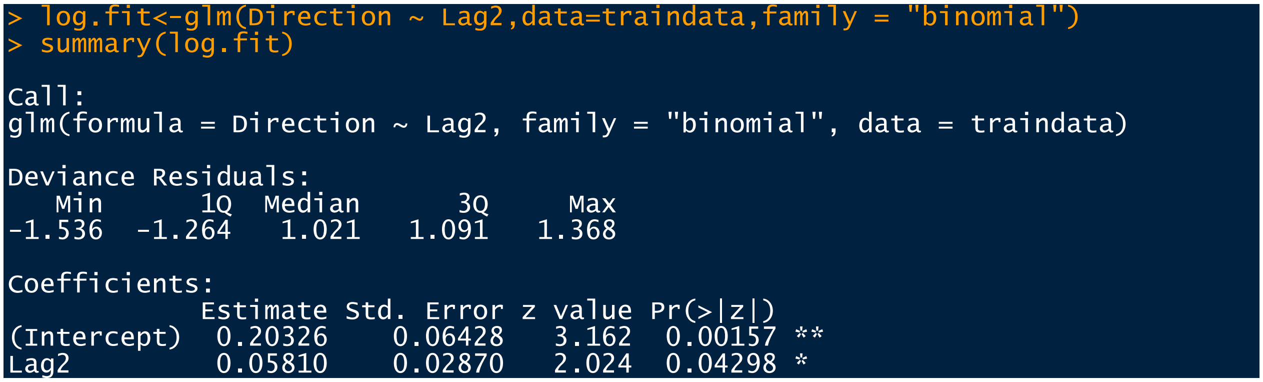


```
> table(testdata$Year)

2009 2010
  52   52
```



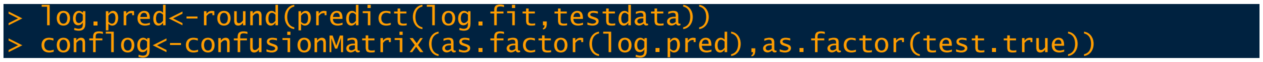
```
> dim(traindata)
[1] 985   9
> dim(testdata)
[1] 104   9
```

Logistic Regression: Response variable Direction only with one predictor Lag2

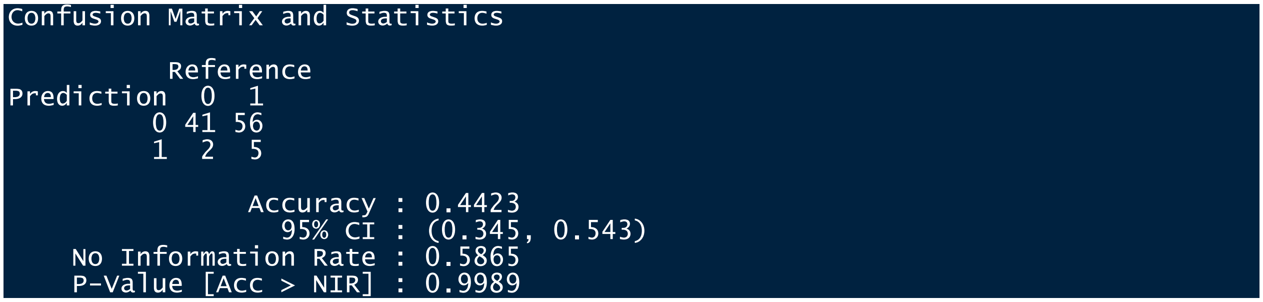


```
> log.fit<-glm(Direction ~ Lag2,data=traindata,family = "binomial")
> summary(log.fit)

Call:
glm(formula = Direction ~ Lag2, family = "binomial", data = traindata)

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 *
```

Prediction accuracy : 44% on held out data



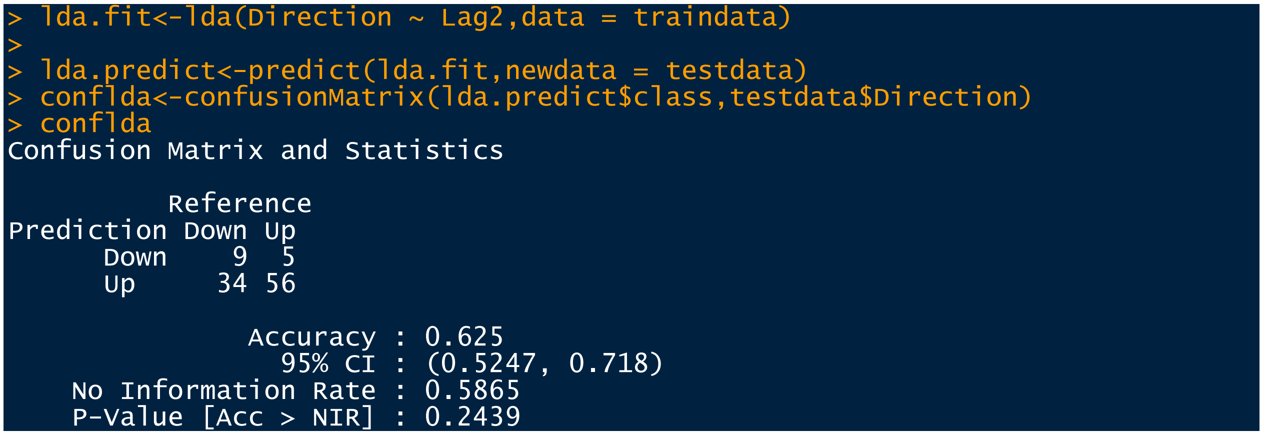
```
> log.pred<-round(predict(log.fit,testdata))
> conflog<-confusionMatrix(as.factor(log.pred),as.factor(test.true))
```



```
Confusion Matrix and Statistics

          Reference
Prediction  0  1
         0 41 56
         1  2  5

               Accuracy : 0.4423
                 95% CI : (0.345, 0.543)
    No Information Rate : 0.5865
    P-Value [Acc > NIR] : 0.9989
```

LDA – Linear discriminant analysis

Finding separation between classes using true decision boundary.

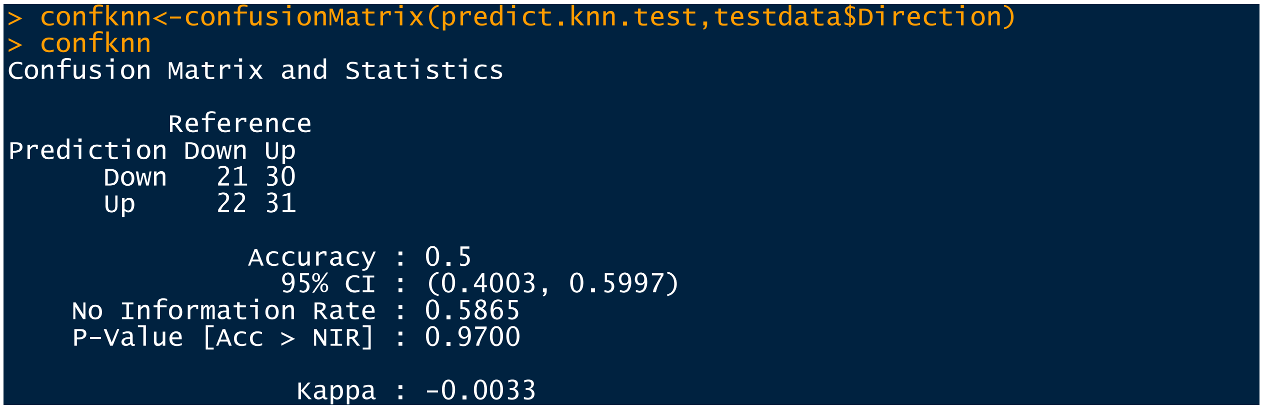


```
> lda.fit<-lda(Direction ~ Lag2,data = traindata)
>
> lda.predict<-predict(lda.fit,newdata = testdata)
> conflda<-confusionMatrix(lda.predict$class,testdata$Direction)
> conflda
Confusion Matrix and Statistics

          Reference
Prediction Down Up
      Down    9  5
      Up     34 56

               Accuracy : 0.625
                 95% CI : (0.5247, 0.718)
    No Information Rate : 0.5865
    P-Value [Acc > NIR] : 0.2439
```

Accuracy of prediction of held out data: **62.5%**

K-Nearest Neighbors

Based on the class of k-nearest neighbors, the given test data point is classified.
k=1



```
> confknn<-confusionMatrix(predict.knn.test,testdata$Direction)
> confknn
Confusion Matrix and Statistics

          Reference
Prediction Down Up
      Down   21 30
      Up     22 31

               Accuracy : 0.5
                 95% CI : (0.4003, 0.5997)
    No Information Rate : 0.5865
    P-Value [Acc > NIR] : 0.9700

                  Kappa : -0.0033
```

Accuracy of prediction of held out data: **50%**

Method	Metric	Accuracy
Logistic Regression	Linear Regression +Sigmoid Function-only for binomial	44.23%
Linear Discriminant Analysis	Linear decision boundary-common variance for all classes	62.50%
k-Nearest Neighbors	Distance from test and k-neighbors to derive class	50%

Conclusion:

If data follows Gaussian distribution its better to use LDA as it separates the classes by drawing the linear decision boundary using the mean and common variance between classes whereas k-NN which uses distance metric and Logistic which uses sigmoid function and it has limitation of binary class which might have higher misclassification rate.

LDA has better accuracy rate compared to other classification methods for the given dataset.