hw_3

Jyoti Ankam

April 4, 2019

Loading the library ISLR

```
library(ISLR)
library(pROC)
library(caret)
library(MASS)
library(class)
```

Viewing the Weekly dataset

```
data(Weekly)
Weekly = Weekly[,-8]
names(Weekly)
```

```
## [1] "Year"      "Lag1"      "Lag2"      "Lag3"      "Lag4"      "Lag5"
## [7] "Volume"    "Direction"
```

```
View(Weekly)
```

a. SUMMARIES OF THE DATASET:

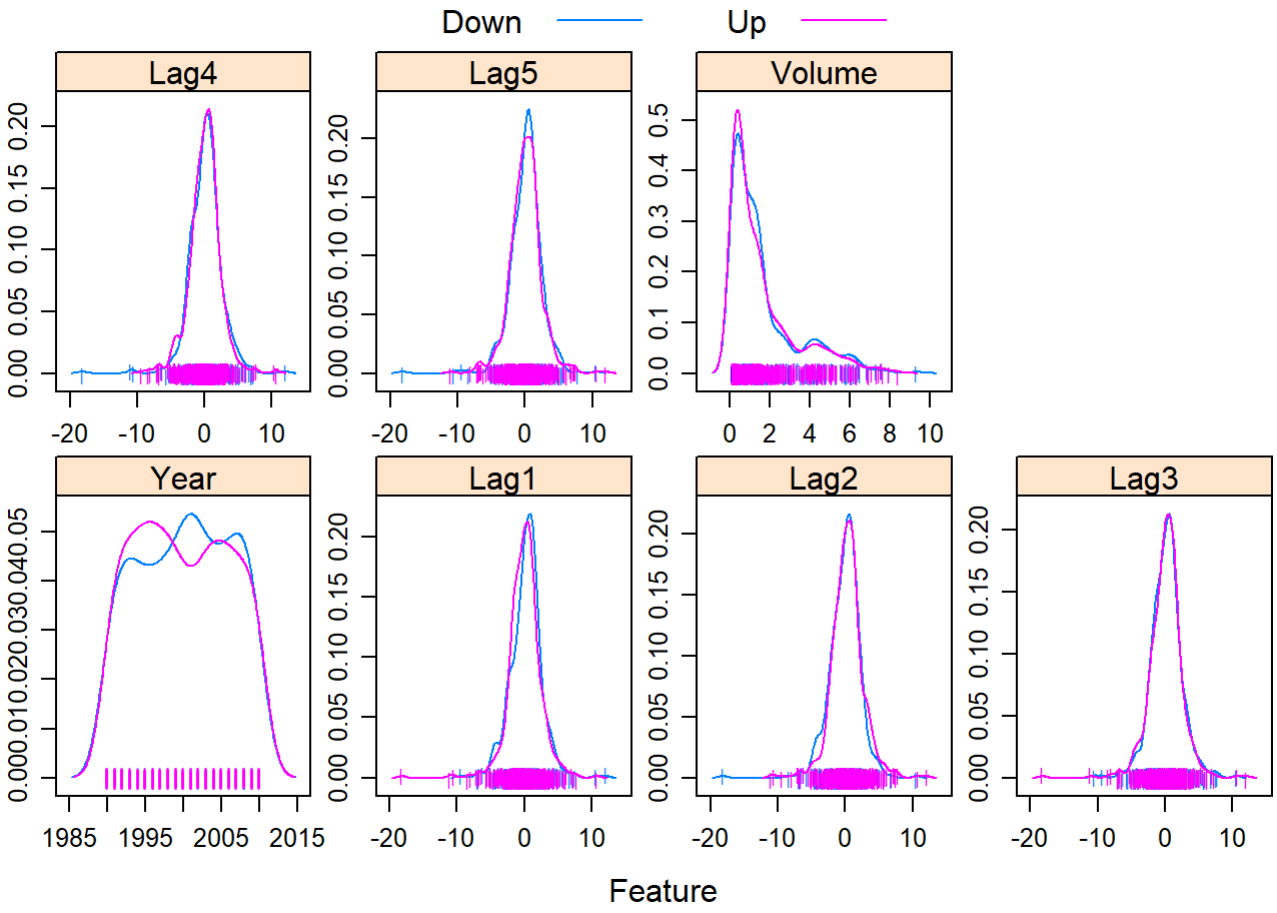
Let's use the summary() fuction to see the numerical summaries of the dataset

```
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        Direction
##  Min.   :-18.1950   Min.   :-18.1950   Min.   :0.08747   Down:484
##  1st Qu.: -1.1580   1st Qu.: -1.1660   1st Qu.:0.33202   Up  :605
##  Median :  0.2380   Median :  0.2340   Median :1.00268
##  Mean   :  0.1458   Mean   :  0.1399   Mean   :1.57462
##  3rd Qu.:  1.4090   3rd Qu.:  1.4050   3rd Qu.:2.05373
##  Max.   : 12.0260   Max.   : 12.0260   Max.   :9.32821
```

Let's see some graphical summaries

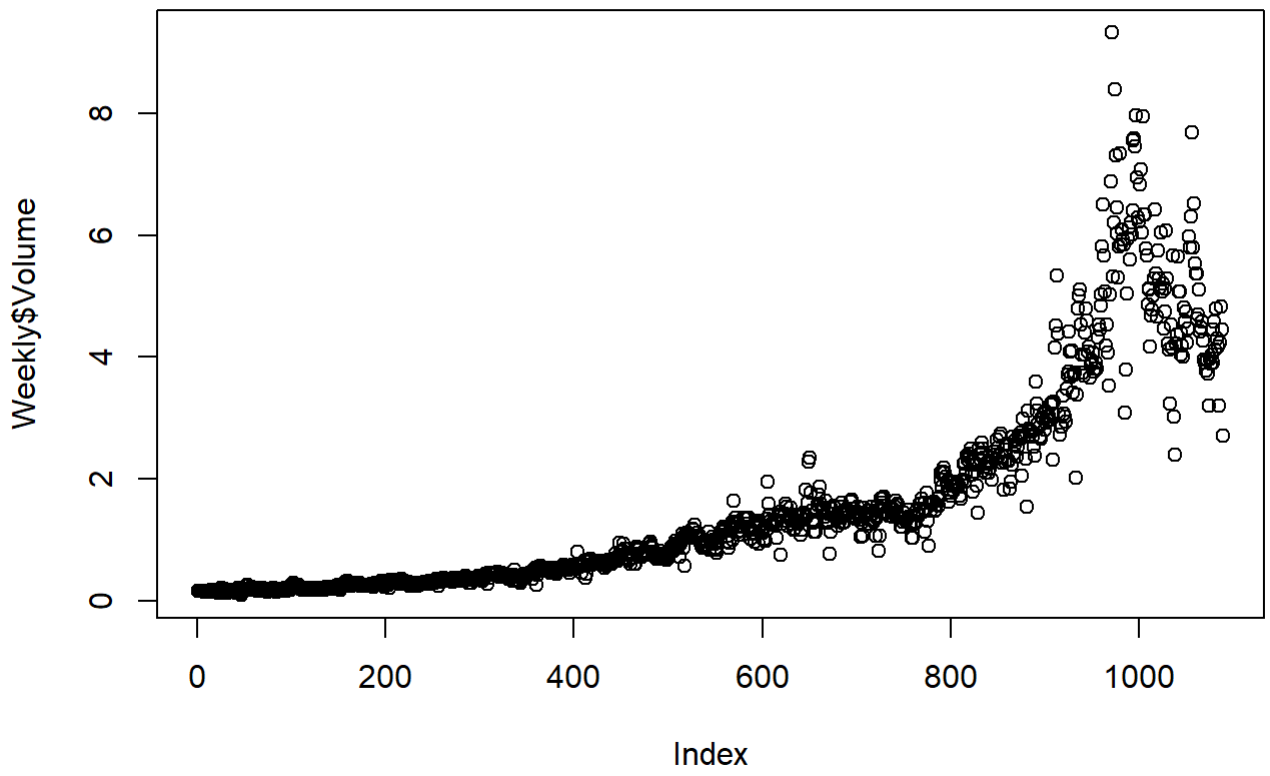
```
featurePlot(x = Weekly[, 1:7],
            y = Weekly$Direction,
            scales = list(x = list(relation = "free"),
            y = list(relation = "free")),
            plot = "density", pch ="|",
            auto.key = list(columns = 2))
```



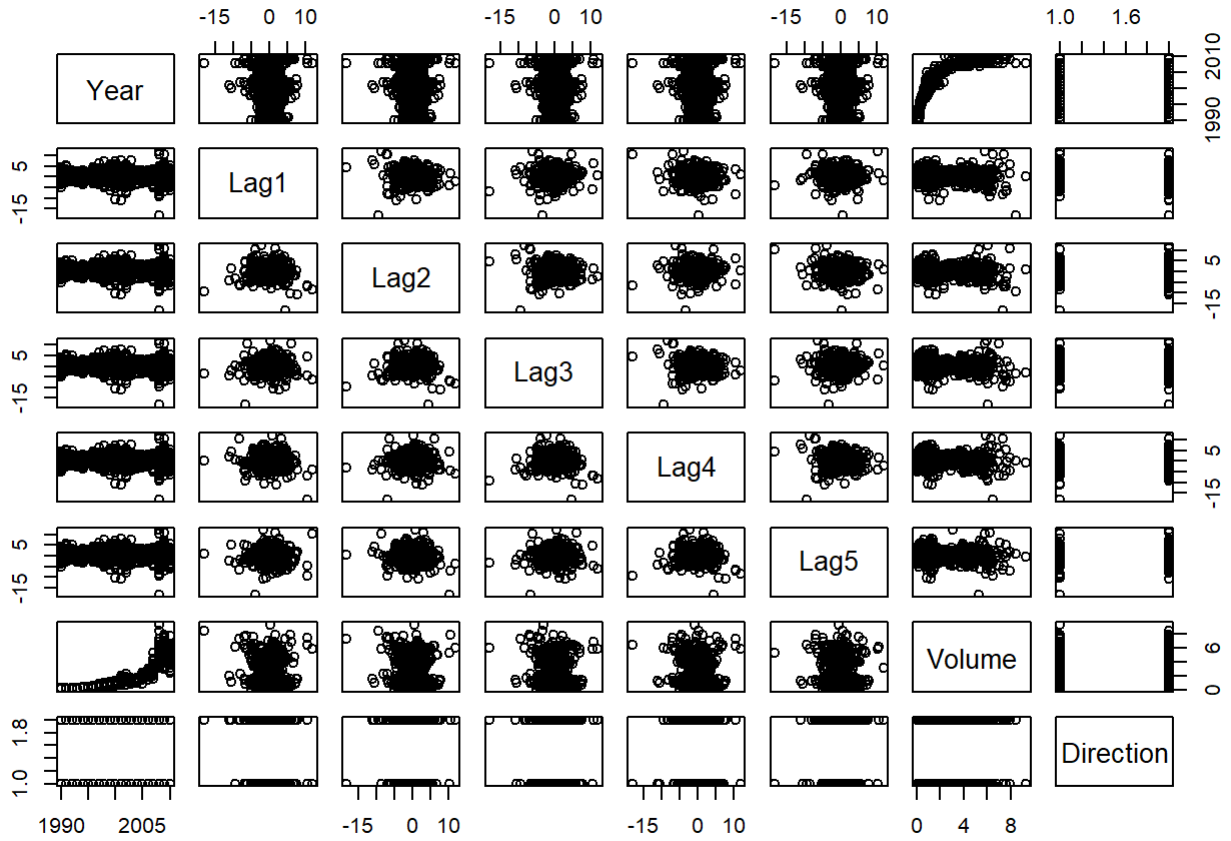

```
cor(Weekly[,-8])
```

```
##              Year         Lag1        Lag2        Lag3        Lag4
## Year    1.00000000 -0.032289274 -0.03339001 -0.03000649 -0.03112792
## Lag1   -0.03228927  1.000000000 -0.07485305  0.05863568 -0.07127388
## Lag2   -0.03339001 -0.074853051  1.00000000 -0.07572091  0.05838153
## Lag3   -0.03000649  0.058635682 -0.07572091  1.00000000 -0.07539587
## Lag4   -0.03112792 -0.071273876  0.05838153 -0.07539587  1.00000000
## Lag5   -0.03051910 -0.008183096 -0.07249948  0.06065717 -0.07567503
## Volume  0.84194162 -0.064951313 -0.08551314 -0.06928771 -0.06107462
##                Lag5      Volume
## Year   -0.030519101  0.84194162
## Lag1   -0.008183096 -0.06495131
## Lag2   -0.072499482 -0.08551314
## Lag3    0.060657175 -0.06928771
## Lag4   -0.075675027 -0.06107462
## Lag5    1.000000000 -0.05851741
## Volume -0.058517414  1.00000000
```

```
plot(Weekly$Volume)
```




```
pairs(Weekly)
```



Summaries

As evident, all the Lags (1 to 5) have similar means and medians which implies that the return percentage has no correlation with time. The correlations between the “lag” variables and today’s returns are close to zero. The only substantial correlation is between “Year” and “Volume”. When we plot “Volume”, we see that it is increasing over time. Apart of these two variables, no other variables display any kind of relationship.

b. Using the full data set to perform a logistic regression with Direction as the response and the five Lag variables plus Volume as predictors:

Logistic Regression

```
log.reg <- glm(Direction ~ Lag1 + Lag2 + Lag3 + Lag4 + Lag5 + Volume, family=binomial, data= Wee
kly)
summary(log.reg)
```

```
##
## 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
```

As seen in the results, only Lag2 is statistically significant with a p-value of 0.0296 which is < 0.05 All the other predictors are statisitcally insignificant. Higher p-values indicates that there is not enough evidence to reject Null hypothesis. With p-value of 0.0296, Lag2 displays some statistical significance.

c. Computing the confusion matrix and overall fraction of correct predictions. Briefly explaining what the confusion matrix is telling you:

Confusion Matrix

We first consider the Bayes classifier (cutoff 0.5) and evaluate its performance on the test data.

```
log.probs <- predict(log.reg,type="response")
log.probs[1:10]
```

```
##         1         2         3         4         5         6         7
## 0.6086249 0.6010314 0.5875699 0.4816416 0.6169013 0.5684190 0.5786097
##         8         9        10
## 0.5151972 0.5715200 0.5554287
```

```
log.pred <- ifelse(log.probs > 0.5, "Up", "Down")

log.table <- table(log.pred,Weekly$Direction)
log.table
```

```
##
## log.pred Down  Up
##     Down   54  48
##     Up    430 557
```

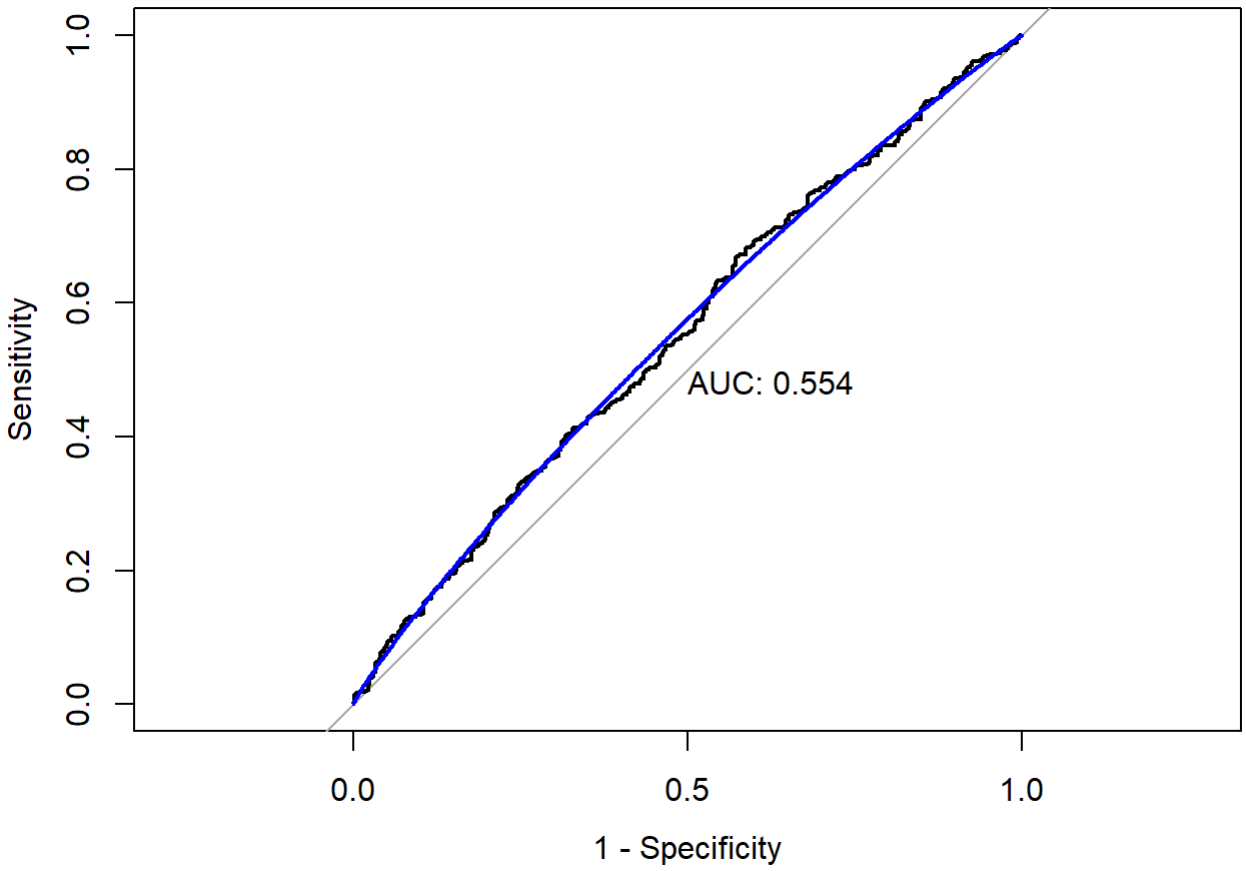
```
accuracy <- (log.table["Down", "Down"] + log.table["Up", "Up"])/nrow(Weekly)
error_rate <- 1 - accuracy

View(accuracy) #0.5610
View(error_rate)#0.4389

sensitivity <-log.table ["Up", "Up"]/(log.table["Down", "Up"] + log.table["Up", "Up"])
View(sensitivity) #0.9206

specificity <-log.table ["Down", "Down"]/(log.table["Down", "Down"] + log.table["Up", "Down"])
View(specificity)#0.1115
```

Based on the accuracy, the % of correct predictions is 56.1% without making Type 1 or 2 errors (i.e. the percentage of correct predictions on the training data is (54+557)/1089 wich is equal to 56.1065%). This means that we were able to perdict the correct trend 56.1% times and also indicates that our prediction is wrong 43.9 % times (i.e. in other words 43.8934803% is the training error rate, which is often overly optimistic) In this case, the model predicts well for UP rather than Down. When the market goes Up, the model predicts it right 92.06% of the times, while, when the market goes Down, it predicts it right only 11.15% of the times. That is to say that for weeks when the market goes up, the model is right 92.0661% of the time (557/(48+557)). For weeks when the market goes down, the model is right only 11.1570% of the time (54/(54+430)).

A confusion matrix is a table that is often used to describe the performance of a classification model (or “classifier”) on a set of test data for which the true values are known. Table () is used to create a confusion matrix. The number at the diagonal gives us the number of correct predictions and the number off the diagonal gives us the number of incorrect predictions.

d. Plotting the ROC curve using the predicted probability from logistic regression and reporting the AUC:

```
roc.glm <- roc(Weekly$Direction, log.probs)
plot(roc.glm, legacy.axes = TRUE, print.auc = TRUE)
plot(smooth(roc.glm), col = 4, add = TRUE)
```



The AUC is 0.554

e. Now fitting the logistic regression model using a training data period from 1990 to 2008, with Lag1 and Lag2 as the predictors.

```
training <- (Weekly$Year < 2009)

Weekly.heldout <- Weekly[!training, ]
Direction.heldout <- Weekly$Direction[!training]

log.reg2 <- glm(Direction ~ Lag1 + Lag2, data = Weekly, family = binomial, subset = training)
summary(log.reg2)
```

```
##
## 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
```

Creating a confusion matrix and performance measures for this heldout model (Note: since the question doen not ask for this, this section is not necessarily a part of the answer to question e)

```
log.probs2 <- predict(log.reg2, newdata=Weekly.heldout, type="response")
log.pred2 <- ifelse(log.probs2 > 0.5, "Up", "Down")
log.table2 <- table(log.pred2, Direction.heldout)

accuracy2 <- (log.table["Down", "Down"] + log.table["Up", "Up"])/nrow(Weekly.heldout)
error_rate2 <- 1 - accuracy2
sensitivity2 <-log.table2 ["Up", "Up"]/(log.table2["Down", "Up"] + log.table2["Up", "Up"])
specificity2 <-log.table2 ["Down", "Down"]/(log.table2["Down", "Down"] + log.table2["Up", "Down"
])


accuracy2
```

```
## [1] 5.875
```

```
error_rate2
```

```
## [1] -4.875
```

```
sensitivity2
```

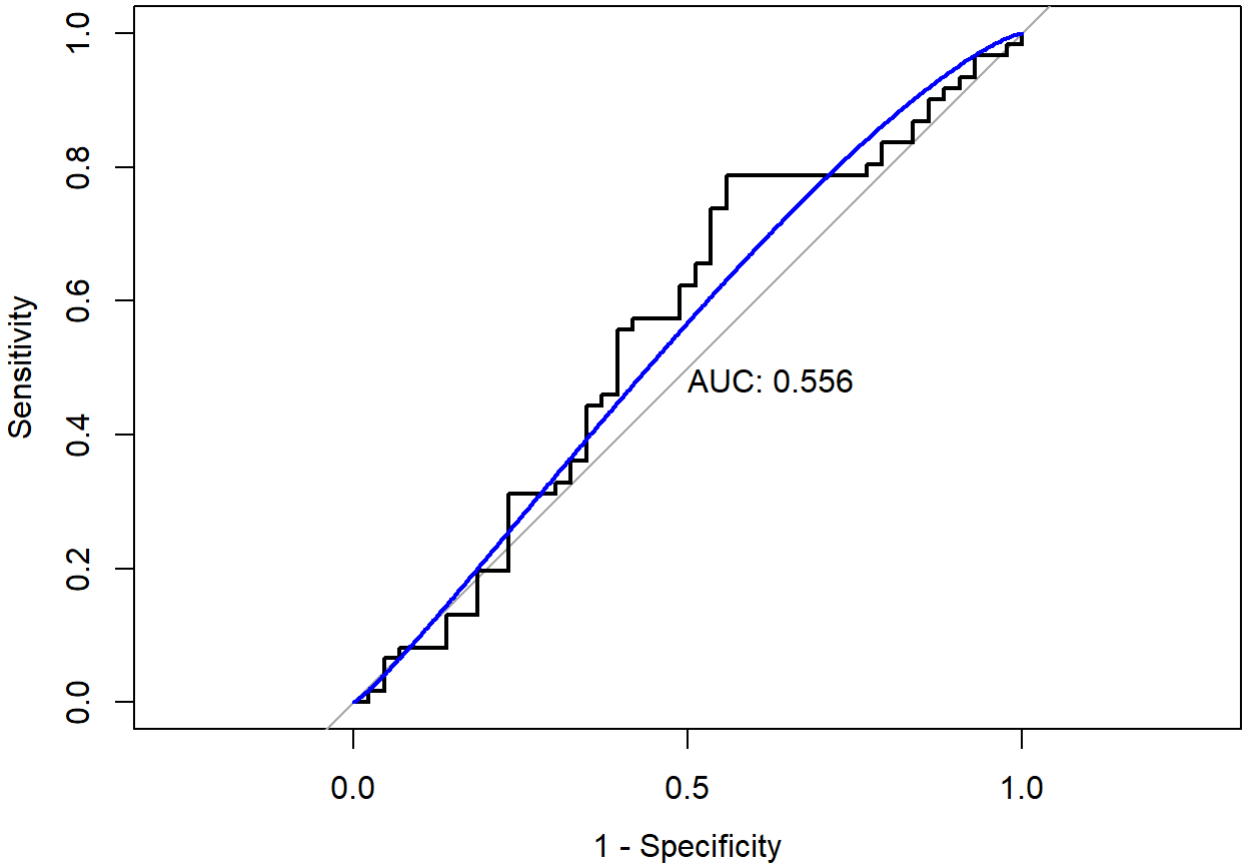
```
## [1] 0.8688525
```

```
specificity2
```

```
## [1] 0.1627907
```

Plotting the ROC curve using the held out data (that is, the data from 2009 and 2010) and reporting the AUC:

```
roc.glm2 <- roc(Direction.heldout, log.probs2)
plot(roc.glm2, legacy.axes = TRUE, print.auc = TRUE)
plot(smooth(roc.glm2), col = 4, add = TRUE)
```



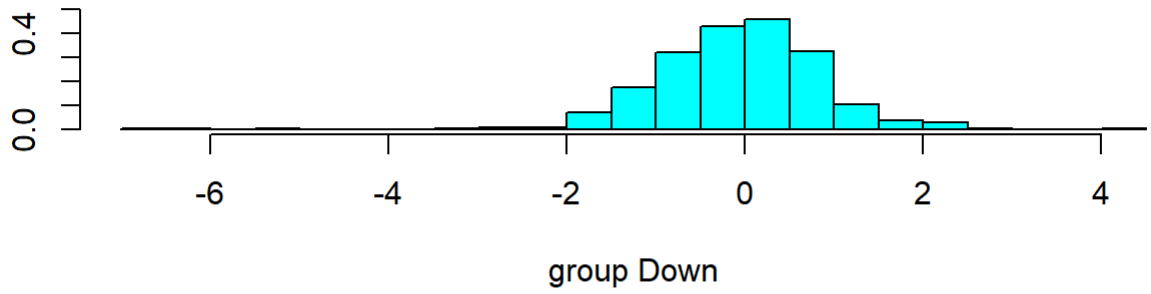
The AUC is 0.556

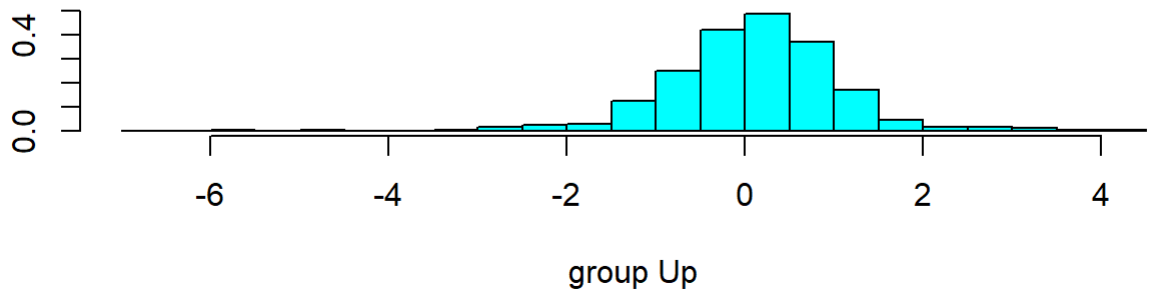
f. Repeating (e) using LDA which implies Linear Discriminant Analysis and QDA which implied Quadratic Discriminant Analysis:

```
fit.lda <- lda(Direction ~ Lag1 + Lag2, data = Weekly, subset = training)
fit.lda
```

```
## 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
```

```
plot(fit.lda)
```



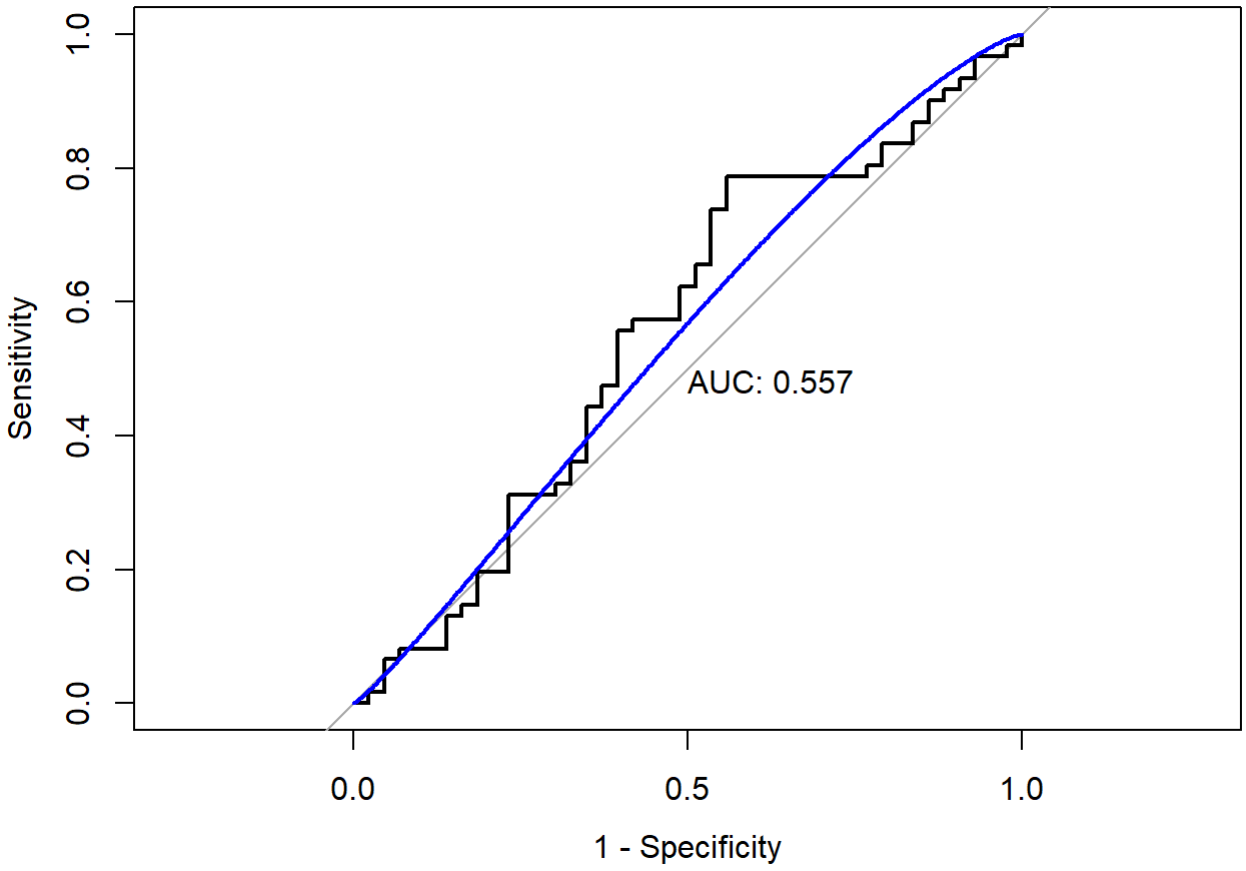



```
pred.lda <- predict(fit.lda, Weekly.heldout)
log.table3 <- table(pred.lda$class, Direction.heldout)
log.table3
```

```
##       Direction.heldout
##        Down Up
##   Down    7  8
##   Up     36 53
```

```
accuracy3 <- (log.table3["Down", "Down"] + log.table3["Up", "Up"])/nrow(Weekly.heldout)
error_rate3 <- 1 - accuracy3
sensitivity3 <-log.table3 ["Up", "Up"]/(log.table3["Down", "Up"] + log.table3["Up", "Up"])
specificity3 <-log.table3 ["Down", "Down"]/(log.table3["Down", "Down"] + log.table3["Up", "Down"
])
```

```
lda.probs = pred.lda$posterior[,2]
roc.glm3 <- roc(as.numeric(Direction.heldout), as.numeric(lda.probs))
plot(roc.glm3, legacy.axes = TRUE, print.auc = TRUE)
plot(smooth(roc.glm3), col = 4, add = TRUE)
```



The

AUC for the logistic regression model with Lag1 and Lag2 is 0.557

```
fit.qda <- qda(Direction ~ Lag1 + Lag2, data = Weekly, subset = training)
fit.qda
```

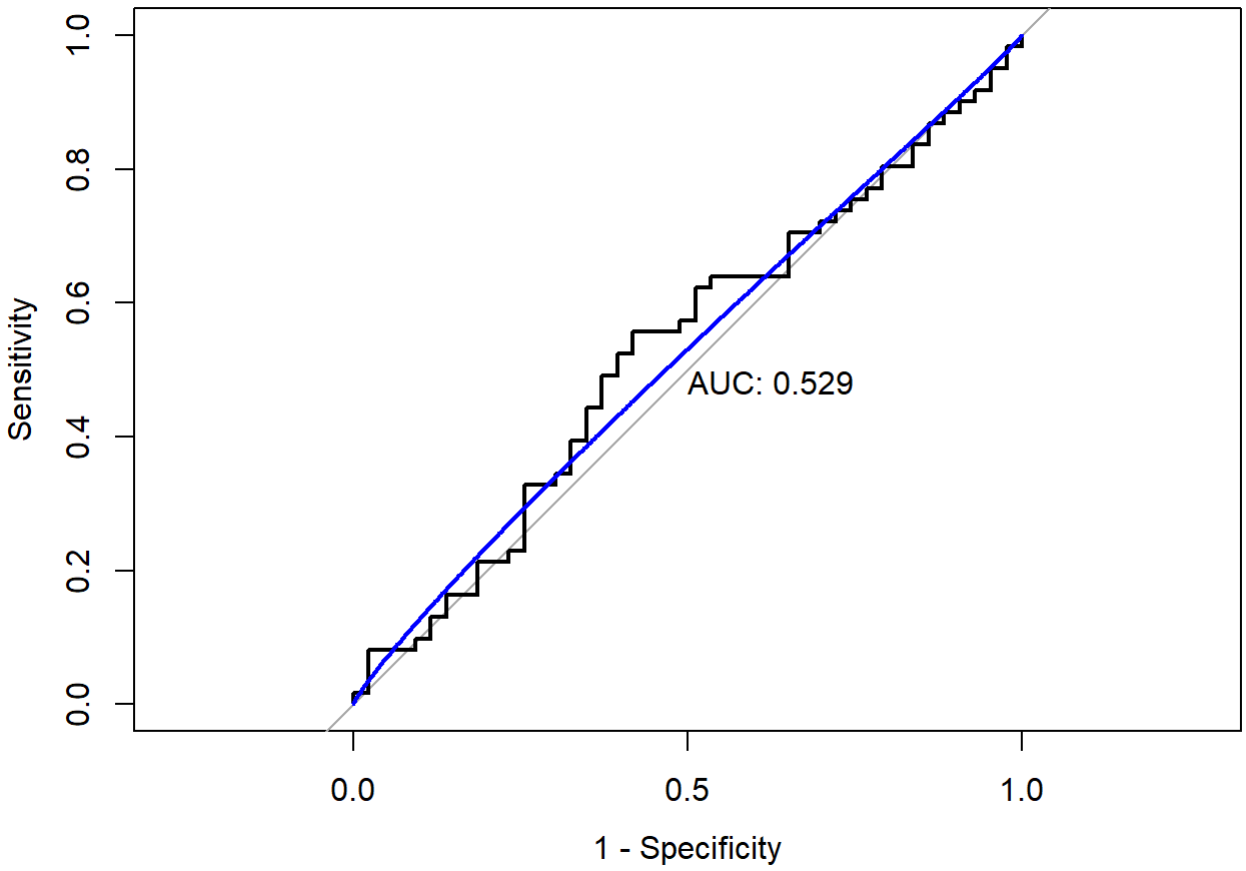
```
## Call:
## qda(Direction ~ Lag1 + Lag2, data = Weekly, subset = training)
##
## Prior probabilities of groups:
##      Down        Up
## 0.4477157 0.5522843
##
## Group means:
##              Lag1        Lag2
## Down  0.289444444 -0.03568254
## Up   -0.009213235  0.26036581
```

```
pred.qda <- predict(fit.qda, Weekly.heldout)
log.table4 <- table(pred.qda$class, Direction.heldout)
log.table4
```

```
##       Direction.heldout
##        Down Up
##   Down    7 10
##   Up     36 51
```

```
accuracy4 <- (log.table4["Down", "Down"] + log.table4["Up", "Up"])/nrow(Weekly.heldout)
error_rate4 <- 1 - accuracy4
sensitivity4 <-log.table4 ["Up", "Up"]/(log.table4["Down", "Up"] + log.table4["Up", "Up"])
specificity4 <-log.table4 ["Down", "Down"]/(log.table4["Down", "Down"] + log.table4["Up", "Down"
])
```

```
qda.probs = pred.qda$posterior[,2]
roc.glm4 <- roc(as.numeric(Direction.heldout), as.numeric(qda.probs))
plot(roc.glm4, legacy.axes = TRUE, print.auc = TRUE)
plot(smooth(roc.glm4), col = 4, add = TRUE)
```

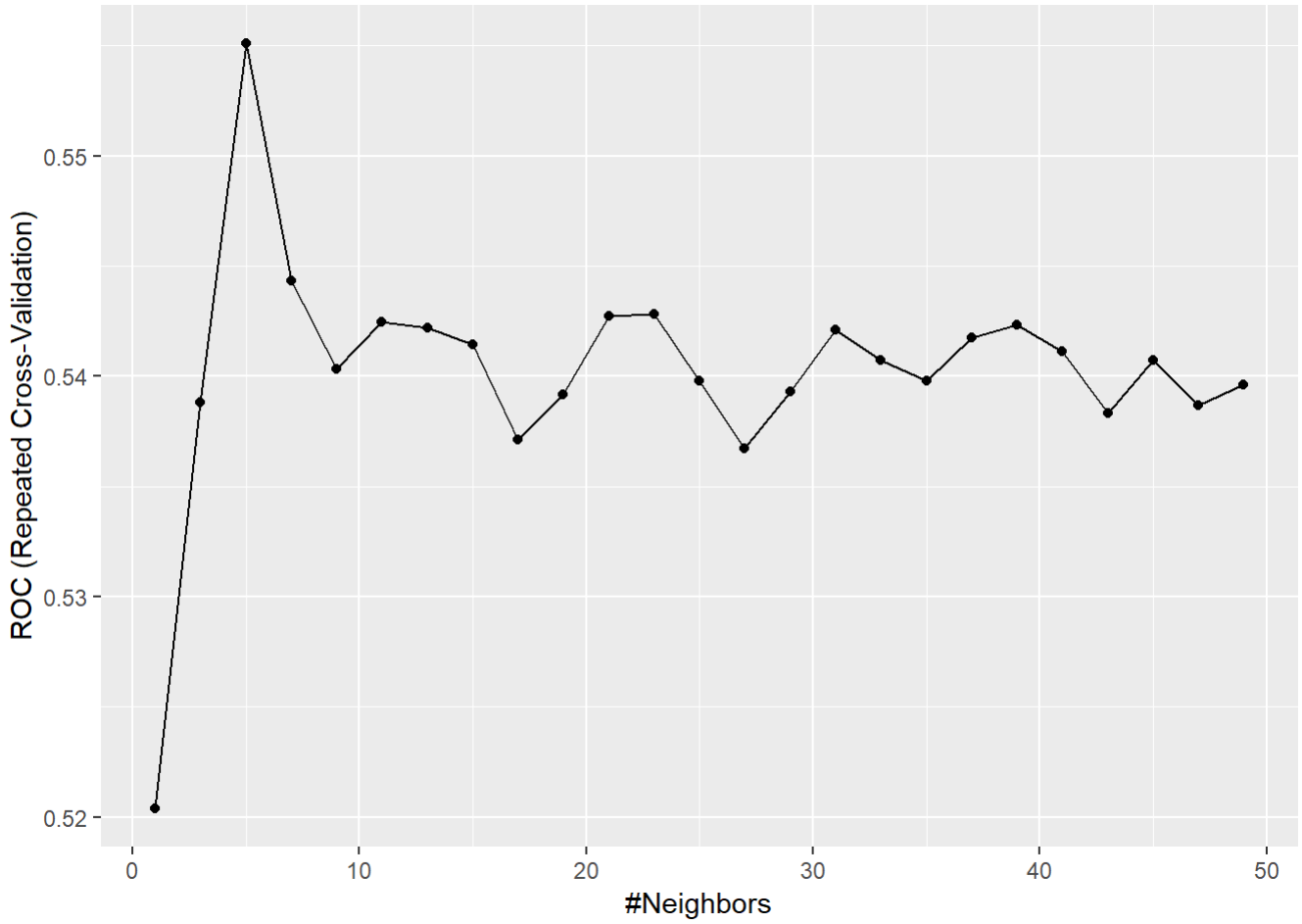


The

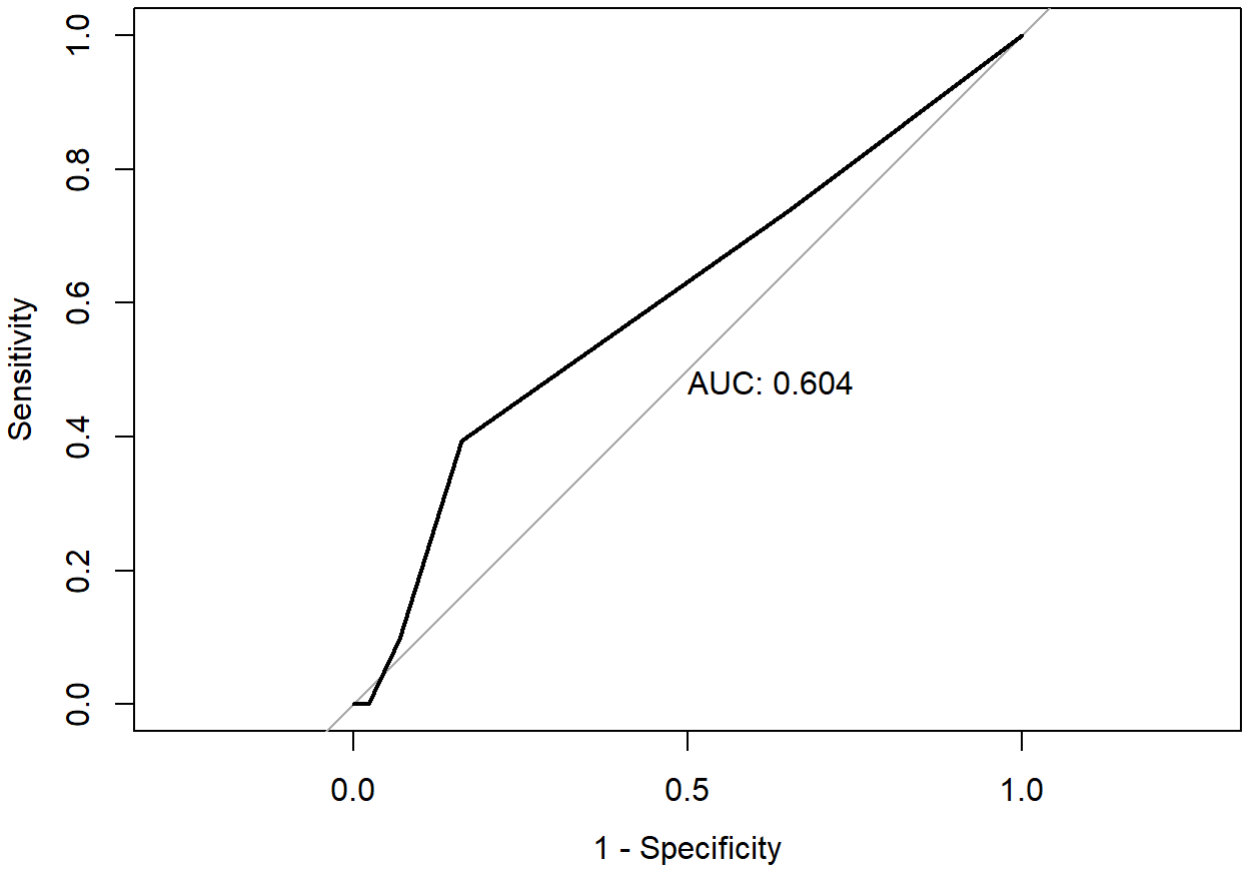
AUC is 0.529

g. Repeat (e) using KNN. Briefly discuss your results.

```
ctrl <- trainControl(method = "repeatedcv",
                     repeats = 5,
                     summaryFunction = twoClassSummary,
                     classProbs = TRUE)
set.seed(1)
model.knn <- train(x= Weekly[training, 1:2],
                   y = Weekly$Direction[training],
                   method = "knn",
                   preProcess = c("center", "scale"),
                   tuneGrid = data.frame(k = seq(1, 50, by = 2)),
                   trControl = ctrl,
                   metric = 'ROC')


ggplot(model.knn)
```




```
pred.knn <- predict(model.knn, newdata = Weekly.heldout, type = "prob")[,2]

roc.knn <- roc(Direction.heldout, pred.knn)
plot(roc.knn, legacy.axes = TRUE, print.auc = TRUE)
```



The AUC for the K Nearest Neighbor is 0.604

Looking at the models, it appears that logistic regression model works a little better than just random guessing. But since we have trained and tested the model, on the same set of observations, our results may be incorrect and misleading. Our training error rate which is 43.893% seems to be too optimistic and underestimating the test error rate. Therefore, for better prediction and accuracy of the logistic regression model, we partition the data and fit the model and then estimate how well it predicts the heldout data, thus getting a more realistic error rate.

In order to compare the performance of a classifier based on different cut off points, we get the AUC of the heldout data. More or less, the AUC of the helout data is similar to the full dataset. We get the AUC of the logisitc model with the complete data set as 0.557 and the AUC of the heldout data as 0.554. The AUC for the K Nearest Neighbor is 0.604. The AUC for QDA is 0.529 and for LDA is 0.557. Although, AUC for KNN is slightly higher, overall they all perform similarly as it is hard to predict changes in stock price based on the previous days (lag1 and lag2). Additionally, using the lag1 and lag2 predictors, the logisitic regression p-values are not significant for the data subset suggesting that lag1 and lag2 may not be significantly associated with the Direction. Predictors which aren't associated with the outcome cause an increase in variance without any corresponding decrease in bias, thereby resulting in inaccurate/inadequate predictions.