P8106 HOMEWORK 3

xc2474 Xinlei Chen

4/9/2019

Problem

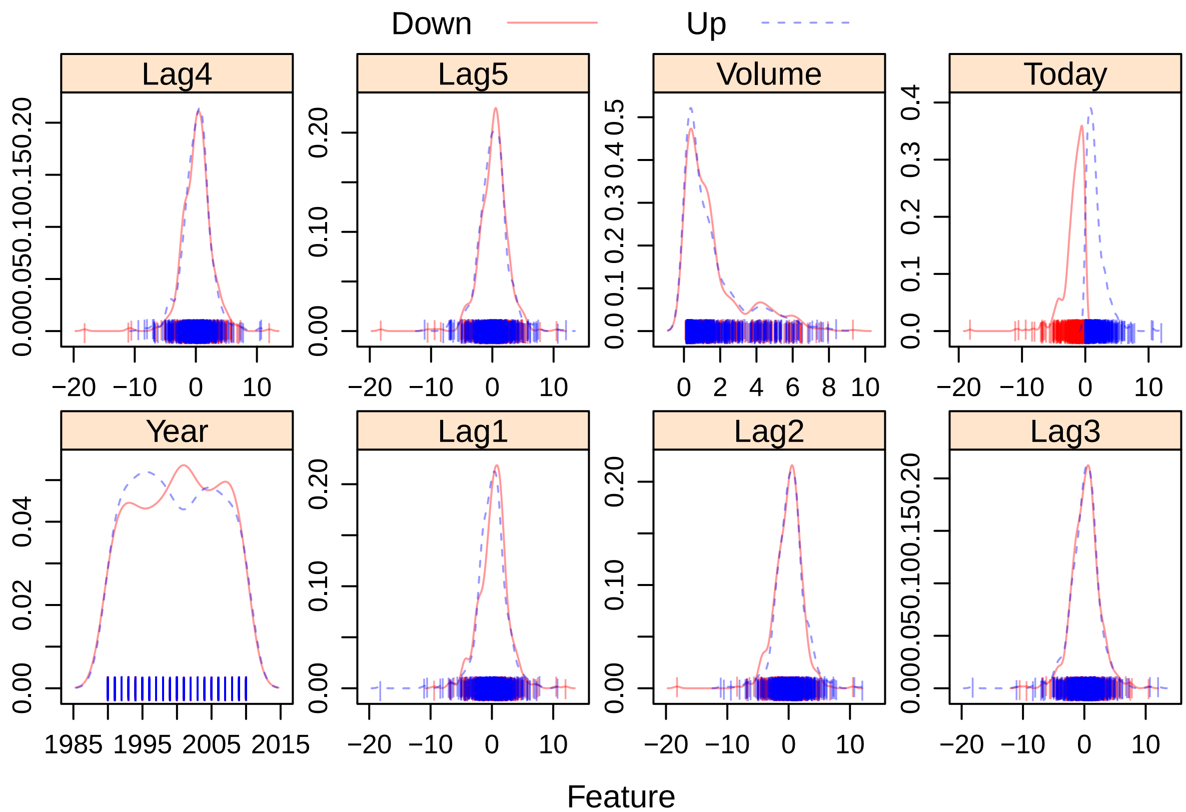
This questions will be answered using the Weekly data set, which is part of the ISLR package. This data is similar in nature to the Smarket data on the textbook except that it contains 1,089 weekly returns for 21 years, from the beginning of 1990 to the end of 2010. A description of the data can be found by typing ?Weekly in the Console. (Note that the column Today is not a predictor.)

```r
# load packages
library(tidyverse)
library(ISLR)
library(caret)
library(AppliedPredictiveModeling)
library(pROC)
library(MASS)
library(class)
#import data
data("Weekly")
dat = Weekly
head(dat)
```

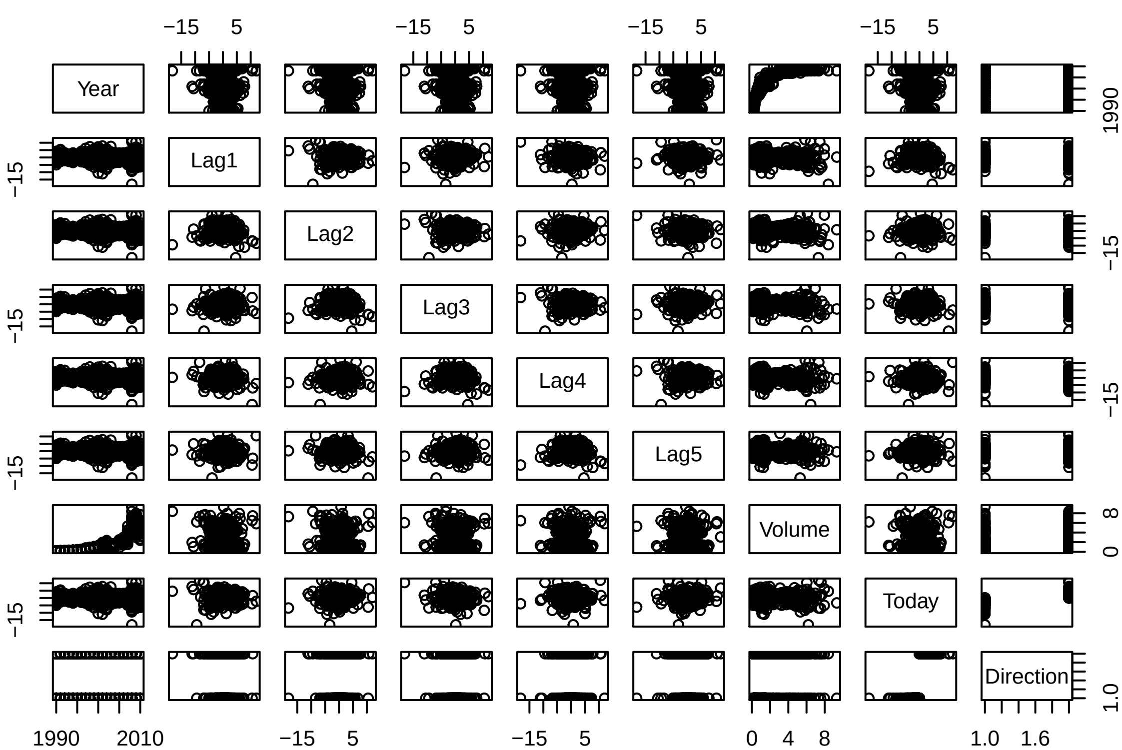
```
##   Year   Lag1   Lag2   Lag3   Lag4   Lag5    Volume  Today Direction
## 1 1990  0.816  1.572 -3.936 -0.229 -3.484 0.1549760 -0.270      Down
## 2 1990 -0.270  0.816  1.572 -3.936 -0.229 0.1485740 -2.576      Down
## 3 1990 -2.576 -0.270  0.816  1.572 -3.936 0.1598375  3.514        Up
## 4 1990  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
```

(a) Produce some graphical summaries of the Weekly data.

```r
transparentTheme(trans = .4)
featurePlot(x = dat[, 1:8],
            y = dat$Direction,
            scales = list(x=list(relation="free"),
                          y=list(relation="free")),
            plot = "density", pch = "|",
            auto.key = list(columns = 2))
```


Down
Up
Lag4
Lag5
Volume
Today
Year
Lag1
Lag2
Lag3
Feature

```r
pairs(dat)
```


Year
Lag1
Lag2
Lag3
Lag4
Lag5
Volume
Today
Direction

(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. Do any of the predictors appear to be statistically significant? If so, which ones?

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

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

Lag2 appears to be statistically significant with p-value 0.0296, which is less than 0.05.

(c) Compute the confusion matrix and overall fraction of correct predictions. Briefly explain what the confusion matrix is telling you.

```
test.pred.prob <- predict(glm.fit, type = "response")
test.pred <- rep("Down", length(test.pred.prob))
test.pred[test.pred.prob>0.5] <- "Up"
confusionMatrix(data = as.factor(test.pred), reference = dat$Direction)
```

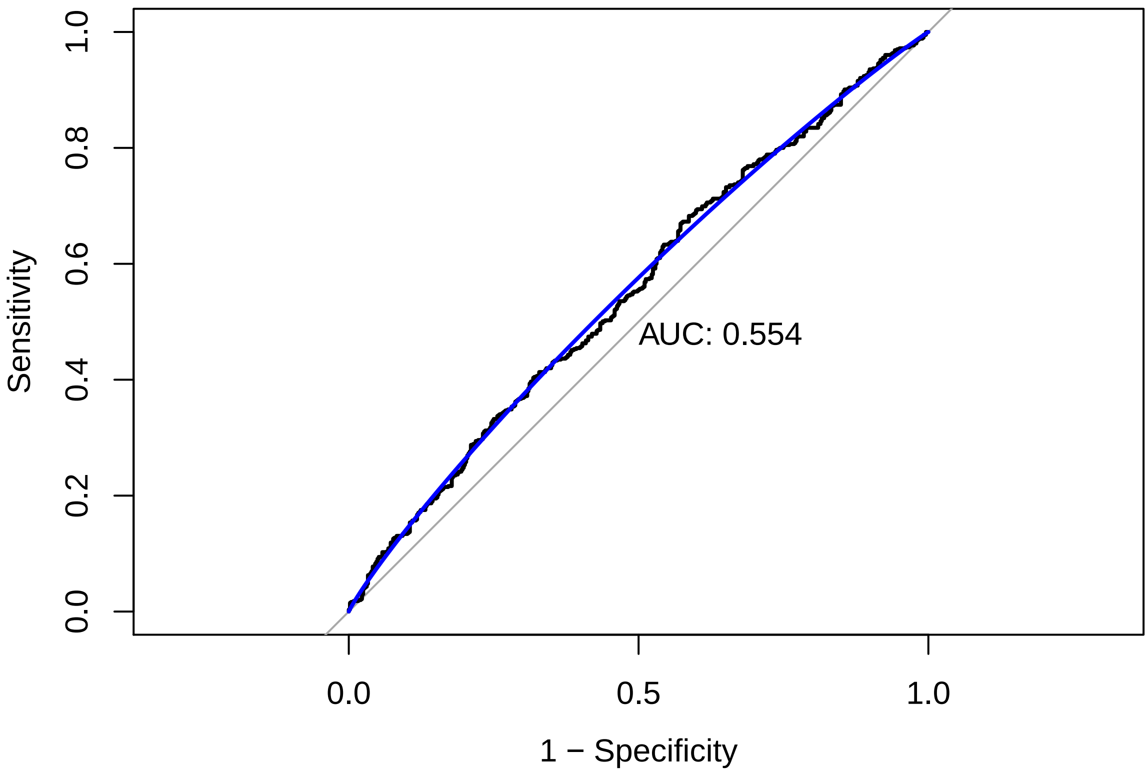
```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction Down  Up
##       Down   54  48
##       Up    430 557
##
```

```
##                Accuracy : 0.5611
##                  95% CI : (0.531, 0.5908)
##     No Information Rate : 0.5556
##     P-Value [Acc > NIR] : 0.369
##
##                   Kappa : 0.035
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.11157
##             Specificity : 0.92066
##          Pos Pred Value : 0.52941
##          Neg Pred Value : 0.56434
##              Prevalence : 0.44444
##          Detection Rate : 0.04959
##    Detection Prevalence : 0.09366
##       Balanced Accuracy : 0.51612
##
##        'Positive' Class : Down
##
```

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. This confusion matrix tells us that (1) the percentage of correct predictions on the training data is 56.11%, or say, the training error rate is 43.89%. (2) For weeks when the market goes up, the model is right 92.07% of the time; for weeks when the market goes down, the model is right 11.16% of the time.

(d) Plot the ROC curve using the predicted probability from logistic regression and report the AUC.

```
roc.glm <- roc(dat$Direction, test.pred.prob)
plot(roc.glm, legacy.axes = TRUE, print.auc = TRUE)
plot(smooth(roc.glm), col = 4, add = TRUE)
```

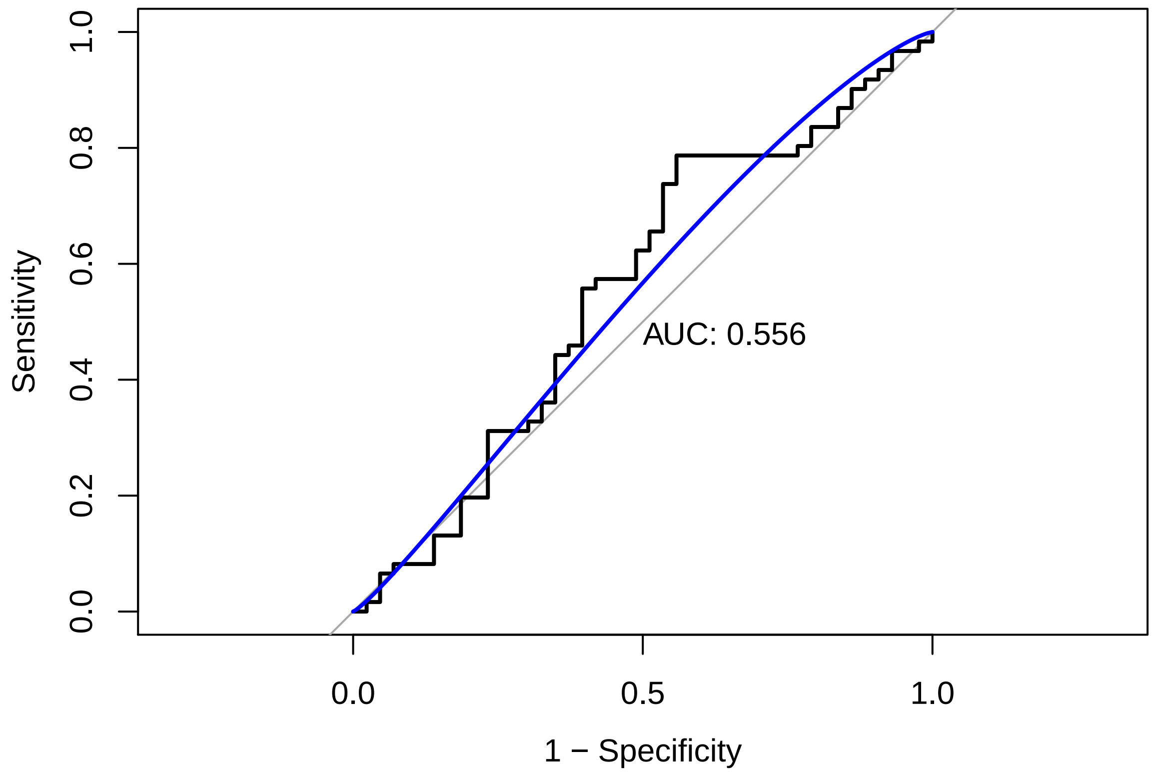


The AUC is 0.554.

(e) Now fit the logistic regression model using a training data period from 1990 to 2008, with Lag1 and Lag2 as the predictors. Plot the ROC curve using the held out data (that is, the data from 2009 and 2010) and report the AUC.

```
trainset = (dat$Year<=2008)
testset = dat[!trainset,]
```

```
glm.fit.d <- glm(Direction ~ Lag1 + Lag2, data=dat, subset=trainset, family="binomial")
glm.probs.d <- predict(glm.fit.d, type="response", newdata=testset)
roc.glm <- roc(testset$Direction, glm.probs.d)
plot(roc.glm, legacy.axes = TRUE, print.auc = TRUE)
plot(smooth(roc.glm), col = 4, add = TRUE)
```



The AUC is 0.556.

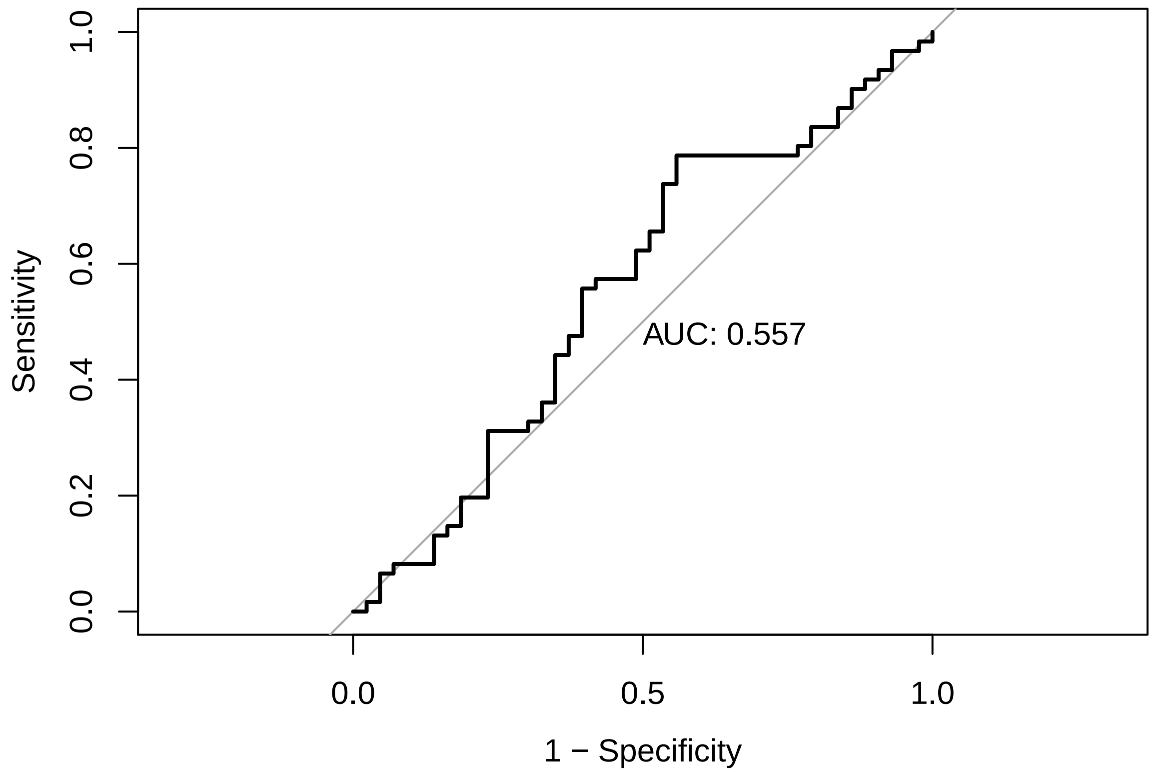
(f) Repeat (e) using LDA and QDA.

```
# LDA
lda.fit <- lda(Direction ~ Lag1 + Lag2, data=dat, subset=trainset)
lda.pred <- predict(lda.fit, newdata = testset)
head(lda.pred$posterior)
```

```
##          Down        Up
## 986 0.5602039 0.4397961
## 987 0.3079163 0.6920837
## 988 0.4458032 0.5541968
## 989 0.4785107 0.5214893
## 990 0.4657943 0.5342057
## 991 0.5262907 0.4737093
```

```
roc.lda <- roc(testset$Direction, lda.pred$posterior[,2],
               levels = c("Down", "Up"))

plot(roc.lda, legacy.axes = TRUE, print.auc = TRUE)
```



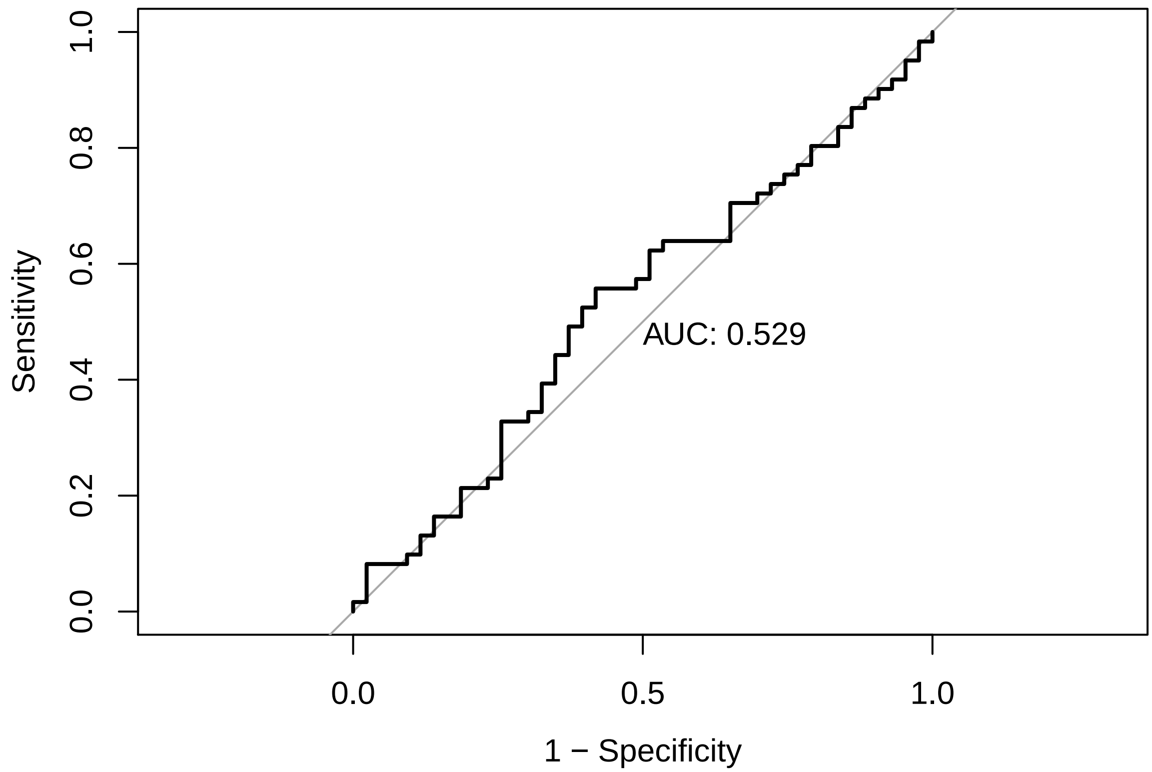
For LDA, the AUC is 0.557.

```
# QDA
qda.fit <- qda(Direction ~ Lag1 + Lag2, data=dat, subset=trainset)
qda.pred <- predict(qda.fit, newdata = testset)
head(qda.pred$posterior)
```

```
##          Down        Up
## 986 0.5436205 0.4563795
## 987 0.3528814 0.6471186
## 988 0.2227273 0.7772727
## 989 0.3483016 0.6516984
## 990 0.4598550 0.5401450
## 991 0.5119613 0.4880387
```

```
roc.qda <- roc(testset$Direction, qda.pred$posterior[,2],
               levels = c("Down", "Up"))

plot(roc.qda, legacy.axes = TRUE, print.auc = TRUE)
```



For QDA, the AUC is 0.529.

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

```
# choose the best K
train = dat %>%
  filter(Year<=2008)

test = dat %>%
  filter(Year>2008)

ctrl <- trainControl(method="repeatedcv",repeats = 3,classProbs=TRUE,summaryFunction = twoClassSummary)
knnFit <- train(Direction ~ Lag1 + Lag2, data = train,
                method = "knn",
                trControl = ctrl,
                preProcess = c("center","scale"),
                tuneLength = 20)
```

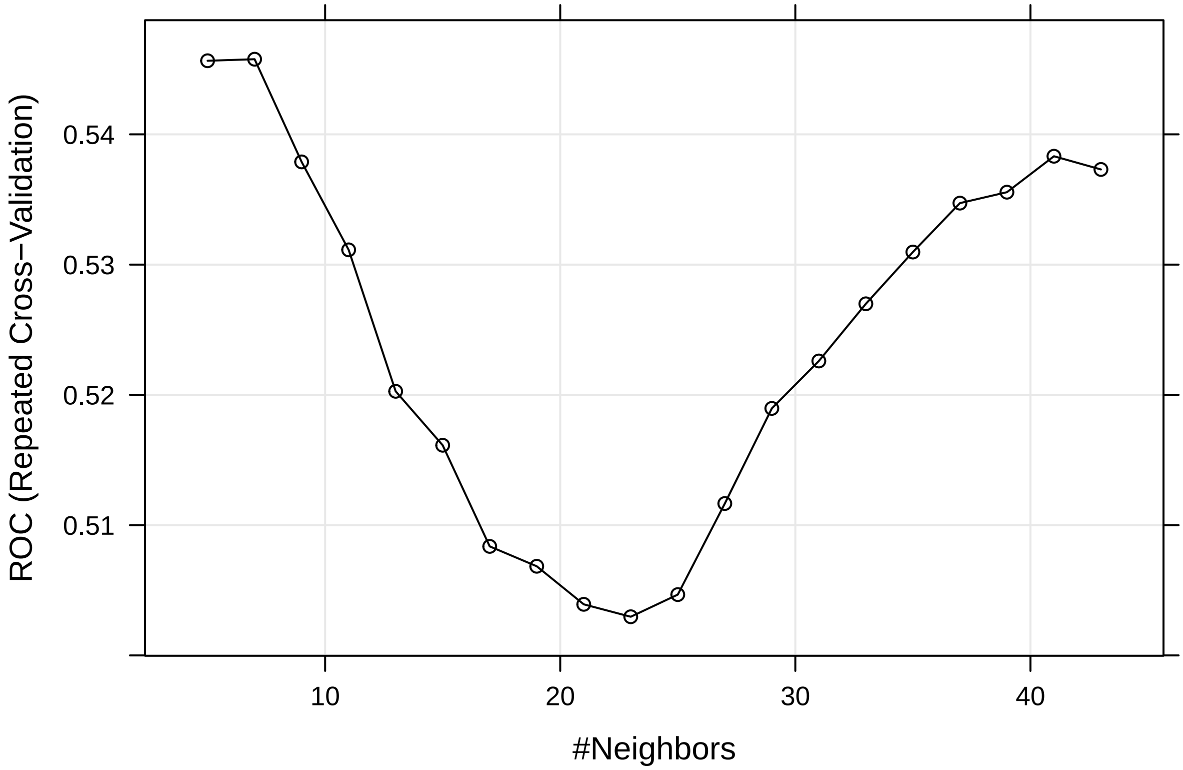
```
## Warning in train.default(x, y, weights = w, ...): The metric "Accuracy" was
## not in the result set. ROC will be used instead.
```

```
knnFit
```

```
## k-Nearest Neighbors
##
## 985 samples
##   2 predictor
##   2 classes: 'Down', 'Up'
##
```

```
## Pre-processing: centered (2), scaled (2)
## Resampling: Cross-Validated (10 fold, repeated 3 times)
## Summary of sample sizes: 887, 887, 886, 886, 887, 887, ...
## Resampling results across tuning parameters:
##
##   k   ROC        Sens       Spec
##    5  0.5456473  0.4467340  0.6317508
##    7  0.5457662  0.4225253  0.6391807
##    9  0.5378838  0.3998653  0.6495735
##   11  0.5311326  0.3869697  0.6441639
##   13  0.5202741  0.3612963  0.6477890
##   15  0.5161328  0.3589899  0.6421437
##   17  0.5083680  0.3597811  0.6446128
##   19  0.5068425  0.3642256  0.6380022
##   21  0.5039217  0.3521717  0.6361279
##   23  0.5029667  0.3545455  0.6508754
##   25  0.5046668  0.3523064  0.6594388
##   27  0.5116593  0.3469697  0.6686195
##   29  0.5189606  0.3431145  0.6648822
##   31  0.5226016  0.3445791  0.6691582
##   33  0.5269912  0.3415993  0.6630415
##   35  0.5309649  0.3347811  0.6783838
##   37  0.5347202  0.3446465  0.6795511
##   39  0.5355586  0.3370707  0.6923906
##   41  0.5383194  0.3423906  0.6887430
##   43  0.5373038  0.3408249  0.6934792
##
## ROC was used to select the optimal model using the largest value.
## The final value used for the model was k = 7.
```

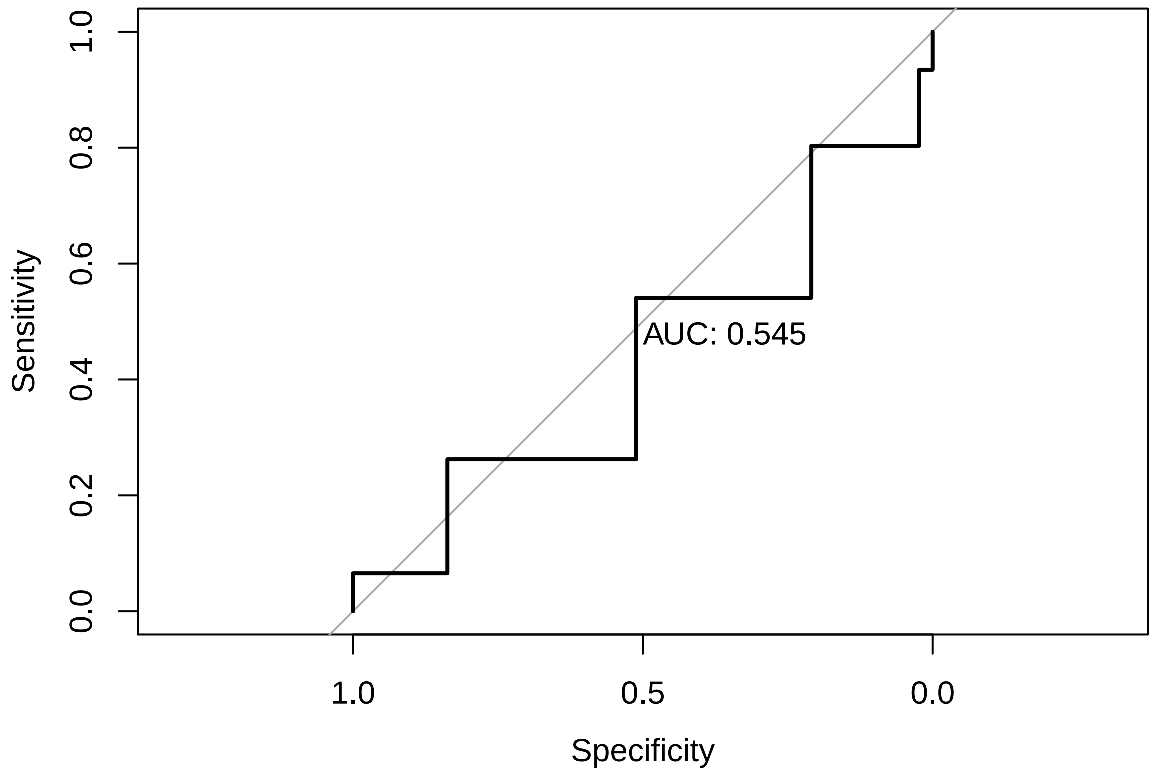
```
plot(knnFit)
```




```r
# K=7 has the highest accuracy rate
knnPredict <- predict(knnFit,newdata = test , type="prob")
knnROC <- roc(test$Direction, knnPredict[,"Down"])
knnROC
```

```
##
## Call:
## roc.default(response = test$Direction, predictor = knnPredict[,     "Down"])
##
## Data: knnPredict[, "Down"] in 43 controls (test$Direction Down) > 61 cases (test$Direction Up).
## Area under the curve: 0.5448
```

```r
plot(knnROC, type="S", print.auc = TRUE)
```



AUC: 0.545
Sensitivity
Specificity
1.0
0.8
0.6
0.4
0.2
0.0
1.0
0.5
0.0