hw3

Benjamin Panny

2023-12-16

Theory

In decision trees for classification,

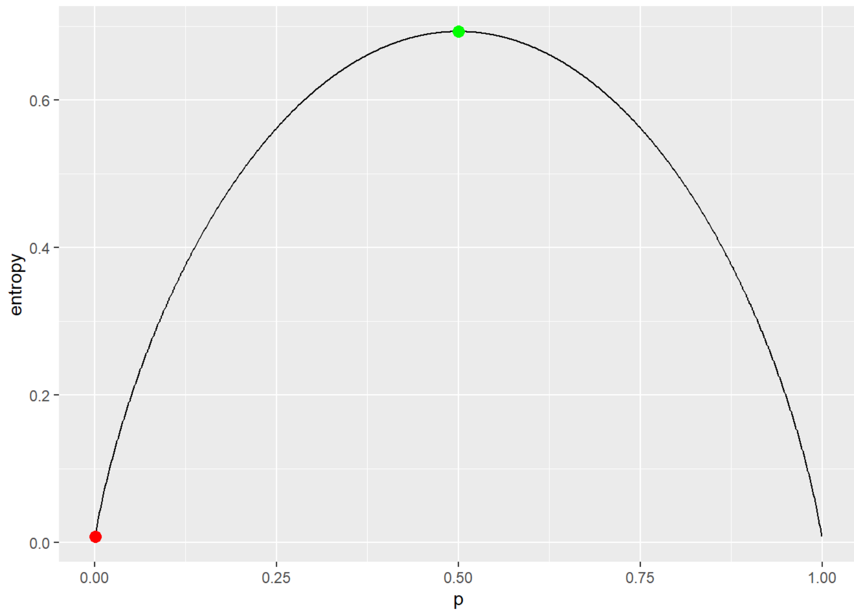
In decision trees for classification, we need to select an impurity function to determine the best split to construct the tree. Gini index and entropy are two most common choices. In order for them to be a valid impurity function, show that the functions

Gini index: $\phi(p) = \sum_{k \neq l} p_k \cdot p_l = 1 - \sum_k p_k(1 - p_k)$Entropy: $\phi(p) = -\sum_k p_k \cdot \log(p_k)$take the maximum (most impure) value when $p_1 = \cdots = p_k = 1/K$take the minimum value when probability c

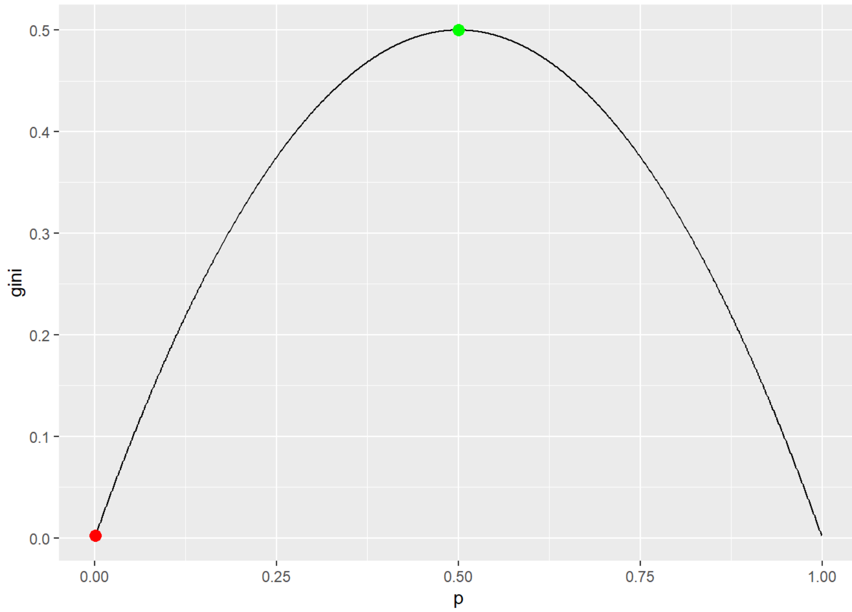
```r
library(tidyverse)
```

```
## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
## ✔ dplyr     1.1.2     ✔ readr     2.1.4
## ✔ forcats   1.0.0     ✔ stringr   1.5.0
## ✔ ggplot2   3.4.2     ✔ tibble    3.2.1
## ✔ lubridate 1.9.2     ✔ tidyr     1.3.0
## ✔ purrr     1.0.1
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```r
classes <- tibble(p = seq(0.001, 0.999, by = 0.001), q = 1 - p)
normalizer <- sum(classes)

get_entropy <- function(p_seq){
  -sum(p_seq * log*p_seq)
}

get_entropy2d <- function(p, q){
  -(p * log (p) + q * log(q))
}

get_gini2d <- function(p, q){
  2 * (p * q)
}

classes %>%
  mutate(entropy = get_entropy2d(p, q),
         log_entropy = log(entropy),) %>%
  ggplot(aes(x = p, y = entropy)) +
  geom_line() +
  geom_point(size = 3, color = 'green', aes(x = p[which.max(entropy)], y = max(entropy))) +
  geom_point(size = 3, color = 'red', aes(x = p[which.min(entropy)], y = min(entropy)))
```



```
classes %>%
    mutate(gini = get_gini2d(p, q),
           log_gini = log(gini)) %>%
    ggplot(aes(x = p, y = gini)) +
    geom_line() +
    geom_point(size = 3, color = 'green', aes(x = p[which.max(gini)], y = max(gini))) +
    geom_point(size = 3, color = 'red', aes(x = p[which.min(gini)], y = min(gini)))
```



As can be seen in green, the max entropy is at p = 0.5, q = 0.5. As can be seen in red, entropy approaches its minimum as p approach 0 and q approaches 1 and vice-versa.

$$\text{Gini index: } \phi(p) = G(p) = \sum_{k \neq l} p_k \cdot p_l = 1 - \sum_k p_k(1 - p_k) \text{Entropy: } \phi(p) = H(p) = -\sum_k p_k \cdot \log(p_k)$$

Binary Entropy Maximization.

$$\arg\max H(p) - p(\log(p)) - (1-p)\log(1-p) dH/dp = -log(p) - p/p + log(1-p) - (1-p)/(1-p)(0-1) dH/dp = log(1-p) - log(p) = 0 log(1-p) = log(p) 1 - p = p p = 0.5$$

Entropy Maximizing in K dimensions by Impurity

$$\arg\max_p H(p) = -\sum_k p_k \cdot \log(p_k) + \lambda\left(\sum_k p_k - 1\right) dH/dp_i = -log(p_i) - 1 + \lambda = 0 \implies p_i = e^{-1+\lambda} dH/d\lambda = \sum_k p_k - 1 = 0 \implies \sum_k p_k = 1 \implies p_i = \frac{e^{-1+\lambda}}{Ke^{-1+\lambda}} = \frac{1}{K}$$

Entropy Minimizing in K dimensions by Purity

$$\arg\min_p H(p) = -\sum_k p_k \cdot \log(p_k) \text{Let } p_j = 1\text{, remembering } H(p) \text{ is non-negative by definition} H(p) = -\sum_{i=1}^{k} p_i \log(p_i) = -p_j \log(p_j) - \sum_{i \neq j} p_i \log(p_i) = -1\log(1) - \sum_{i \neq j} 0 \log(0) = 0$$

Gini Maximization

$$\text{Gini index: } \phi(p) = G(p) = \sum_{k \neq l} p_k \cdot p_l = 1 - \sum_k p_k(1 - p_k) \arg\max_p G(p) = 1 - \sum_k p_k(1-p_k) - \lambda(\sum_k p_k - 1) dG/dp_i = 0 - 1 + 2p_i - \lambda = 0 \implies p_i = \frac{1+\lambda}{2} dG/d\lambda = \sum_k p_k - 1 = 0 \implies \quad = \sum_k p_k =$$

Gini Minimization follows the same logic as Entropy minimization.

(5 points)

Computing

2. Question 10 in Chapter 4.7 in ISLR.

The question should be answered using the Weekly data set, which is part of the ISLR package. This data contains 1,089 weekly returns for 21 years, from the beginning of 1990 to the end of 2010. Write a data analysis report addressing the following problems. (15 points)

a. Produce some numerical and graphical summaries of the Weekly data. Do there appear to any patterns?

```
library(ISLR)
```

```
## Warning: package 'ISLR' was built under R version 4.3.2
```

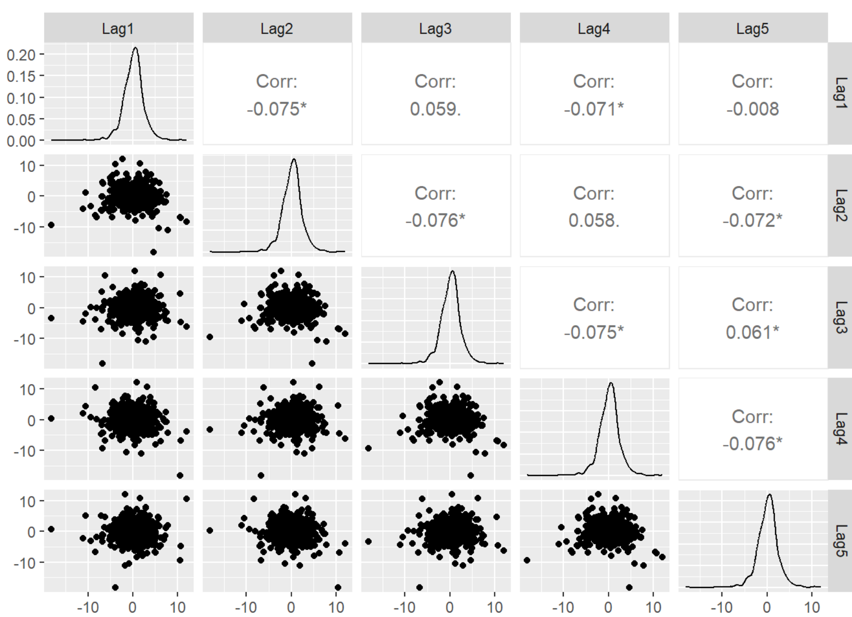
```
weekly <- Weekly
weekly %>% summary()
```

```
##       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            Today
##  Min.   :-18.1950   Min.   :-18.1950   Min.   :0.08747   Min.   :-18.1950
##  1st Qu.: -1.1580   1st Qu.: -1.1660   1st Qu.:0.33202   1st Qu.: -1.1540
##  Median :  0.2380   Median :  0.2340   Median :1.00268   Median :  0.2410
##  Mean   :  0.1458   Mean   :  0.1399   Mean   :1.57462   Mean   :  0.1499
##  3rd Qu.:  1.4090   3rd Qu.:  1.4050   3rd Qu.:2.05373   3rd Qu.:  1.4050
##  Max.   : 12.0260   Max.   : 12.0260   Max.   :9.32821   Max.   : 12.0260
##  Direction
##  Down:484
##  Up  :605
##
##
##
##
```

```
weekly %>% count(Year)
```

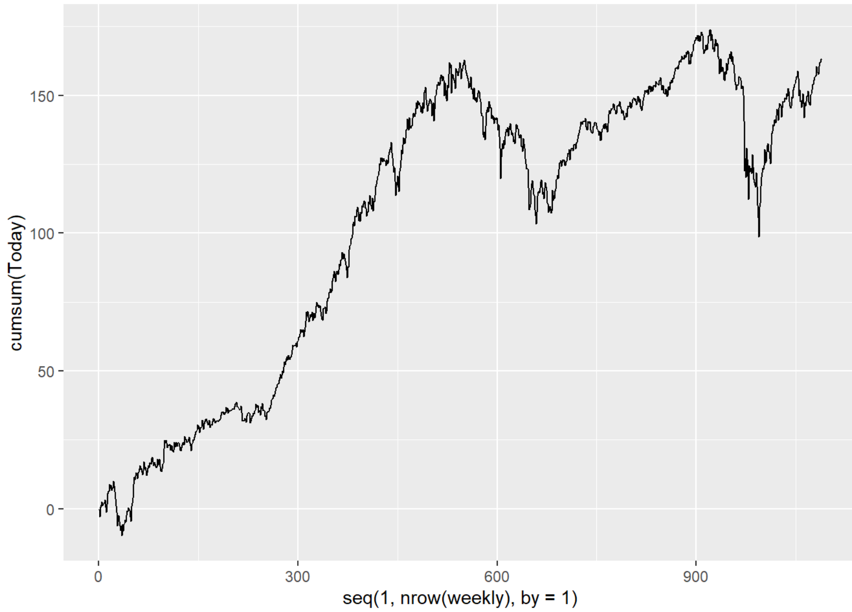
```
##    Year  n
## 1  1990 47
## 2  1991 52
## 3  1992 52
## 4  1993 52
## 5  1994 52
## 6  1995 52
## 7  1996 53
## 8  1997 52
## 9  1998 52
## 10 1999 52
## 11 2000 52
## 12 2001 52
## 13 2002 52
## 14 2003 52
## 15 2004 52
## 16 2005 52
## 17 2006 52
## 18 2007 53
## 19 2008 52
## 20 2009 52
## 21 2010 52
```

```
weekly %>% GGally::ggpairs(columns = 2:6)
```



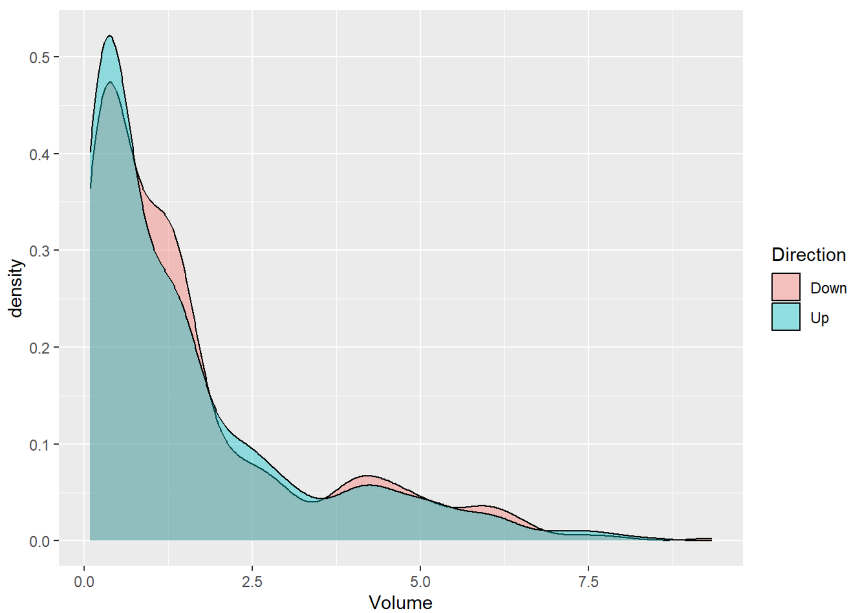
There are some small autocorrelations between the lags. There are roughly 47-52 observations per year from 1990 to 2010. The minimum and maximum for price action is -18 and 12

```
weekly %>%
  ggplot(aes(x = seq(1, nrow(weekly), by = 1), y = cumsum(Today))) +
  geom_line()
```

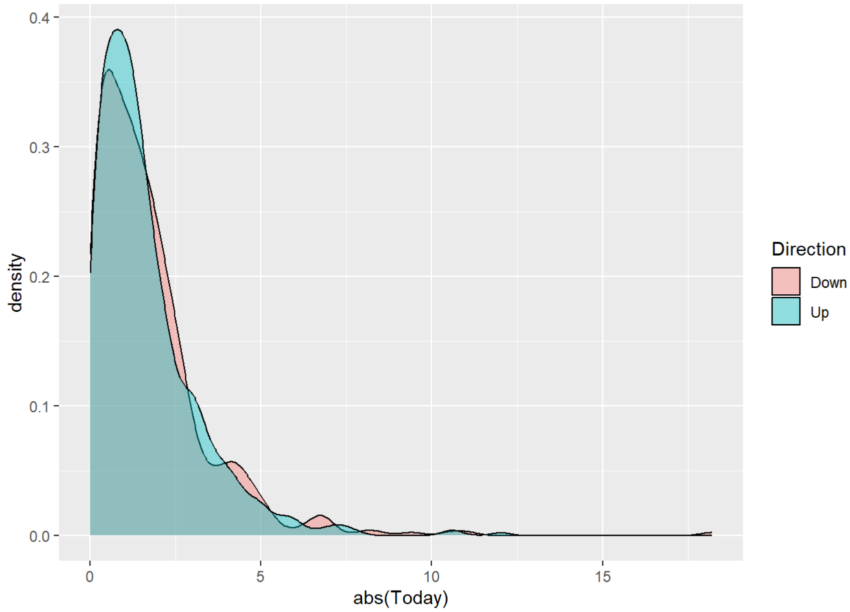


If you invested in this price action in 1990 you'd be doing well, but the 2008 shock might have scared you.

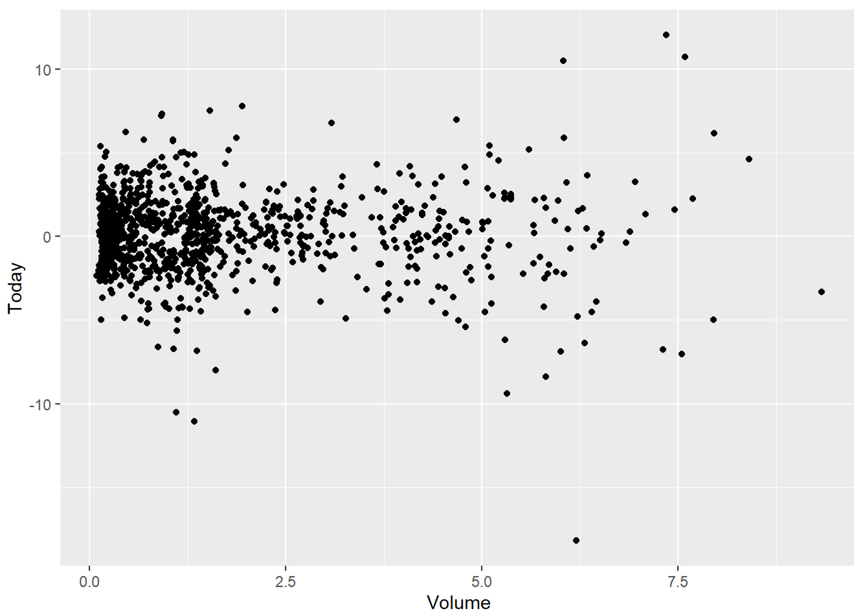
```
weekly %>%
  ggplot(aes(x = Volume, fill = Direction)) +
  geom_density(alpha = .4)
```




```
weekly %>%
  ggplot(aes(x = abs(Today), fill = Direction)) +
  geom_density(alpha=.4)
```




```
weekly %>%
  ggplot(aes(x = Volume, y = Today)) +
  geom_point()
```



There is no obvious, if any, relationship between volume and price action for a week Also, there doesn't seem to be much difference in price action or volume and direction for the week. Most weeks are under 5 in terms of price action, with most under 2.5. Most weeks follow a similar pattern for 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?

```
modb <- weekly %>%
  glm(Direction ~ Lag1 + Lag2 + Lag3 + Lag4 + Lag5 + Volume,
                     data = .,
                     family = 'binomial')

summary(modb)
```

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

Only Lag2 is a statistically significant predictor. The Lag2 coefficient indicates a 1.05x higher odds of an Up day compared to a Down day for each additional unit increase in the value of Lag2 over the previous unit value.

c. Compute the confusion matrix and overall fraction of correct predictions. Explain what the confusion matrix is telling you about the types of mistakes made by logistic regression.

```
cutoff <- .5
pred_dir <- ifelse(predict(modb, type = 'response') > cutoff, 'Up', 'Down')
table(pred_dir, ifelse(modb$y == 1, 'Up', 'Down'))
```

```
##
## pred_dir Down  Up
##     Down   54  48
##     Up    430 557
```

Using a cutoff of 0.5 for the predicted probability, It is clear that the logistic regression model is biased towards predicting “Up” days, regardless of whether the actual day was “Up” or “Down”.

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 the overall fraction of correct predictions for the held out data (that is, the data from 2009 and 2010).

```
weekly_split <- split(weekly, weekly$Year %in% c(2009, 2010))
weekly_train <- weekly_split$`FALSE`
weekly_test <- weekly_split$`TRUE`
mod_glm <- glm(Direction ~ Lag2, family = 'binomial', data = weekly_train)

cutoff <- .5
pred_test_glm <- ifelse(predict(mod_glm, newdata = weekly_test, type = 'response') > cutoff, 'Up', 'Down')
table(pred_test_glm, weekly_test$Direction)
```

```
##
## pred_test_glm Down Up
##          Down    9  5
##          Up     34 56
```

The result follows the same pattern as in c).

e. Repeat (d) using LDA.

```
library(MASS)
```

```
##
## Attaching package: 'MASS'
```

```
## The following object is masked from 'package:dplyr':
##
##     select
```

```
mod_lda <- lda(Direction ~ Lag2, data = weekly_train)
cutoff <- .5
# plot(mod_lda)

# predict the probability
pred_test_lda <- predict(mod_lda, newdata = weekly_test)


head(pred_test_lda$x) #linear discriminants of each observation
```

```
##            LD1
## 986 -0.8059467
## 987  2.9275517
## 988 -2.0198413
## 989 -2.0507404
## 990 -0.9997284
## 991 -0.3786558
```

```
head(pred_test_lda$posterior) # matrix whose kth column contains the posterior probability that the corresponding observatio
n belongs to the kth class
```

```
##          Down        Up
## 986 0.4736555 0.5263445
## 987 0.3558617 0.6441383
## 988 0.5132860 0.4867140
## 989 0.5142948 0.4857052
## 990 0.4799727 0.5200273
## 991 0.4597586 0.5402414
```

```
# head(pred_test_lda$class) #classified using a 50% posterior probability cutoff


# table(pred_test_lda$class) #predicted outcome
table(pred_test_lda$class, weekly_test$Direction) #contingency table of predicted (row) and true (column) outcome
```

```
##
##        Down Up
##   Down    9  5
##   Up     34 56
```

The same pattern is obtained. as in c) and d)

f. Repeat (d) using QDA.

```
qda.fit <- qda(Direction ~ Lag2, data = weekly_train)
qda.fit
```

```
## Call:
## qda(Direction ~ Lag2, data = weekly_train)
##
## Prior probabilities of groups:
##      Down        Up
## 0.4477157 0.5522843
##
## Group means:
##             Lag2
## Down -0.03568254
## Up    0.26036581
```

```
# predict the probability
pred_test_qda <- predict(qda.fit, newdata = weekly_test)
head(pred_test_qda$posterior)
```

```
##          Down        Up
## 986 0.4784630 0.5215370
## 987 0.2693952 0.7306048
## 988 0.4735416 0.5264584
## 989 0.4729118 0.5270882
## 990 0.4802735 0.5197265
## 991 0.4709913 0.5290087
```

```
head(pred_test_qda$class)
```

```
## [1] Up Up Up Up Up Up
## Levels: Down Up
```

```
table(pred_test_qda$class, weekly_test$Direction)
```

```
##
##        Down Up
##   Down    0  0
##   Up     43 61
```

This follows the same pattern in the extreme, as QDA only predicts the “Up” class.

g. Which of these methods appears to provide the best results on this data?

At a cutoff of 0.5, logistic regression and LDA are indistinguishable in their predictive performance. QDA is worse than logistic regression and LDA because it only predicts Up days, which is useless for decision-making.

h. 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.

```
mod_lda_all <- lda(Direction ~ Lag1 + Lag2 + Lag3 + Lag4 + Lag5 + Volume, data = weekly_train)
cutoff <- .5
# plot(mod_lda)

# predict the probability
pred_test_lda_all <- predict(mod_lda_all, newdata = weekly_test)


# head(pred_test_lda_all$x) #linear discriminants of each observation
# head(pred_test_lda_all$posterior) # matrix whose kth column contains the posterior probability that the corresponding obse
rvation belongs to the kth class
# head(pred_test_lda_all$class) #classified using a 50% posterior probability cutoff


# table(pred_test_lda_all$class) #predicted outcome
table(pred_test_lda_all$class, weekly_test$Direction) #contingency table of predicted (row) and true (column) outcome
```

```
##
##        Down Up
##   Down   31 44
##   Up     12 17
```

Including all covariates degrades accuracy of LDA, but makes its predictions more balanced.

```
mod_lda_lag <- lda(Direction ~ Lag1 + Lag2 + Lag3 + Lag4 + Lag5, data = weekly_train)
cutoff <- .5
# plot(mod_lda)

# predict the probability
pred_test_lda_lag <- predict(mod_lda_lag, newdata = weekly_test)


# head(pred_test_lda_lag$x) #linear discriminants of each observation
# head(pred_test_lda_lag$posterior) # matrix whose kth column contains the posterior probability that the corresponding obse
rvation belongs to the kth class
# head(pred_test_lda_lag$class) #classified using a 50% posterior probability cutoff


# table(pred_test_lda_lag$class) #predicted outcome
table(pred_test_lda_lag$class, weekly_test$Direction) #contingency table of predicted (row) and true (column) outcome
```

```
##
##        Down Up
##   Down    9 13
##   Up     34 48
```

Removing Volume info restores the accuracy performance but is biased toward Up predictions again.

```
mod_lda_lag_int <- lda(Direction ~ Lag1*Lag2 + Lag3*Lag4 + Lag5, data = weekly_train)
cutoff <- .5
# plot(mod_lda)

# predict the probability
pred_test_lda_lag_int <- predict(mod_lda_lag_int, newdata = weekly_test)


# head(pred_test_lda_lag_int$x) #linear discriminants of each observation
# head(pred_test_lda_lag_int$posterior) # matrix whose kth column contains the posterior probability that the corresponding
observation belongs to the kth class
# head(pred_test_lda_lag_int$class) #classified using a 50% posterior probability cutoff


# table(pred_test_lda_lag_int$class) #predicted outcome
table(pred_test_lda_lag_int$class, weekly_test$Direction) #contingency table of predicted (row) and true (column) outcome
```

```
##
##        Down Up
##   Down   11 10
##   Up     32 51
```

Adding some random interactions improves accuracy slightly while not really addressing the bias in predicted values.

```
qda_lag_int <- qda(Direction ~ Lag1*Lag2 + Lag3*Lag4 + Lag1*Lag5, data = weekly_train)
qda_lag_int
```

```
## Call:
## qda(Direction ~ Lag1 * Lag2 + Lag3 * Lag4 + Lag1 * Lag5, data = weekly_train)
##
## Prior probabilities of groups:
##      Down        Up
## 0.4477157 0.5522843
##
## Group means:
##              Lag1        Lag2       Lag3       Lag4       Lag5  Lag1:Lag2
## Down  0.289444444 -0.03568254 0.17080045 0.15925624 0.21409297 -0.8014495
## Up   -0.009213235  0.26036581 0.08404044 0.09220956 0.04548897 -0.1393632
##         Lag3:Lag4  Lag1:Lag5
## Down -0.991496916  0.3615693
## Up    0.007162822 -0.1833118
```

```
# predict the probability
pred_test_qda_int <- predict(qda_lag_int, newdata = weekly_test)
head(pred_test_qda_int$posterior)
```

```
##             Down        Up
## 986 7.409976e-01 0.2590024
## 987 2.172569e-05 0.9999783
## 988 9.499288e-02 0.9050071
## 989 1.397212e-05 0.9999860
## 990 2.136856e-02 0.9786314
## 991 5.942302e-01 0.4057698
```

```
head(pred_test_qda_int$class)
```

```
## [1] Down Up   Up   Up   Up   Down
## Levels: Down Up
```

```
table(pred_test_qda_int$class, weekly_test$Direction)
```

```
##
##        Down Up
##   Down   22 31
##   Up     21 30
```

Following the same strategy of random interactions for LDA degrades the accuracy for QDA yet balances the predictions more.

It is not terribly surprising that interactions between lags don't seem to do much because there is only week correlations, if any, between different lags when viewed jointly.

What happens if we include all interactions with statistically significant correlations?

```
mod_glm_int_sig <- glm(Direction ~ Lag1*Lag2 + Lag1*Lag4 + Lag2*Lag3 + Lag2*Lag5 + Lag3*Lag4 + Lag3*Lag5 + Lag4*Lag5, family
= 'binomial', data = weekly_train)

cutoff <- .5
pred_test_glm_int_sig <- ifelse(predict(mod_glm_int_sig, newdata = weekly_test, type = 'response') > cutoff, 'Up', 'Down')
table(pred_test_glm_int_sig, weekly_test$Direction)
```

```
##
## pred_test_glm_int_sig Down Up
##                  Down   10 13
##                  Up     33 48
```

```
mod_lda_lag_int_sig <- lda(Direction ~Lag1*Lag2 + Lag1*Lag4 + Lag2*Lag3 + Lag2*Lag5 + Lag3*Lag4 + Lag3*Lag5 + Lag4*Lag5, dat
a = weekly_train)
cutoff <- .5
# plot(mod_lda)

# predict the probability
pred_test_lda_lag_int_sig <- predict(mod_lda_lag_int_sig, newdata = weekly_test)


# head(pred_test_lda_lag_int_sig$x) #linear discriminants of each observation
# head(pred_test_lda_lag_int_sig$posterior) # matrix whose kth column contains the posterior probability that the correspond
ing observation belongs to the kth class
# head(pred_test_lda_lag_int_sig$class) #classified using a 50% posterior probability cutoff


# table(pred_test_lda_lag_int_sig$class) #predicted outcome
table(pred_test_lda_lag_int_sig$class, weekly_test$Direction) #contingency table of predicted (row) and true (column) outcom
e
```

```
##
##        Down Up
##   Down   10 13
##   Up     33 48
```

```
qda_lag_int_sig <- qda(Direction ~ Lag1*Lag2 + Lag1*Lag4 + Lag2*Lag3 + Lag2*Lag5 + Lag3*Lag4 + Lag3*Lag5 + Lag4*Lag5, data =
weekly_train)
# qda_lag_int_sig

# predict the probability
pred_test_qda_int_sig <- predict(qda_lag_int_sig, newdata = weekly_test)
# head(pred_test_qda_int_sig$posterior)
# head(pred_test_qda_int_sig$class)

table(pred_test_qda_int_sig$class, weekly_test$Direction)
```

```
##
##        Down Up
##   Down   13 11
##   Up     30 50
```

This results in the best performance for QDA and this QDA model is the best performing model overall according to accuracy by the confusion matrix with a cutoff of 0.5

```
library(pROC) # build a ROC curve
```

```
## Type 'citation("pROC")' for a citation.
```

```
##
## Attaching package: 'pROC'
```

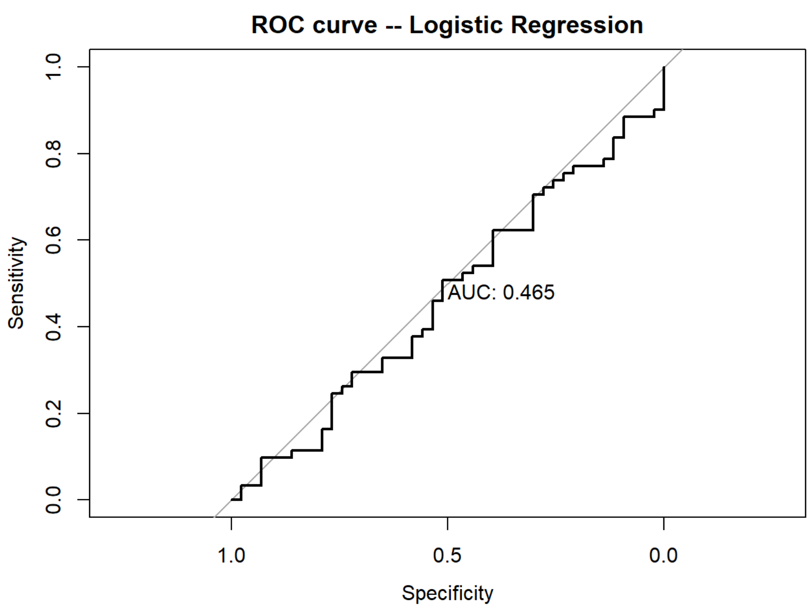
```
## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

```
par(las=F);par(mfrow=c(1,1))
roc.glm <- roc(weekly_test$Direction, predict(mod_glm_int_sig, newdata = weekly_test, type = 'response'))
```

```
## Setting levels: control = Down, case = Up
```

```
## Setting direction: controls > cases
```

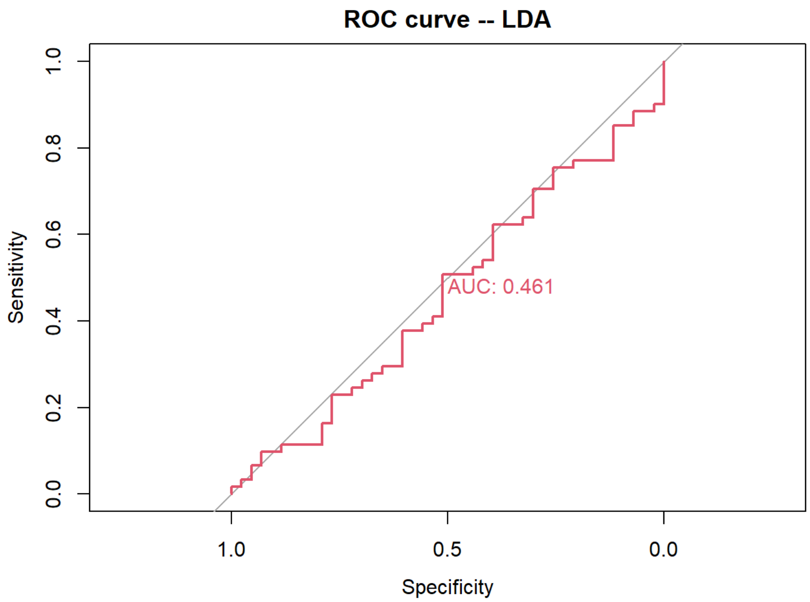
```
plot(roc.glm,  col=1, print.auc=TRUE,main ="ROC curve -- Logistic Regression")
```




```
roc.lda <- roc(weekly_test$Direction, as.numeric(pred_test_lda_lag_int_sig$x))
```

```
## Setting levels: control = Down, case = Up
## Setting direction: controls > cases
```

```
plot(roc.lda,  print.auc=TRUE,
     col=2,main ="ROC curve -- LDA" )
```

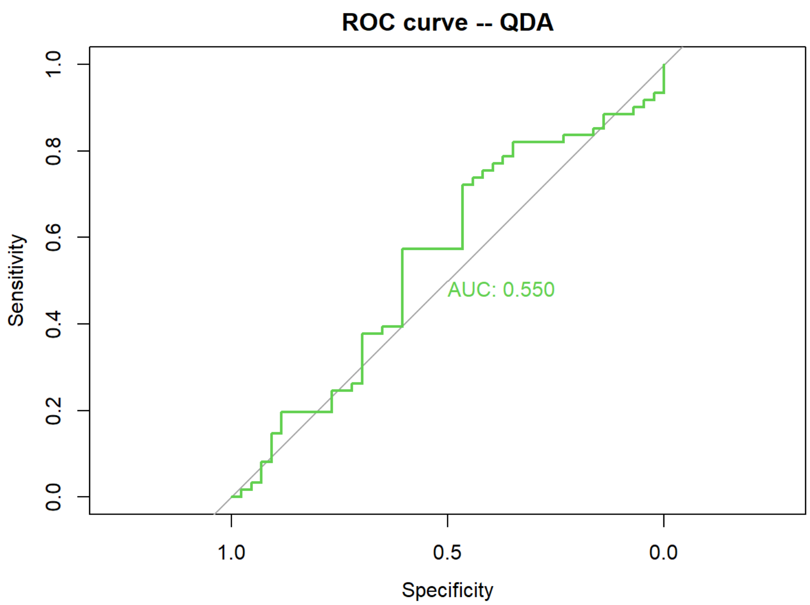



```
roc.qda <- roc(weekly_test$Direction, pred_test_qda_int_sig$posterior[,2])
```

```
## Setting levels: control = Down, case = Up
```

```
## Setting direction: controls < cases
```

```
plot(roc.qda, print.auc=TRUE, col=3,main ="ROC curve -- QDA"  )
```



I don't think I would trust any of these models with my money.