Assignment 4

Brian Morales

September 22, 2022

Part E

```
library(readr)
library(MASS)
Weekly <- read_csv("~/Desktop/Fall-2022/Stats-Learning/ALL-CSV-FILES/Weekly.csv", show_col_types = FALS
Weekly$Direction = as.factor(Weekly$Direction)
```

```
train <- (Weekly$Year < 2009)
Weekly.2009 <- Weekly[!train, ]
Direction.2010 <- Weekly$Direction[!train]

lda.fit <- lda(Direction ~ Lag2, Weekly, subset = train)
lda.fit
```

```
## Call:
## lda(Direction ~ Lag2, data = Weekly, subset = train)
##
## Prior probabilities of groups:
##      Down        Up
## 0.4477157 0.5522843
##
## Group means:
##              Lag2
## Down -0.03568254
## Up    0.26036581
##
## Coefficients of linear discriminants:
##             LD1
## Lag2 0.4414162
```

```
lda.pred <- predict(lda.fit, Weekly.2009, type='response')
lda.class <- lda.pred$class
table(lda.class, Direction.2010)
```

```
##          Direction.2010
## lda.class Down Up
##      Down    9  5
##      Up     34 56
```

```
65/104
```

```
## [1] 0.625
```

LDA presents information about the mean of Lag2, showing that $\hat{\pi_1} = 0.447$ and $\hat{\pi_2} = 0.552$. This means that 44.7% of the training observations correspond to days during which the market went down and 55.2% correspond to days during the market went up. The model will predict correctly 62.5% of the time, the same as our previous model

Part F

```
qda.fit <- qda(Direction ~ Lag2, Weekly, subset = train)
qda.class <- predict(qda.fit, Weekly.2009)$class
table(qda.class, Direction.2010)
```

```
##          Direction.2010
## qda.class Down Up
##      Down    0  0
##      Up     43 61
```

```
61/104
```

```
## [1] 0.5865385
```

The QDA model predicts 58.6 correct. This is lower than the previous model of 62.5.

Part G

```
library(class)
train.X = cbind(Weekly$Lag2)[train, ]
test.X = cbind(Weekly$Lag2)[!train, ]
train.X = cbind(train.X)
test.X = cbind(test.X)
train.Direction = Weekly$Direction[train]

set.seed(1)
knn.pred <- knn(train.X, test.X, train.Direction, k = 1)
table(knn.pred, Direction.2010)
```

```
##         Direction.2010
## knn.pred Down Up
##     Down   21 30
##     Up     22 31
```

```
52/104
```

```
## [1] 0.5
```

When $K = 1$ our chances are the same as just randomly guessing.

Part H

```
library(e1071)
nb.fit <- naiveBayes(Direction ~ Lag2, data = Weekly, subset = train)
nb.class <- predict(nb.fit, Weekly.2009)
table(nb.class, Direction.2010)
```

```
##          Direction.2010
## nb.class Down Up
##     Down    0  0
##     Up     43 61
```

```
59/104
```

```
## [1] 0.5673077
```

The result for our Naive Bayes model predicts 56.7% correct.

Part I

The best result are LDA and GLM; both coming in a tie, 62.5. The other classifiers did slightly lower than LDA and GLM with KNN doing the worst, 0.5

Part J

```
set.seed(1)
knn.pred <- knn(train.X, test.X, train.Direction, k = 145)
table(knn.pred, Direction.2010)
```

```
##          Direction.2010
## knn.pred Down Up
##     Down    9  4
##     Up     34 57
```

```
66/104
```

```
## [1] 0.6346154
```

When $K = 4$ our accuracy increases by 10%. Fairing much better than when $K = 1$.

```
lda.fit <- lda(Direction ~ Lag1 + Lag2, Weekly, subset = train)
lda.pred <- predict(lda.fit, Weekly.2009, type='response')
lda.class <- lda.pred$class
table(lda.class, Direction.2010)
```

```
##           Direction.2010
## lda.class Down Up
##      Down    7  8
##      Up     36 53
```

```r
60/104
```

```
## [1] 0.5769231
```

Adding `Lag1` to our LDA model reduced our accuracy.

```r
lda.fit = lda(Direction ~ Lag2:Lag1, data = Weekly, subset = train)
lda.pred <- predict(lda.fit, Weekly.2009, type='response')
lda.class <- lda.pred$class
table(lda.class, Direction.2010)
```

```
##          Direction.2010
## lda.class Down Up
##      Down    0  1
##      Up     43 60
```

```r
60/104
```

```
## [1] 0.5769231
```

Applying `Lag1` as an interaction term we still return the same accuracy as above, however, our confusion matrix is different. The model predicted 0 correct for the market declining and 60 correct of the market increasing.

```r
lda.fit = lda(Direction ~ Lag2 + log(abs(Lag2)), data = Weekly, subset = train)
lda.class <- predict(lda.fit, Weekly.2009)$class
table(lda.class, Direction.2010)
```

```
##          Direction.2010
## lda.class Down Up
##      Down    6  3
##      Up     37 58
```

```r
64/104
```

```
## [1] 0.6153846
```

Add a `log` transformation to `Lag2` causes a rounded 1% decrease in our accuracy.