# ['STAT 4610] HW-4

Michael Ghattas

9/24/2022

## Chapter - 4

### Problem - 13

```
library(ISLR)
library(corrplot)

## corrplot 0.92 loaded

library(MASS)

## Warning: package 'MASS' was built under R version 4.1.2

library(class)

## Warning: package 'class' was built under R version 4.1.2

library(e1071)

## Warning: package 'e1071' was built under R version 4.1.2
```

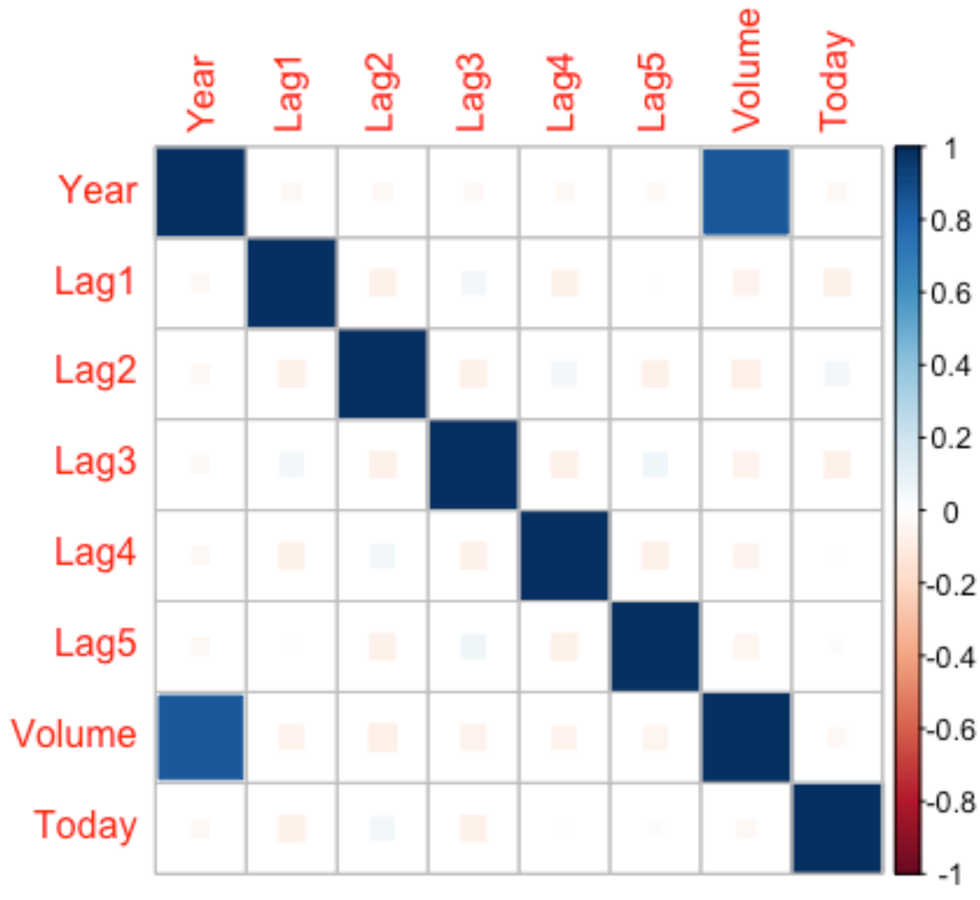
#### Part - (a)

```
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             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
##
##
##
##
```

```
corrplot(cor(Weekly[, -9]), method = "square")
```




-> Year and Volume are the variables that seem to have a significant linear relation.

### Part - (b)

```
Weekly.fit <- glm(Direction ~ Lag1 + Lag2 + Lag3 + Lag4 + Lag5 + Volume, data
= Weekly, family = binomial)
summary(Weekly.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
## ---
## 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 seems to be the only variable that has statistical significant at the level of significance.

### Part - (c)

```
logWeekly.prob = predict(Weekly.fit, type = 'response')
logWeekly.pred = rep("Down", length(logWeekly.prob))
logWeekly.pred[logWeekly.prob > 0.5] = "Up"

table(logWeekly.pred, Weekly$Direction)

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

-> The model predicted the weekly market trend correctly 56.11% of the time.

-> $\frac{54+557}{54+48+430+557} = 0.5611$

-> The model correctly predicted the Upward weekly trends 92.07% of the time.

-> $\frac{557}{48+557} = 0.9207$

-> The model correctly predicted the Downward weekly trends 11.15% of the time.

-> $\frac{54}{54+430} = 0.1115$

### part - (d)

```
Direction = Weekly$Direction
train = (Weekly$Year < 2009)
test <- Weekly[!train, ]

Weekly.fit <- glm(Direction ~ Lag2, data = Weekly, family = binomial, subset
= train)
```

```
logWeekly.prob = predict(Weekly.fit, test, type = "response")
logWeekly.pred = rep("Down", length(logWeekly.prob))
logWeekly.pred[logWeekly.prob > 0.5] = "Up"
Direction.test = Direction[!train]

table(logWeekly.pred, Direction.test)

##                Direction.test
## logWeekly.pred Down Up
##           Down    9  5
##           Up     34 56

mean(logWeekly.pred == Direction.test)

## [1] 0.625
```

-> The model correctly predicted weekly trends at rate of 62.5% of the time.

-> The model predicted upward trends 91.80% of the time.

-> The model predicted downward trends 20.93%of the time.

### part - (e)

```
WeeklyLDA.fit <- lda(Direction ~ Lag2, data = Weekly, family = binomial,
subset = train)
WeeklyLDA.pred <- predict(WeeklyLDA.fit, test)

table(WeeklyLDA.pred$class, Direction.test)

##       Direction.test
##        Down Up
##   Down    9  5
##   Up     34 56

mean(WeeklyLDA.pred$class == Direction.test)

## [1] 0.625
```

-> The Linear Discriminant Analysis (LDA) classifying model results are identical to the logistic regression model from part (e).

### part - (f)

```
WeeklyQDA.fit <- qda(Direction ~ Lag2, data = Weekly, subset = train)
WeeklyQDA.pred <- predict(WeeklyQDA.fit, test)

table(WeeklyQDA.pred$class, Direction.test)

##       Direction.test
##        Down Up
##   Down    0  0
##   Up     43 61

mean(WeeklyQDA.pred$class == Direction.test)

## [1] 0.5865385
```

-> The Quadratic Linear Analysis (QDA) model has 58.65% accuracy, which is lower than LDA, which has an accuracy of 62.5%.

-> The QDA model only predicting the correctness of weekly upward trends while missing the downward weekly trends.

### part - (g)

```
Week.train = as.matrix(Weekly$Lag2[train])
Week.test = as.matrix(Weekly$Lag2[!train])
Direction.train = Direction[train]

set.seed(111)
WeekKNN.pred = knn(Week.train, Week.test, Direction.train, k = 1)

table(WeekKNN.pred, Direction.test)

##              Direction.test
## WeekKNN.pred Down Up
##         Down   21 30
##         Up     22 31
```

```
mean(WeekKNN.pred == Direction.test)

## [1] 0.5
```

-> The K-Nearest Neighbors (KNN) model resulted in a classifying model has ~51% accuracy.

-> The KNN model has the lowest accuracy.

### part - (h)

```
WeeklyNB.fit <- naiveBayes(Direction ~ Lag2, data = Weekly, subset = train)
WeeklyNB.pred <- predict(WeeklyNB.fit, test)

table(WeeklyNB.pred, Direction.test)

##              Direction.test
## WeeklyNB.pred Down Up
##          Down    0  0
##          Up     43 61

mean(WeeklyNB.pred == Direction.test)

## [1] 0.5865385
```

-> The Naive Bayes (NB) model has 58.65% accuracy, which is identical to the QDA model from part (f).

-> Like the QDA model, the NB model also only predicting the correctness of weekly upward trends while missing the downward weekly trends.

### part - (i)

-> Both the Logistic Regression model and LDA model have the best accuracy rate of 62.5%.

### part - (j)

```
#Logistic Regression with Lag2
Weekly.fit <- glm(Direction ~ Lag2, data = Weekly, family = binomial, subset
= train)
```

```r
logWeekly.prob = predict(Weekly.fit, test, type = "response")
logWeekly.pred = rep("Down", length(logWeekly.prob))
logWeekly.pred[logWeekly.prob > 0.5] = "Up"
Direction.test = Direction[!train]

table(logWeekly.pred, Direction.test)

##                Direction.test
## logWeekly.pred Down Up
##           Down    9  5
##           Up     34 56

mean(logWeekly.pred == Direction.test)

## [1] 0.625

#LDA with Lag2
WeeklyLDA.fit <- lda(Direction ~ Lag2, data = Weekly,family = binomial,
subset = train)
WeeklyLDA.pred <- predict(WeeklyLDA.fit, test)

table(WeeklyLDA.pred$class, Direction.test)

##       Direction.test
##        Down Up
##   Down    9  5
##   Up     34 56

mean(WeeklyLDA.pred$class == Direction.test)

## [1] 0.625

#QDA with with the 2nd power polynomial of Lag2
WeeklyQDA.fit = qda(Direction ~ poly(Lag2, 2), data = Weekly, subset = train)
WeeklyQDA.pred = predict(WeeklyQDA.fit, test)

table(WeeklyQDA.pred$class, Direction.test)

##       Direction.test
##        Down Up
```

```
##   Down    7  3
##   Up     36 58
```

```
mean(WeeklyQDA.pred$class == Direction.test)
```

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

```
#KNN with Lag2 & K = 10
Week.train = as.matrix(Weekly$Lag2[train])
Week.test = as.matrix(Weekly$Lag2[!train])
Direction.train = Direction[train]

set.seed(222)
WeekKNN.pred = knn(Week.train, Week.test, Direction.train, k = 10)
```

```
table(WeekKNN.pred, Direction.test)
```

```
##             Direction.test
## WeekKNN.pred Down Up
##         Down   17 18
##         Up     26 43
```

```
mean(WeekKNN.pred == Direction.test)
```

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

```
#KNN with Lag2 & K = 100
Week.train = as.matrix(Weekly$Lag2[train])
Week.test = as.matrix(Weekly$Lag2[!train])
Direction.train = Direction[train]

set.seed(222)
WeekKNN.pred = knn(Week.train, Week.test, Direction.train, k = 100)
```

```
table(WeekKNN.pred, Direction.test)
```

```
##             Direction.test
## WeekKNN.pred Down Up
##         Down    9 12
##         Up     34 49
```

```r
mean(WeekKNN.pred == Direction.test)

## [1] 0.5576923
```

-> The Logistic Regression, LDA, and QDA(lag2^2) models have the best accuracy rate of 62.5%.

-> While there where some improvement in accuracy with the KNN (k= 10, k = 100) models, their accuracy remains lower.

## End.