Math448: Chapter 4 HW

Zahraa Alshalal

2023-04-03

library(tidyverse)

## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──  
## ✔ dplyr 1.1.0 ✔ readr 2.1.4  
## ✔ forcats 1.0.0 ✔ stringr 1.5.0  
## ✔ ggplot2 3.4.1 ✔ tibble 3.2.0  
## ✔ 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 ]8;;http://conflicted.r-lib.org/conflicted package]8;; to force all conflicts to become errors

library(ISLR2)  
library(ISLR)

##   
## Attaching package: 'ISLR'  
##   
## The following objects are masked from 'package:ISLR2':  
##   
## Auto, Credit

library(MASS)

##   
## Attaching package: 'MASS'  
##   
## The following object is masked from 'package:ISLR2':  
##   
## Boston  
##   
## The following object is masked from 'package:dplyr':  
##   
## select

library(class)  
library(e1071)

## Conceptual Questions

### Exercise 3:

* The Bayes classifier is not linear: quadratic
* in the final term, the discriminant is not linear.

### Exercise 6:

## Applied Questions:

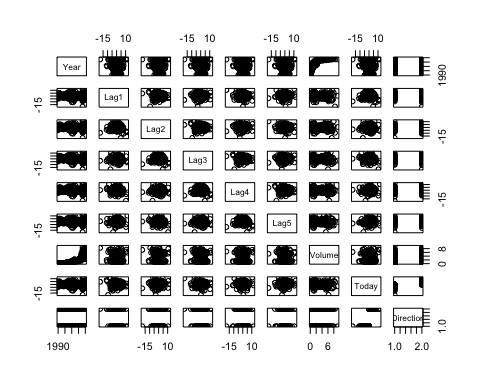
### Exercise 13:

#### 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   
##   
##   
##   
##

# Scatterplot matrix.  
pairs(Weekly)

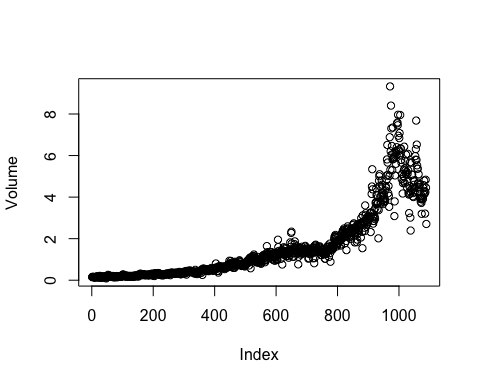


# Correlation matrix.  
cor(Weekly[, -9])

## Year Lag1 Lag2 Lag3 Lag4  
## Year 1.00000000 -0.032289274 -0.03339001 -0.03000649 -0.031127923  
## Lag1 -0.03228927 1.000000000 -0.07485305 0.05863568 -0.071273876  
## Lag2 -0.03339001 -0.074853051 1.00000000 -0.07572091 0.058381535  
## Lag3 -0.03000649 0.058635682 -0.07572091 1.00000000 -0.075395865  
## Lag4 -0.03112792 -0.071273876 0.05838153 -0.07539587 1.000000000  
## Lag5 -0.03051910 -0.008183096 -0.07249948 0.06065717 -0.075675027  
## Volume 0.84194162 -0.064951313 -0.08551314 -0.06928771 -0.061074617  
## Today -0.03245989 -0.075031842 0.05916672 -0.07124364 -0.007825873  
## Lag5 Volume Today  
## Year -0.030519101 0.84194162 -0.032459894  
## Lag1 -0.008183096 -0.06495131 -0.075031842  
## Lag2 -0.072499482 -0.08551314 0.059166717  
## Lag3 0.060657175 -0.06928771 -0.071243639  
## Lag4 -0.075675027 -0.06107462 -0.007825873  
## Lag5 1.000000000 -0.05851741 0.011012698  
## Volume -0.058517414 1.00000000 -0.033077783  
## Today 0.011012698 -0.03307778 1.000000000

* The correlations between the “lag” variables and today’s returns are close to zero. There is correlation is between “Year” and “Volume”. No other patterns are discernible.

attach(Weekly)  
plot(Volume)



* The “Volume” plot shows that it is increasing over time.

#### b)

fit.glm = glm(Direction ~ Lag1 + Lag2 + Lag3 + Lag4 + Lag5 + Volume, data = Weekly, family = binomial)  
summary(fit.glm)

##   
## 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

* Lag 2 appears to have some statistical significance with a p-value is less than 0.05.

#### c)

probs = predict(fit.glm, type = "response")  
pred.glm = rep("Down", length(probs))  
pred.glm[probs > 0.5] = "Up"  
table(pred.glm, Direction)

## Direction  
## pred.glm Down Up  
## Down 54 48  
## Up 430 557

#the percentage of correct predictions on the training data  
correct\_predictions = (54+557)/(54+557+48+430)  
correct\_predictions

## [1] 0.5610652

error\_predictions = 1 - correct\_predictions  
error\_predictions

## [1] 0.4389348

market\_up = (557)/(557+48)  
market\_up

## [1] 0.9206612

market\_down = 54/(54+430)  
market\_down

## [1] 0.1115702

* We may conclude that the percentage of correct predictions on the training data is 56.11%. The training error rate 43.89%. For the 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 only 11.16%.

#### d)

train = (Year < 2009)  
Weekly.910 = Weekly[!train, ]  
Direction.910 = Direction[!train]  
fit.glm2 = glm(Direction ~ Lag2, data = Weekly, family = binomial, subset = train)  
summary(fit.glm2)

##   
## Call:  
## glm(formula = Direction ~ Lag2, family = binomial, data = Weekly,   
## subset = train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.536 -1.264 1.021 1.091 1.368   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 0.20326 0.06428 3.162 0.00157 \*\*  
## Lag2 0.05810 0.02870 2.024 0.04298 \*   
## ---  
## 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: 1350.5 on 983 degrees of freedom  
## AIC: 1354.5  
##   
## Number of Fisher Scoring iterations: 4

probs2 = predict(fit.glm2, Weekly.910, type = "response")  
pred.glm2 = rep("Down", length(probs2))  
pred.glm2[probs2 > 0.5] <- "Up"  
table(pred.glm2, Direction.910)

## Direction.910  
## pred.glm2 Down Up  
## Down 9 5  
## Up 34 56

#the percentage of correct predictions on the training data  
correct\_predictions1 = (9+56)/(104)  
correct\_predictions1

## [1] 0.625

error\_predictions1 = 1 - correct\_predictions1  
error\_predictions1

## [1] 0.375

market\_up1 = (56)/(56+5)  
market\_up1

## [1] 0.9180328

market\_down1 = 9/(9+34)  
market\_down1

## [1] 0.2093023

* In this case, the percentage of correct predictions on the test data is 62.5%. the test error rate is 37.5%. For the weeks when the market goes up, the model is right 91.80% of the time. For the weeks when the market goes down, the model is right only 20.93% of the time.

#### e)

lda.fit = lda(Direction ~ Lag2, data = 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.910)  
table(lda.pred$class, Direction.910)

## Direction.910  
## Down Up  
## Down 9 5  
## Up 34 56

mean(lda.pred$class == Direction.910)

## [1] 0.625

* The percentage of correct predictions on the test data is 62.5%.

#### f)

qda.fit = qda(Direction ~ Lag2, data = Weekly, subset = train)  
qda.fit

## Call:  
## qda(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

pred.qda = predict(qda.fit, Weekly.910)$class  
table(pred.qda, Direction.910)

## Direction.910  
## pred.qda Down Up  
## Down 0 0  
## Up 43 61

mean(pred.qda == Direction.910)

## [1] 0.5865385

* The percentage of correct predictions on the test data is 58.7%.

#### g)

train.X = as.matrix(Lag2[train])  
test.X = as.matrix(Lag2[!train])  
train.Direction = Direction[train]  
set.seed(1)  
knn.pred = knn(train.X, test.X, train.Direction, k = 1)  
table(knn.pred, Direction.910)

## Direction.910  
## knn.pred Down Up  
## Down 21 30  
## Up 22 31

mean(knn.pred == Direction.910)

## [1] 0.5

* The percentage of correct predictions on the test data is 50%.

#### h)

bayes.fit=naiveBayes(Direction~Lag2 ,data=Weekly ,subset=train)  
bayes.fit

##   
## Naive Bayes Classifier for Discrete Predictors  
##   
## Call:  
## naiveBayes.default(x = X, y = Y, laplace = laplace)  
##   
## A-priori probabilities:  
## Y  
## Down Up   
## 0.4477157 0.5522843   
##   
## Conditional probabilities:  
## Lag2  
## Y [,1] [,2]  
## Down -0.03568254 2.199504  
## Up 0.26036581 2.317485

bayes.pre = predict(bayes.fit, Weekly.910)  
table(bayes.pre, Direction.910)

## Direction.910  
## bayes.pre Down Up  
## Down 0 0  
## Up 43 61

correct\_predictions1 = (61)/(104)  
correct\_predictions1

## [1] 0.5865385

* The percentage of correct predictions on the test data is 58.65%%.

#### i)

The regression model percentage of correct predictions is 62.5% of the time which is the highest between all the models. The regression method appears to provide the best results.

#### j)

# Logistic regression with Lag2:Lag1  
glm.fit = glm(Direction ~ Lag2:Lag1, data = Weekly, family = binomial, subset = train)  
glm.probs = predict(glm.fit, Weekly.910, type = "response")  
glm.pred = rep("Down", length(glm.probs))  
glm.pred[glm.probs > 0.5] = "Up"  
Direction.0910 = Direction[!train]  
table(glm.pred, Direction.910)

## Direction.910  
## glm.pred Down Up  
## Down 1 1  
## Up 42 60

mean(glm.pred == Direction.910)

## [1] 0.5865385

# LDA with Lag2 interaction with Lag1  
lda.fit = lda(Direction ~ Lag2:Lag1, data = Weekly, subset = train)  
lda.pred = predict(lda.fit, Weekly.910)  
mean(lda.pred$class == Direction.910)

## [1] 0.5769231

# QDA with sqrt(abs(Lag2))  
qda.fit = qda(Direction ~ Lag2 + sqrt(abs(Lag2)), data = Weekly, subset = train)  
qda.class = predict(qda.fit, Weekly.910)$class  
table(qda.class, Direction.910)

## Direction.910  
## qda.class Down Up  
## Down 12 13  
## Up 31 48

mean(qda.class == Direction.910)

## [1] 0.5769231

# KNN k =10  
knn.pred = knn(train.X, test.X, train.Direction, k = 10)  
table(knn.pred, Direction.910)

## Direction.910  
## knn.pred Down Up  
## Down 17 18  
## Up 26 43

mean(knn.pred == Direction.0910)

## [1] 0.5769231

# KNN k = 100  
knn.pred = knn(train.X, test.X, train.Direction, k = 100)  
table(knn.pred, Direction.910)

## Direction.910  
## knn.pred Down Up  
## Down 9 12  
## Up 34 49

mean(knn.pred == Direction.910)

## [1] 0.5576923

* Out of these combinations, the original logistic regression and LDA have the best performance in terms of test error rates.

### Exercise 15:

#### a)

Power = function() {  
 2^3  
}  
print(Power())

## [1] 8

#### b)

Power2 = function(x, a) {  
 x^a  
}  
Power2(3, 8)

## [1] 6561

#### c)

Power2(10, 3)

## [1] 1000

Power2(8, 17)

## [1] 2.2518e+15

Power2(131, 3)

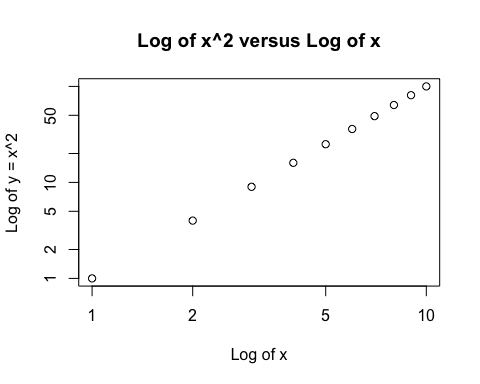
## [1] 2248091

#### d)

Power3 = function(x, a) {  
 result = x^a  
 return(result)  
}

#### e)

x = 1:10  
plot(x, Power3(x, 2), log = "xy", ylab = "Log of y = x^2", xlab = "Log of x",   
 main = "Log of x^2 versus Log of x")



#### f)

PlotPower = function(x, a) {  
 plot(x, Power3(x, a))  
}  
PlotPower(1:10, 3)

