Machine Learning_Assignment2

谢嘉薪 2020111142

最后编译于 11/11/2022

1 练习 & 第二次作业

This question should be answered using the `Weekly` data set, which is part of the `ISLR2` package. It contains 1, 089 weekly returns for 21 years, from the beginning of 1990 to the end of 2010.

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

```
library(ISLR2)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.3.6      v purrr   0.3.4
## v tibble  3.1.8      v dplyr   1.0.10
## v tidyr   1.2.1      v stringr 1.4.1
## v readr   2.1.3      v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
data(Weekly)
attach(Weekly)
str(Weekly)
```

```
## 'data.frame':    1089 obs. of  9 variables:
##  $ Year     : num  1990 1990 1990 1990 1990 1990 1990 1990 1990 1990 ...
##  $ Lag1     : num  0.816 -0.27 -2.576 3.514 0.712 ...
##  $ Lag2     : num  1.572 0.816 -0.27 -2.576 3.514 ...
##  $ Lag3     : num  -3.936 1.572 0.816 -0.27 -2.576 ...
##  $ Lag4     : num  -0.229 -3.936 1.572 0.816 -0.27 ...
##  $ Lag5     : num  -3.484 -0.229 -3.936 1.572 0.816 ...
##  $ Volume   : num  0.155 0.149 0.16 0.162 0.154 ...
##  $ Today    : num  -0.27 -2.576 3.514 0.712 1.178 ...
##  $ Direction: Factor w/ 2 levels "Down","Up": 1 1 2 2 2 1 2 2 2 1 ...
```

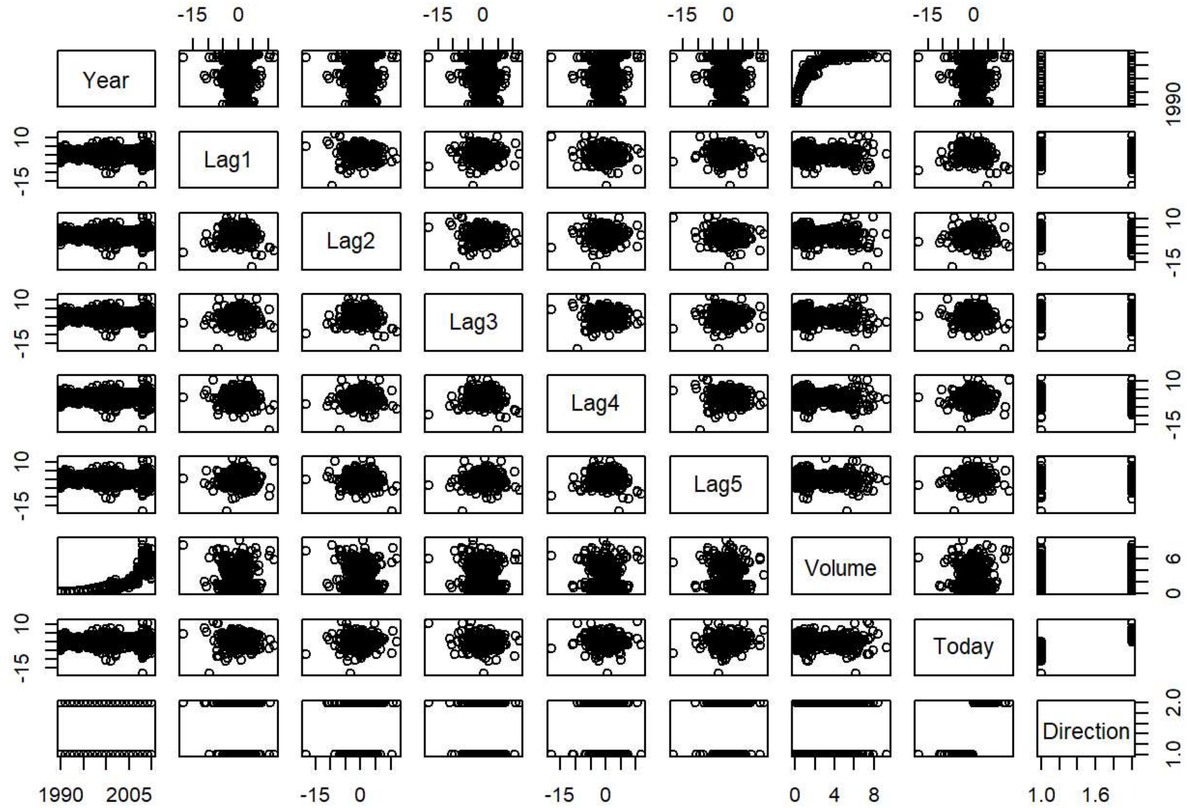
```
summary(Weekly)
```

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

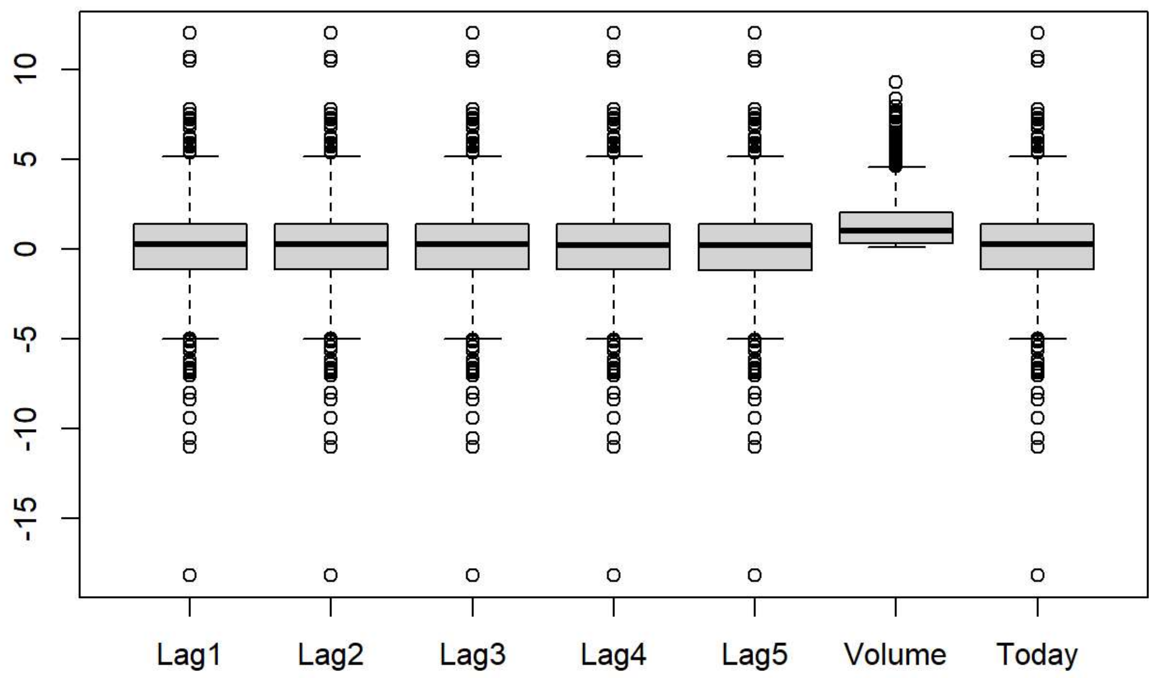
```
cor(Weekly[,-9])
```

```
##               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
```

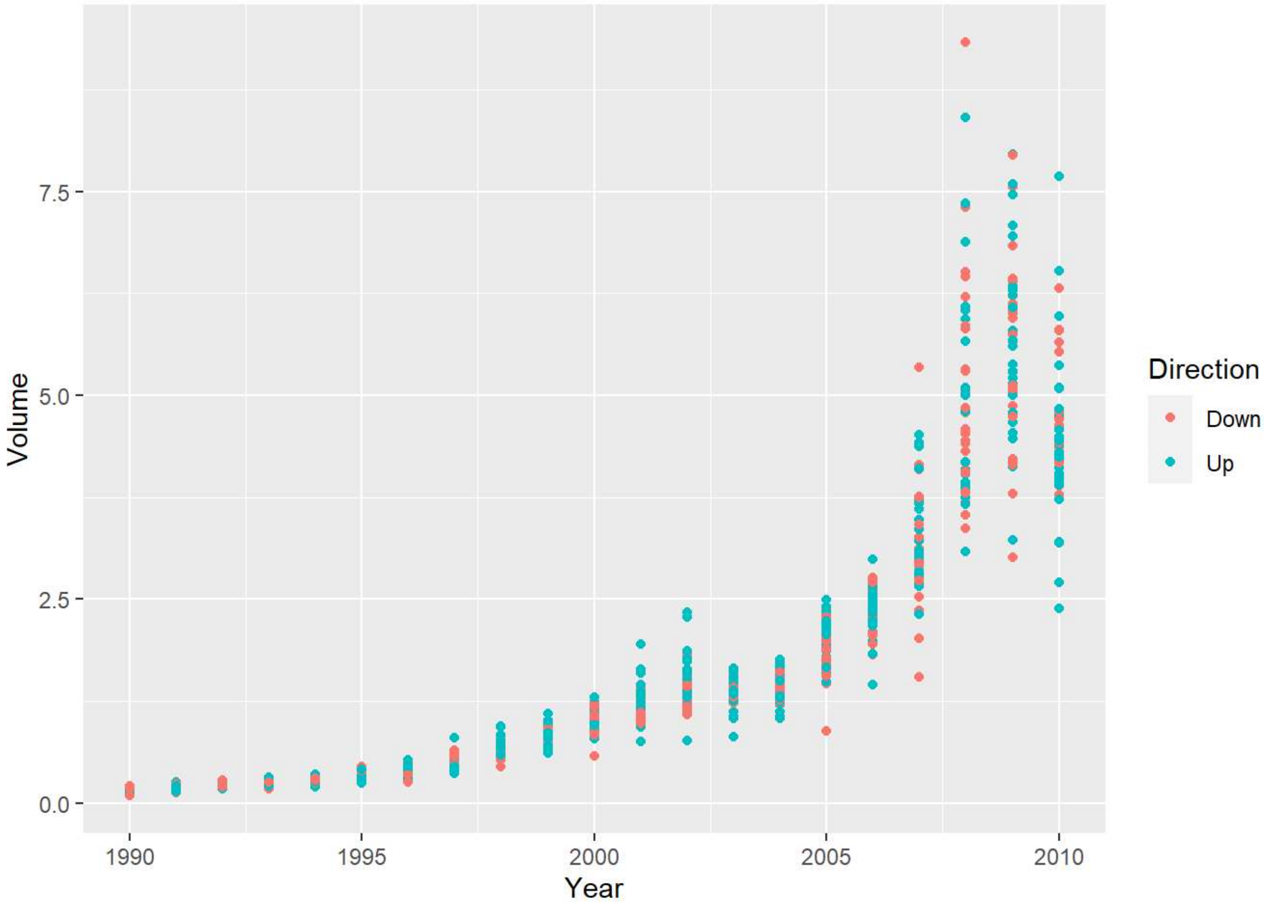
```
pairs(Weekly)
```



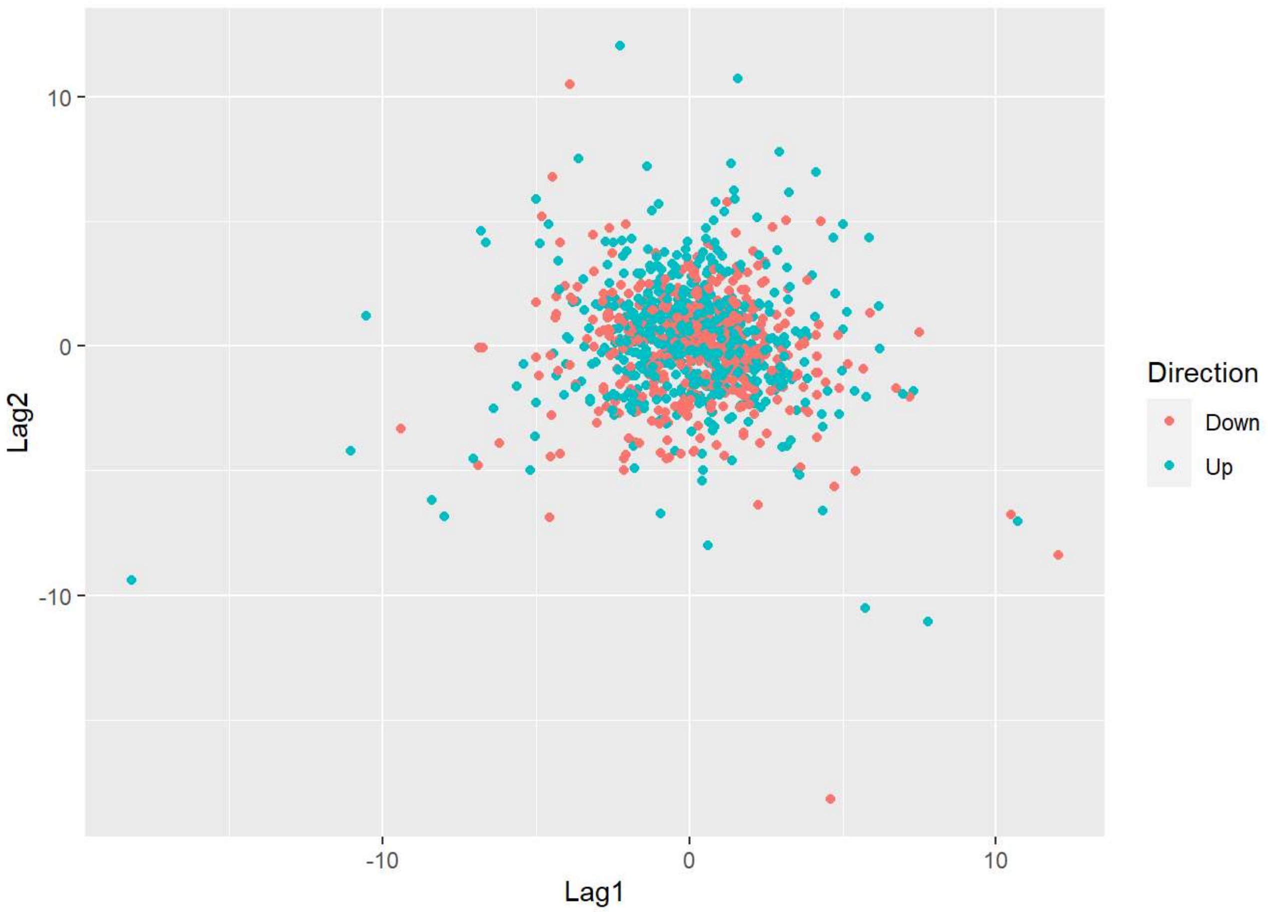

```
boxplot(Weekly[,2:8])
```



```
ggplot(Weekly) +
  geom_point(mapping = aes(x=Year, y=Volume, color=Direction))
```

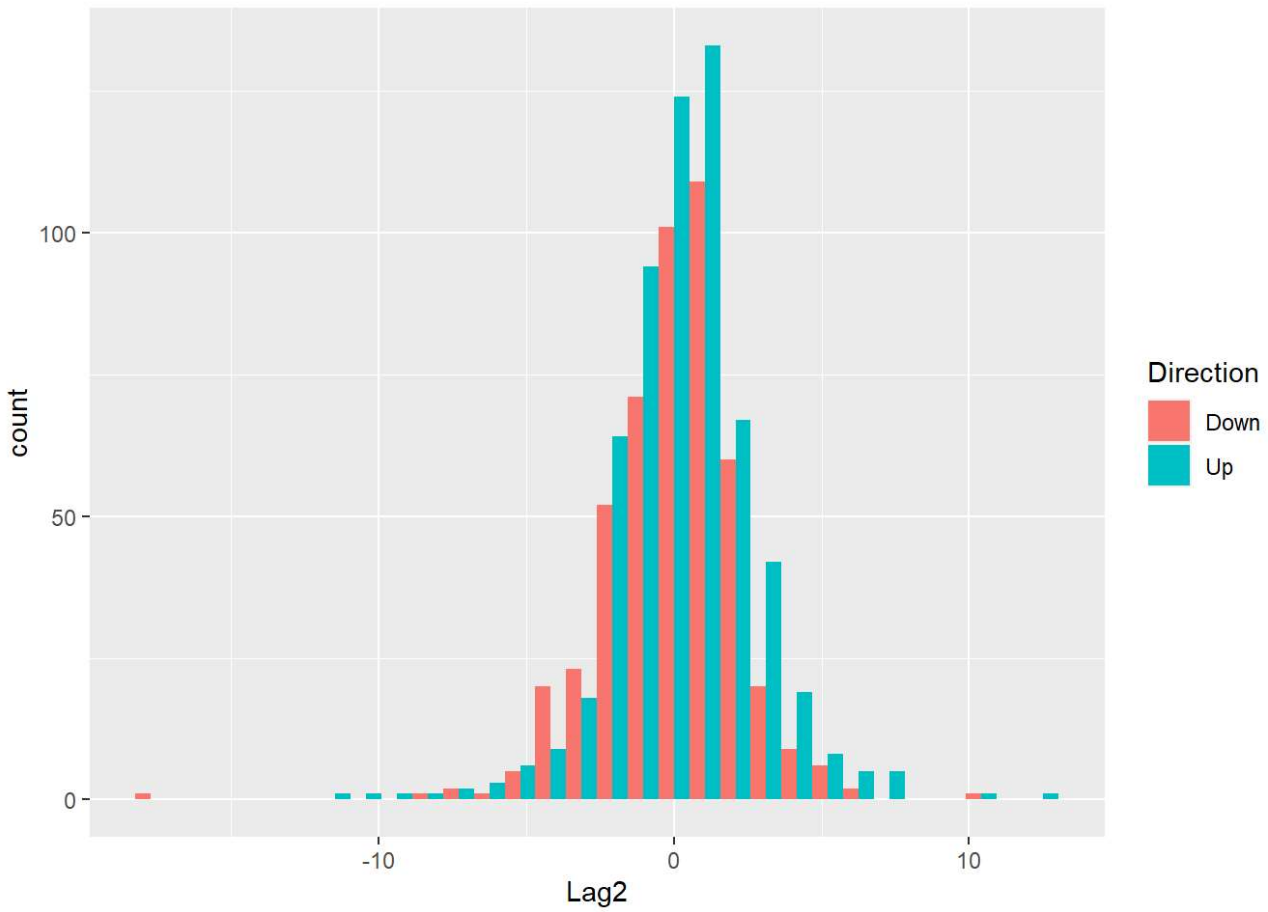



```
ggplot(Weekly) +
  geom_point(mapping = aes(x=Lag1, y=Lag2, color=Direction))
```




```
ggplot(Weekly) +
  geom_histogram(aes(x=Lag2,fill=Direction),position = "dodge")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



2. 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?

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

3. 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.

```
glm.probs <- predict(glm.fit, type='response')
pred <- rep('Down', 1089)
pred[glm.probs>.5] = 'Up'
table(pred, Direction)
```

```
##       Direction
## pred   Down  Up
##   Down   54  48
##   Up    430 557
```

```
mean(pred == Direction)
```

```
## [1] 0.5610652
```

> 左上角的54表示真实值和预测值都为Down的有54个；右下角557表示真实值和预测值都为Up的有557个；右上角表示真实为Up，预测为Down的有48个，即假阴错误；左下角表示真实为Down，预测为Up的有430个，即假阳错误。

4. 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).

```
train <- Weekly[Weekly['Year'] < 2009,]
test <- Weekly[Weekly['Year'] >= 2009,]
dim(train)
```

```
## [1] 985   9
```

```
dim(test)
```

```
## [1] 104   9
```

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

```
##
## Call:
## glm(formula = Direction ~ Lag2, family = binomial, data = 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
```

```
glm.prob <- predict(glm.fit, test, type = 'response')
glm.pred <- rep('Down', 104)
glm.pred[glm.prob>0.5] <- 'Up'
table (glm.pred, test$Direction)
```

```
##
## glm.pred Down Up
##     Down    9  5
##     Up     34 56
```

```
mean(glm.pred == test$Direction)
```

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

5. Repeat 4. using LDA.

```
library(MASS)
```

```
##
## 载入程辑包：'MASS'
```

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

```
## The following object is masked from 'package:ISLR2':
##
##     Boston
```

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

```
## Call:
## lda(Direction ~ Lag2, data = 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, test, type = 'response')
table(lda.pred$class, test$Direction)
```

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

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

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

6. Repeat 4. using QDA.

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

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

```
qda.pred <- predict(qda.fit, test, type = 'response')
table(qda.pred$class, test$Direction)
```

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

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

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

7. Repeat 4. using KNN with K = 1. You can also experiment with values for K in the KNN classifier. (Hint: Use `knn()` in the `class` package.)

```
library(class)
train.matrix = as.matrix(train['Lag2'])
test.matrix = as.matrix(test['Lag2'])
set.seed(2020111142)
knn.pred = knn(train.matrix, test.matrix, train$Direction, k = 1)
table(knn.pred, test$Direction)
```

```
##
## knn.pred Down Up
##     Down   21 29
##     Up     22 32
```

```
mean(knn.pred == test$Direction)
```

```
## [1] 0.5096154
```

8. Repeat 4. using naive Bayes.

```
library(e1071)
nb.fit <- naiveBayes(Direction ~ Lag2, data = train)
nb.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
```

```
nb.class <- predict(nb.fit, test)
table(nb.class, test$Direction)
```

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

```
mean(nb.class == test$Direction)
```

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

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

> 根据混淆矩阵和准确率来看，Logistic回归和LDA方法准确率最高。

要求

- 11月11日周五晚24点截止上交，上交pdf文件（一定要pdf，否则无法批改，可以Knit直接生成或html转存）至邮箱：lyfsufe@163.com (mailto:lyfsufe@163.com)
- 务必创建一个新的Rmd文件，不要使用我们的教学文档直接上交作业